

# **10 heurísticas para o design de interface (Jacob Nielsen)**

## **1. Visibilidade do Status do Sistema**

No geral o ser humano é muito dependente da visão, ainda assim, quando estamos em um ambiente físico temos outros sentidos que nos informam o que está acontecendo, quando falamos de um sistema, software ou interface essa história muda um pouco. Em ambientes digitais a dependência da visão é ainda maior e por conta disso é essencial que a interface forneça ao usuário o status em relação à sua posição dentro do sistema, ou seja, informar o usuário sobre qual ambiente ele estava, em qual ele está e para quais outros ambientes ele poderá se dirigir a partir de sua localização.

## **2. Compatibilidade entre o sistema e o mundo real**

Um sistema precisa falar a mesma linguagem do usuário, não apenas o idioma em si mas a linguagem que o usuário utiliza no dia-a-dia. Outro aspecto da compatibilidade entre sistema e mundo real são os símbolos utilizados dentro de uma interface. Parece algo sem muita importância mas escolher corretamente quais ícones serão utilizados para colocar em uma interface pode facilitar ou prejudicar a compreensão das informações.

## **3. Controle e liberdade para o usuário**

Aquele momento em que você acaba deletando um e-mail que tinha um documento importante anexado sem querer e fica desesperado, mas ao ir até a lixeira você o encontra e fica aliviado. O ponto aqui é que quando o usuário realiza ações por engano, algo que é frequente, o sistema deve apresentar ao usuário uma “saída de emergência” daquele estado no qual ele entrou para o estado de segurança em que estava, em outras palavras, dê ao usuário o “Ctrl+Z” pois essa possibilidade de reverter ações remove a insegurança do usuário ao utilizar o aplicativo, site, etc.

## 4. Consistência e Padronização

Essa heurística é um tanto simples mas muitas vezes não aplicada na construção de interfaces. Manter consistência entre as telas de uma aplicação é essencial para que não seja necessário o entendimento de vários padrões e formas de interações diferentes para cada tela, uma vez aprendido será algo replicável em outros contextos. Além disso a experiência de uso se torna muito mais interessante pois não existirá aquela sensação de estar perdido. Muitas vezes o motivo pelo qual usuários não interagem com as aplicações é essa sensação causada pela falta de consistência e padronização.

Um grande exemplo de consistência e padronização é o próprio Google ([Material Design](#)).

## 5. Prevenção de erros

Existem dois tipos de erros que os usuários normalmente cometem: o deslize e o engano. Pode até parecer a mesma coisa mas são sutilmente diferentes. O deslize é quando um usuário pretende realizar uma ação mas acaba realizando outra, isso acontece tipicamente quando ele não está totalmente focado em sua ação dentro da aplicação. Já o engano é quando a compreensão de alguma informação é equivocada ou entendida de outra forma. Portanto melhor do que mensagens dizendo que o usuário cometeu algum erro é prevenir que o mesmo não cometa esse erro. Caixas de confirmação, como as que aparecem quando você deleta um arquivo, são um exemplo de como evitar erros.

## 6. Reconhecimento em vez de memorização

O cérebro é muito bom em reconhecer padrões e na medida em que objetos, ações e opções ficam expostos para o usuário mais dicas chegam ao cérebro tornando certas ações familiares, ou seja, é preferível dar ao usuário formas de reconhecer padrões do que ter que obrigá-lo a memorizar várias informações na medida que ele navega pela aplicação. A grande diferença entre reconhecer e memorizar é a quantidade de dicas que são fornecidas para que um conhecimento seja acessado, reconhecer padrões fornece muito mais dicas do que tentar acessar memórias. Tente lembrar de uma senha (memorização) ou tente salvar um arquivo no Excel (reconhecimento).

## **7. Eficiência e flexibilidade de uso**

O ideal é que a interface seja útil tanto para usuários leigos como para experientes. Leigos precisam de informações mais detalhadas para poderem realizar tarefas mas na medida que crescem em conhecimento sobre a interface, a necessidade de formas mais rápidas de interação para realizar uma tarefa começam a surgir. “Alt+Tab ou Ctrl+C e Ctrl+V ou Windows+D” são exemplos de atalhos que permitem ao usuário mais experiente realizar tarefas mais rapidamente.

## **8. Estética e design minimalista**

Quanto maior a quantidade de informação maior será a quantidade de informações que serão analisadas e decisões que o usuário precisará tomar, por isso, é crucial manter apenas as informações que são realmente necessárias, as informações secundárias podem ser deixadas em segundo plano (menus, abas, etc.) assim a aplicação se torna muito eficiente no quesito transmitir informação aos usuários. Um exemplo bem bacana é o próprio [Medium](#).

## **9. Ajude os usuários a reconhecerem, diagnosticarem e recuperarem-se de erros**

O ponto aqui é ajudá-lo a reparar um erro depois de ele ter cometido caso o “Ctrl+Z” não tenha funcionado.

Um exemplo são aqueles avisos de formulários nos campos que não foram preenchidos corretamente. É uma forma simples de mostrar para o usuário que ele cometeu um erro, onde errou e o que precisa ser feito para corrigir tal erro.

## **10. Ajuda e documentação**

Normalmente essas são as áreas menos acessadas mas ainda assim elas são importantes dentro de um sistema pois nunca se sabe quando um usuário irá precisar de uma ajudinha. É como se fosse uma maneira “faça você mesmo” do usuário resolver suas dúvidas de quais ações tomar dentro da aplicação tornando-o mais independente do suporte.