# Interface Homem-Máquina

# Trabalho 1

# Versão 1.0

# Histórico da Revisão

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** |
| 07/12/2012 | 1.0.0 | Criação do documento | Lucas de OLiveira |

# Índice Analítico

1. As Heurísticas de Nielsen 4

1.1 Visibilidade do estado do sistema: 4

1.2 Correspondência entre o sistema e o mundo real 4

1.3 Liberdade e controle dos usuários 4

1.4 Consistência e normas 4

1.5 Prevenção de erros 5

1.6 Reconhecimento em vez de Recordação 5

1.7 Flexibilidade e eficiência na utilização 5

1.8 Desenho estético e minimalista 6

1.9 Ajudar os usuários a reconhecer, diagnosticar e recuperar erros 6

1.10 Documentação e Ajuda 6

2. Os Princípios de Desenho de Norman 7

2.1 Fornecer um bom modelo conceitual 7

2.2 Tornar as coisas visíveis 7

2.3 O Princípio do Mapeamento 7

2.4 O Princípio do Feedback 7

3. Os Princípios de Desenho de Constantine & Lockwood 8

3.1 Simplicidade 8

3.2 Visibilidade 8

3.3 Feedback 8

3.4 Tolerância 8

3.5 Reutilização 9

4. As Oito Regras de Ouro de Shneiderman 9

4.1 Lutar pela consistência 9

4.2 Permitir aos usuários freqüentes a utilização de atalhos 9

4.3 Utilizar feedback Informativo 9

4.4 Desenhar as Caixas de Diálogo Fechadas 10

4.5 Utilizar Prevenção e Tratamento de Erros 10

4.6 Permitir a fácil reposição de ações 10

4.7 Suportar a localização interna de controle 10

4.8 Reduzir a carga sobre a memória de curta-duração 11

# Trabalho 1

# As Heurísticas de Nielsen

## Visibilidade do estado do sistema:

O sistema deve manter os usuários sempre informados sobre o que está acontecendo através de feedback rápido e apropriado;

* **definição pessoal:** Todo aplicativo tem, como pressuposto para o seu bom desempenho, manter o usuário informado de suas atividades e condição atual (Ex: preparado para uso, processando, baixando algum arquivo), sob pena de o usuário ter que descobrir por si mesmo a condição do sistema (e geralmente o usuário chega à conclusão errada). Portanto, devemos dizer através do sistema, de forma explícita, qual o seu atual estado. Isso possibilita que o usuário tome decisões acertadas sobre a atividade que está executando, no momento.
* **exemplos concretos de utilização:** Notificação de recebimento de mensagens do Facebook, aviso de novas atualizações disponiveis no Windows.
* **exemplos de não utilização:**

## Correspondência entre o sistema e o mundo real

O sistema deve “falar” a linguagem do usuário, com conceitos que lhe são familiares, e de uma forma lógica e natural que evite termos técnicos;

* **definição pessoal:** Não usar palavras de sistema, que não fazem sentido pro usuário. Toda a comunicação do sistema precisa ser contextualizada ao usuário, e ser coerente com o chamado modelo mental do usuário.
* **exemplos concretos de utilização:** calculadora do Windows, site de busca (Google), Microsoft Office.
* **exemplos de não utilização:**

## Liberdade e controle dos usuários

O sistema deve fornecer formas do usuário sair de estados não desejados sem ter que passar por diálogos extensos. Suportar undo/redo (desfazer/refazer).

* **definição pessoal:** Facilite as “saídas de emergência” para o usuário, permitindo desfazer ou refazer a ação no sistema e retornar ao ponto anterior, quando estiver perdido ou em situações inesperadas.
* **exemplos concretos de utilização:** Desfazer e refazer do pacote Microsoft Office.
* **exemplos de não utilização:**

## Consistência e normas

Evitar termos, situações e ações diferentes com o mesmo significado. Seguir as convenções das plataformas, normas e regras.

