ENGENHARIA DE USABILIDADE

HEURÍSTICAS DE USABILIDADE

Olá!

Nesta aula, discutiremos sobre as heurísticas de usabilidade.

Em computação, o termo heurística refere-se a uma boa recomendação ou a uma boa prática que se aplica a todos os casos de um mesmo tipo.

As heurísticas de usabilidade são, portanto, recomendações a serem seguidas ao longo do projeto de interfaces, de modo a garantir que o resultado seja uma aplicação que não cause problemas de interação ao usuário.

Ao fim desta aula, você será capaz de:

1- Definir as heurísticas de usabilidade;

2- Reconhecer um problema relacionado a uma das heurísticas;

3- Propor soluções com base nas recomendações propostas pelas heurísticas.

1 Heurísticas de Nielsen e Molich

Existem listas imensas de recomendações para a construção de interfaces livres de problemas de usabilidade.

Entretanto, de tão extensas, essas listas chegam a intimidar os projetistas, que acabam por ignorá-las por completo.

Como proposta de solução para tal problema, Jakob Nielsen e Rolf Molich resolveram criar 10 heurísticas gerais, relacionadas aos princípios mais amplos de usabilidade, que podem ser utilizadas para explicar grande parte dos problemas observados nas interfaces.

Rolf Molich é consultor independente e possui mais de 25 anos de experiência na indústria de software. Molich trabalha na área de usabilidade desde 1984. Atualmente, é proprietário da DialogDesign, empresa a qual se dedica desde 1997. Molich planejou e coordenou a Avaliação Comparativa de Usabilidade, na qual cerca de 100 equipes profissionais testaram ou avaliaram um mesmo grupo de aplicações.

Conheça cada uma a seguir.

1.1 Heurística 1: Ofereça atalhos

Embora devesse ser possível operar uma interface com o conhecimento de apenas algumas regras gerais, também deveria ser possível que o usuário executasse suas tarefas mais frequentes de modo mais rápido, através de atalhos.

Aceleradores típicos incluem abreviações (teclas de função de executam uma sequência de comandos) com o pressionamento de somente uma tecla (o clique duplo em um objeto), de modo a executar as operações mais comuns, e botões disponíveis, de maneira mais direta, para acesso a funções importantes, sem ter de passar por uma série de opções de menu.

A sugestão de entrada, que recomenda alternativas de palavras ou trechos para conclusão da entrada antes mesmo que o usuário conclua sua digitação, não é bem um atalho, mas acelera a interação e permite que o usuário não precise estar atento durante todo esse processo.

Um recurso semelhante ao da sugestão de entrada é o do clique antecipado: os usuários clicam antecipadamente no local do botão “Ok” antes mesmo que ele apareça.

Você precisa saber, entretanto, que é perigoso disponibilizar os recursos de sugestão de entrada e clique antecipado em todas as circunstâncias.

Uma mensagem crítica de alerta, por exemplo, não deve desaparecer antes de ser lida. Da mesma forma, o *buffer* de sugestões de entrada deve ser esvaziado caso haja algum erro na execução de um comando anterior, que pode fazer com que todo o restante da entrada do usuário torne-se inválida.

Também é interessante que os usuários possam **reutilizar seu histórico de interações**. Para isso, basta disponibilizar um menu com as últimas ações do usuário para que ele possa reutilizá-las sem ter de executar o passo a passo de cada comando.

Reutilizar seu histórico de interações

Embora seja mais fácil reutilizar comandos em interfaces de linha de comando, algumas interfaces de manipulação direta permitem que os usuários executem novamente a última tarefa ou repitam a última busca simplesmente clicando em um botão de atalho.

1.2 Heurística 2: Crie diálogos simples e naturais

As interfaces devem ser simples!

Cada informação ou recurso adicional na tela é mais uma coisa para o usuário aprender, mais uma possibilidade de interpretação errônea pelo mesmo ou algo a mais para atrapalhá-lo na busca pelo que realmente procura.

Além disso, as interfaces devem estar de acordo com a tarefa do usuário e devem reproduzi-la o mais fielmente possível.

Portanto, o mapeamento entre os conceitos do computador e os conceitos do usuário deve ser bastante simples, para que a dificuldade de navegação na interface seja minimizada.

O ideal é que sejam apresentadas somente as informações das quais o usuário necessita, no lugar e no ritmo que ele deve recebê-las. Informações utilizadas em conjunto devem ser exibidas também em conjunto e na mesma tela.

Além disso, os objetos de informação e as operações devem ser acessados em uma sequência que seja equivalente ao modo como os usuários irão executar suas tarefas, de maneira mais efetiva e produtiva.

Algumas vezes, a própria interface impõe essa sequência, mas costuma ser melhor deixar que o usuário controle o diálogo com a aplicação.

Para tal, é importante permitir que o usuário individual ajuste a sequência de atividades ou opções para que a mesma atenda a suas preferências. Ainda assim, o sistema pode facilitar o entendimento do diálogo por parte do usuário, sugerindo uma sequência preferencial.

A figura seguinte apresenta a tela principal da máquina de busca do Google, um excelente exemplo para a heurística ora discutida (de diálogos simples e naturais com o usuário). Observe que não há poluição visual e são mostrados ao usuário somente os objetos necessários para que ele realize uma busca na internet.

1.3 Heurística 3: Crie um sistema de ajuda e documente o sistema

Embora haja preferência que o sistema seja fácil de usar, a ponto de não precisar de nenhum tipo de **ajuda ou documentação** complementar à interface, esse objetivo nem sempre pode ser alcançado.

Exceto nos sistemas de máquinas automatizadas – nos quais o usuário deve ser guiado pela interface até a tarefa que deseja executar –, muitas aplicações contam com tanto recursos que acabam demandando a criação de um manual ou sistema de ajuda.

Além disso, os usuários mais avançados podem querer recorrer a um manual para aumentar o conhecimento da aplicação.

Ajuda ou documentação

Seja online ou impresso, um aspecto importante do sistema de ajuda e da documentação é que sua redação deve ser de qualidade! Estudos já concluíram que a qualidade do texto de ajuda é muitíssimo mais importante do que os mecanismos utilizados para acessá-lo. As informações contidas no texto devem ser corretas e refletir a versão do sistema atualmente em uso pelos usuários.

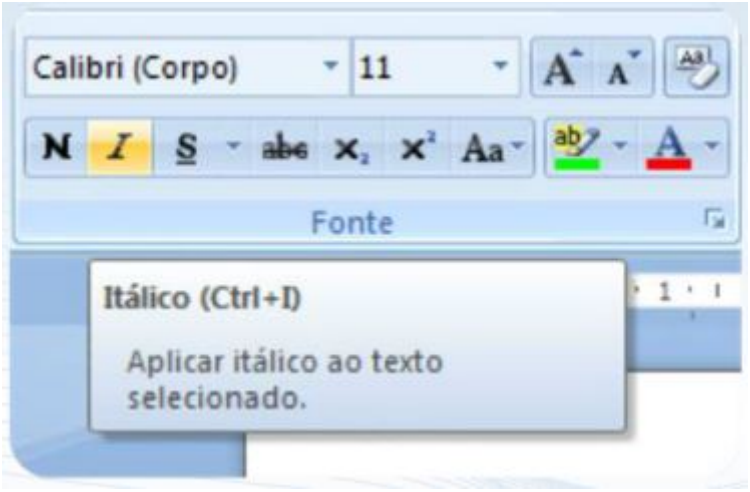
Temos de observar, entretanto, que a existência de um sistema de ajuda ou documentação não diminui os requisitos de usabilidade da interface em si. Dizer que as operações são explicadas no manual não é uma boa desculpa quando os usuários reclamam que a interface é complicada demais.

Infelizmente, a verdade é que muitos usuários não leem manuais, mas preferem investir tempo na execução de atividades que lhes permitem sentirem-se produtivos. Por isso, esses usuários tendem a iniciar a utilização do sistema sem ler as instruções.

Veja, na figura, um exemplo de sistema de ajuda online do aplicativo Paint. A explicação das funcionalidades do aplicativo é a primeira a ser mostrada ao usuário.