* **definição pessoal:** Esse é um grande problema dos aplicativos, em todos os tempos. Um sistema, para ser compreensível, deve utilizar símbolos iguais para a mesma funcionalidade, em qualquer contexto, e símbolos diferentes para funcionalidades diferentes. Em tempo: símbolos, aqui, referem-se às palavras e ícones utilizados na interface, para distinguir as funcionalidades. Ainda, a localização dos itens de interface, os códigos de cores e os símbolos de uso geral devem estar de acordo com os utilizados no sistema operacional corrente (isto é, não pode fugir do Guia de Estilo daquela plataforma).
* **exemplos concretos de utilização:** O site Eureka.
* **exemplos de não utilização:**

## Prevenção de erros

Desenhar o sistema de forma cuidadosa prevenindo os problemas antes de acontecerem.

* **definição pessoal:** Na tradução livre das palavras do próprio Nielsen “Ainda melhor que uma boa mensagem de erro é um design cuidadoso que possa prevenir esses erros”. Por exemplo, ações definitivas, como deleções ou solicitações podem vir acompanhadas de um checkbox ou uma mensagem de confirmação.
* **exemplos concretos de utilização:** Sistemas de gravação de disco, ImgBurn, onde o sistema avisa que o disco não pode ser suficiente e pode gerar erros antes de grava-lo.
* **exemplos de não utilização:**

## Reconhecimento em vez de Recordação

Tornar os objetos, ações e opções visíveis. O usuário não deve ter que recordar informação de uma parte do diálogo para outra, As instruções de utilização do sistema devem ser visíveis e acessíveis quando necessárias.

* **definição pessoal:** Evite acionar a memória do usuário o tempo inteiro, fazendo com que cada ação precise ser revista mentalmente antes de ser executada. Permita que a interface ofereça ajuda contextual, e informações capazes de orientar as ações do usuário - ou seja - que o sistema dialogue com o usuário.
* **exemplos concretos de utilização:** Chats como Windows Live Messanger, onde é guardado o histórico de conversas com o usuário. Browsers como Google Chrome que guarda os usuários e senha.
* **exemplos de não utilização:**

## Flexibilidade e eficiência na utilização

Acomodar diferentes níveis de experiência fornecendo “aceleradores” para os usuários experientes que são invisíveis para os novatos. Permitir aos usuários customizar as ações freqüentes.

* **definição pessoal:** O sistema precisa ser fácil para usuários leigos, mas flexível o bastante para se tornar ágil à usuários avançados. Essa flexibilidade pode ser conseguida com a permissão de teclas de atalhos, por exemplo. No caso de websites, uso de máscaras e navegação com tab em formulários são outros exemplos.
* **exemplos concretos de utilização:**
* **exemplos de não utilização:**

## Desenho estético e minimalista

Evitar informação, irrelevante ou raramente necessária, em dialogos desta forma diminuindo a visibilidade da informação relevante.

* **definição pessoal:** Evite que os textos e o design fale mais do que o usuário necessita saber. Os “diálogos” do sistema precisam ser simples, diretos e naturais, presentes nos momentos em que são necessários.
* **exemplos concretos de utilização:**
* **exemplos de não utilização:**

## Ajudar os usuários a reconhecer, diagnosticar e recuperar erros

As mensagens de erro devem ser expressas em linguagem simples, indicando precisamente o problema, e sugerindo uma solução de forma construtiva.

* **definição pessoal:** As mensagens de erro do sistema devem possuir uma redação simples e clara que ao invés de intimidar o usuário com o erro, indique uma saída construtiva ou possível solução.
* **exemplos concretos de utilização:**
* **exemplos de não utilização:**

## Documentação e Ajuda

A ajuda e documentação, quando necessária, deve estar concentrada nas tarefas dos usuários, e deve ser fácil, concisa e acessível.

* **definição pessoal:** Um bom design deveria evitar ao máximo à necessidade de ajuda na utilização do sistema. Ainda assim, um bom conjunto de documentação e ajuda deve ser utilizado para orientar o usuário em caso de dúvida. Deve ser visível, facilmente acessada, e oferecer uma ferramenta de busca na ajuda.
* **exemplos concretos de utilização:**
* **exemplos de não utilização**

# Os Princípios de Desenho de Norman

## Fornecer um bom modelo conceitual

Um bom modelo conceitual permite aos usuários prever os efeitos das suas ações, o que esperar e como reagir quando as coisas correm mal. O modelo conceitual fornecido pelos designers deve resultar numa imagem consistente e coerente do sistema.

* **definição pessoal:** 1. Permite prever os efeitos de nossas ações. Relacionamento entre os controles e os resultados. 2. Sem um bom modelo conceitual operamos às cegas. Fazemos o que nos dizem, sem saber as conseqüências. 3. Quando as coisas não funcionam perfeitamente. Não sabemos o que fazer para restaurar o processo
* **exemplos concretos de utilização:**
* **exemplos de não utilização:**

## Tornar as coisas visíveis

O usuário não deve ter problemas em perceber o estado do sistema e as alternativas para ação.

* **definição pessoal:** O usuário deve saber dizer o estado em que se encontra um dispositivo e as opções de ação a partir daquele estado apenas olhando para ele. Visibilidade é a capacidade de uma interface de informar ao seu usuário sobre as ações disponíveis em determinado momento ou sobre o estado em que se encontra o equipamento ou sistema de software.
* **exemplos concretos de utilização:**
* **exemplos de não utilização:**

## O Princípio do Mapeamento

Os usuários devem determinar claramente a relação entre as ações e os resultados, os controles e os seus efeitos, e entre o estado do sistema e o que é visível.

* **definição pessoal:** Mapeamento é uma propriedade que se refere à determinação do relacionamento entre as ações e os resultados, entre os controles e seus efeitos, entre o estado do sistema e o que é visível. Um bom desenho de um objeto deve facilitar o mapeamento ao usuário. Bons mapeamentos tiram vantagens de analogias físicas e padrões culturais.
* **exemplos concretos de utilização:**
* **exemplos de não utilização:**

## O Princípio do Feedback

Os usuários devem receber feedback contínuo e informativo sobre o resultado das suas ações.

* **definição pessoal:** Todo aplicativo tem, como pressuposto para o seu bom desempenho, manter o usuário informado de suas atividades e condição atual.
* **exemplos concretos de utilização:**
* **exemplos de não utilização:**

# Os Princípios de Desenho de Constantine & Lockwood

## Simplicidade

Tornar ações simples e comuns fáceis de executar, comunicando de forma clara e simples na linguagem do usuário. Fornecer atalhos com significado para as ações mais demoradas.

* **definição pessoal:** Tornar ações simples e comuns fáceis de executar, comunicando de forma clara e simples na linguagem do usuário.
* **exemplos concretos de utilização:**
* **exemplos de não utilização:**

## Visibilidade

Manter visíveis todas as opções e materiais necessários para uma determinada tarefa, sem distrair os usuários com informação redundante.

* **definição pessoal:** Mantenha visíveis as opções necessárias para a realização de uma tarefa, sem distrair o usuário com informação redundante e fora de contexto. As ferramentas e materiais devem estar disponíveis onde e quando forem necessárias, todas as opções.Permita que usuário identifique facilmente as ferramentas disponíveis.
* **devem ser explícitas e aparentes ao usuário.**
* **exemplos concretos de utilização:**
* **exemplos de não utilização:**

## Feedback

Manter os usuários informados das ações ou interpretações, das mudanças de estado ou condição, e dos erros e exceções que são relevantes ou de interesse para o usuário. Oferecer essa informação de forma clara, concisa e não ambígua numa linguagem familiar aos usuários.

* **definição pessoal:** Mantenha os usuários informados de ações e interpretações, mudanças de estado ou condição, erros ou exceções que sejam de seu interesse , através de uma linguagem clara, concisa, não ambígua e familiar ao usuário. O estado do sistema é uma informação importante para o usuário se situar na interação. Em muitas situações, o feedback sutil, eficiente mas sem chamar a atenção, não intrusivo, é recomendado. Um bom exemplo disso é um cursor de mouse que assume diferentes formas dependendo do elemento da interface que está sendo apontado.
* **exemplos concretos de utilização:**
* **exemplos de não utilização:**

## Tolerância

Ser flexível e tolerante, reduzindo os custos dos erros e da má utilização, permitindo a reversão de ações e a prevenção dos erros.