Outra vantagem do sistema de ajuda online é que pode ser sensível ao contexto.

Veja um exemplo de caixa de mensagem de ajuda do aplicativo MS Word.



Ao posicionar o ponteiro do mouse sobre o botão, o usuário recebe a explicação sobre o que esse botão faz, além de ser informado sobre a combinação de teclas de atalho para utilização do recurso.

Em uma interface gráfica, por exemplo, o usuário pode apontar para um objeto e imediatamente receber um auxilio através de uma explicação exibida em uma caixa de mensagem de ajuda. Além disso, mensagens de erro podem ser linkadas a textos do sistema de ajuda que possam explicar o erro e sugerir soluções para o mesmo.

Precisamos lembrar também que o problema do usuário costuma estar relacionado ao fato de querer fazer algo diferente do que lhe é oferecido pelo estado corrente do sistema.

Por isso, o sistema de ajuda deve permitir que o usuário faça perguntas orientadas à tarefa que deseja executar.

Veja um exemplo de sistema de ajuda online do aplicativo MS Word, no qual o usuário pode perguntar como executar determinada tarefa.

1.4 Heurística 4: Fale a linguagem do usuário

A terminologia utilizada para compor a parte textual das interfaces deve basear-se na **linguagem do usuário**, e não na terminologia orientada a sistemas.

Linguagem do usuário

Falar a linguagem do usuário não significa limitar o vocabulário da interface a um pequeno conjunto de palavras comumente utilizadas, mas atentar para o grupo de palavras que fazem parte do jargão empregado pelos potenciais usuários da aplicação.

Outro aspecto a ser observado é que os diálogos devem acontecer na língua nativa do usuário, e não em uma língua estrangeira.

Por exemplo:

Embora ainda não dicionarizado, o termo backupear (fazer uma cópia de segurança de um ou mais arquivos) é muito comum para a comunidade de Informática, mas será que é de fácil compreensão para usuários de outras áreas?

O site da Ordem dos Advogados do Brasil (OAB) conta com uma opção de menu que disponibiliza um conjunto de documentos a seus visitantes. Dentre as sub-opções desse menu estão Ementários e Súmulas.

Veja, a seguir, a tela principal do site da OAB e seu menu de opções definidas com terminologia específica da área de Direito:

Falar a linguagem do usuário também envolve visualizar as interações a partir de sua perspectiva.

Ao informar o saldo bancário de um usuário, por exemplo, o sistema deve dizer:

Você tem R$ 1.000.000,00 em sua conta (perspectiva do usuário).

Sendo assim, o sistema NÃO deve informar:

Estamos guardando R$ 1.000.000,00 para você (perspectiva do banco).

Uma maneira de tentar alcançar a meta de falar a linguagem do usuário é realizar um bom mapeamento entre as informações exibidas nas interfaces e o modelo conceitual que o usuário mantém sobre as mesmas informações.

As **metáforas** são um bom caminho para se conseguir tal mapeamento. Por meio delas, criam-se analogias entre a interface e o mundo real – ou seja, aquele onde está inserido o usuário –, de modo a criar interações cujo funcionamento o usuário já conheça.

1.5 Heurística 5: Previna a ocorrência de erros

Melhor do que exibir boas mensagens de erro é evitar que os erros aconteçam!

Muitas situações de interação estão propensas a erros, e os sistemas podem ser projetados de modo a evitar que o usuário cometa erros que podem ser antecipadamente tratados.

Erros com consequências especialmente graves também podem ter sua frequência reduzida quando são feitas perguntas de confirmação antes da execução da ação perigosa, do tipo:

Você realmente deseja executar esta operação?

É muito comum que erros aconteçam durante a entrada de dados em formulários!

Sem tratamento, esses erros podem acabar no banco de dados do sistema, criando uma série de inconsistências.

O ideal é auxiliar o usuário no que diz respeito à forma como os dados devem ser digitados, informando-o, por exemplo, se números de documentos devem ou não conter pontos e traços e se, nas datas, o ano deve conter dois ou quatro dígitos.