* **definição pessoal: S**eja flexível e tolerante, reduzindo os custos de erros e mau uso através da disponibilização de “desfazer” (undos) e “refazer” ( redos), prevenindo erros, tolerando uma variedade de entradas e seqüências e interpretando todas ações razoáveis de uma maneira razoável. O sistema deve ser flexível, para, quando possível, buscar a interação com o usuário de forma razoável.
* **exemplos concretos de utilização:**
* **exemplos de não utilização:**

## Reutilização

Reutilizar componentes e comportamentos internamente e externamente, mantendo a consistência com propósito e não de uma forma arbitrária. Desta forma reduzir a necessidade dos usuários repensarem e recordarem informação.

* **definição pessoal:** Reuse objetos e comportamentos internos e externos, mantendo consistência ao invés de fazê-la arbitrariamente, desse modo reduzindo a necessidade do usuário pensar e ter que se lembrar. A reutilização ajuda não só os usuários como também os programadores, porem a reutilização deve ser racional e utilizar consistências soluções bem elaboradas como fonte do reuso.
* **exemplos concretos de utilização:**
* **exemplos de não utilização:**

# As Oito Regras de Ouro de Shneiderman

## Lutar pela consistência

Terminologia, menus, telas de ajuda, cor, maiúsculas/minúsculas...

* **definição pessoal:** Seqüência de ações similares para procedimentos similares. Manter um padrão visual para as cores, Layout e fontes. Utilizar a mesma terminologia em menus.
* **exemplos concretos de utilização:**
* **exemplos de não utilização:**

## Permitir aos usuários freqüentes a utilização de atalhos

Abreviaturas, teclas especiais, comandos escondidos, capacidades macro

* **definição pessoal:** Teclas de atalho, macros e navegação simples facilitam e agilizam a interação do usuário mais experientes com a interface.
* **exemplos concretos de utilização:**
* **exemplos de não utilização:**

## Utilizar feedback Informativo

Cada ação do usuário deve gerar feedback.

* **definição pessoal:** Toda e qualquer ação do usuário requer uma resposta do sistema, cujo qual será mais ou menos explicativa dependendo do tipo de ação a ser executada.
* **exemplos concretos de utilização:**
* **exemplos de não utilização:**

## Desenhar as Caixas de Diálogo Fechadas

As seqüências de ações devem estar organizadas em grupos (sentido de início, meio e fim).

* **definição pessoal:** As seqüências de ações do sistema deve ser organizada de tal forma que o usuário consiga entender os passos e saiba quando cada um deles for executado com sucesso.
* **exemplos concretos de utilização:**
* **exemplos de não utilização:**

## Utilizar Prevenção e Tratamento de Erros

Evitar situações de erro e que o sistema fique instável.

* **definição pessoal:** A interface não pode dar vias para o usuário cometer erros graves, e caso ocorram erros, devem haver mecanismos que tratem, corrijam na medida do possível, e caso não seja possível, instrua o usuário para uma possível solução.
* **exemplos concretos de utilização:**
* **exemplos de não utilização:**

## Permitir a fácil reposição de ações

Sempre que possível as ações devem ser reversíveis.

* **definição pessoal:** Sempre que possível, as ações devem ser reversíveis, de forma que tranqüilize o usuário e lhe dá mais coragem para explorar o sistema.
* **exemplos concretos de utilização:**
* **exemplos de não utilização:**

## Suportar a localização interna de controle

Os usuários gostam de sentir que detêm o controle do sistema. Evitar situações casuais obrigando o usuário a iniciar as ações em vez de simplesmente responder.

* **definição pessoal:** Os usuários mais experientes devem ter a sensação de que eles dominam os processos do sistema e que ele apenas responde a suas ações.
* **exemplos concretos de utilização:**
* **exemplos de não utilização:**

## Reduzir a carga sobre a memória de curta-duração

A regra do +7 ou -2 blocos de informação. Simplificar o desenho, consolidar páginas múltiplas, reduzir a freqüência de movimento de janelas e assegurar treino para mnemônicas, seqüências de ações e códigos.

* **definição pessoal:** O sistema deve conter uma interface simples para memorização. Para isso requer uma boa Estrutura e Equilíbrio para relacionar elementos e facilitar a memorização subjetiva das telas, sem exigir esforço.
* **exemplos concretos de utilização:**
* **exemplos de não utilização:**