1.6 Heurística 6: Seja consistente

A consistência é um dos princípios mais básicos de usabilidade!

Se os usuários sabem que o mesmo comando ou a mesma ação terá sempre o mesmo efeito, sentir-se-ão mais confiantes na utilização da aplicação, além de encorajados a explorar a interface, já que possuirão parte do conhecimento do qual precisam para operá-la.

A mesma informação deve ser apresentada na mesma localização em todas as telas e caixas de diálogo, além de estar formatada da mesma maneira, de modo a facilitar seu reconhecimento.

Na versão mais atual da família Microsoft Office, por exemplo, as opções para formatação de fontes estão sempre na guia Início.

Apesar de os aplicativos contarem com opções específicas, as opções comuns a todos eles devem estar no mesmo lugar, pois isso facilita o aprendizado da interface para o usuário.

Veja, a seguir, as telas do MS Excel e do MS Access. Nos diferentes aplicativos do pacote, o MS Office mantém as opções de formatação de fonte na guia Início.

Muitos aspectos relacionados à consistência são mais facilmente alcançados quando se segue um padrão no projeto da interface, pois o padrão é responsável por especificar muitos detalhes do diálogo, como, por exemplo, sobre que fonte utilizar nas diferentes telas.

Infelizmente, adotar um padrão não é suficiente para garantir a consistência, pois tais padrões deixam muitas brechas para os projetistas.

ATENÇÃO!

A consistência não se refere somente ao projeto de telas, mas inclui também considerações a respeito da tarefa e da estrutura da funcionalidade do sistema. Ainda que você seja um usuário frequente de interfaces de linha de comando, vai sentir diferença se precisar alternar entre o uso do prompt de comando do Windows e o modo terminal do Linux. Do ponto de vista do projeto de interface, as telas são bastante semelhantes, mas os sistemas operacionais por trás das mesmas são muito diferentes.

1.7 Heurística 7: Forneça feedback

O sistema deve informar continuamente ao usuário sobre o que está fazendo e sobre como está interpretando as entradas que lhe são fornecidas.

O feedback não deve acontecer somente quando um erro ou algo inesperado acontece. É necessário que sejam fornecidos também feedbacks positivos e feedbacks parciais à medida que as informações se tornam disponíveis.

O feedback dado pelo sistema não deve ser expresso de forma abstrata ou em termos gerais. O ideal é que ele reformule a entrada do usuário, de modo a informá-lo sobre o que está sendo feito com os dados fornecidos.

É uma boa ideia, por exemplo, que mensagens de alerta sejam exibidas quando o usuário está prestes a executar urna ação irreversível. Imagine que o usuário está prestes a copiar uma pasta para um destino em que já existe urna pasta com o mesmo nome. É importante que o usuário seja alertado sobre o problema e que tenha a opção de escolher o que fazer - se deseja cancelar a operação, substituir a pasta-destino ou mesclar o conteúdo de ambas.

Diferentes tipos de feedback podem demandar diferentes graus de persistência por parte da interface.

Alguns feedbacks são relevantes somente durante a existência de determinado **fenômeno** e apresentam, portanto, uma baixa persistência, desaparecendo quando não são mais necessários.

Outros feedbacks, entretanto, têm **persistência média** e permanecem na tela até que o usuário o faça desaparecer.

Há ainda alguns tipos de feedback tão importantes que requerem **alta persistência** e acabam tornando-se um item permanente na interface.

Fenômenos

Uma mensagem que alerte o usuário sobre a falta de papel na impressora deve desaparecer automaticamente assim que o problema for resolvido. Imagine que o usuário está prestes a copiar uma pasta para um destino em que já existe uma pasta com o mesmo nome. É importante que o usuário seja alertado sobre o problema e que

tenha a opção de escolher o que fazer - se deseja cancelar a operação, substituir a pasta-destino ou mesclar o conteúdo de ambas.

Persistência média

Um exemplo disso seria uma mensagem alertando o usuário sobre o redirecionamento de um arquivo para outra impressora por problemas na impressora escolhida - a mensagem somente seria fechada após o dique do usuário no botão Ok.

Alta persistência

Um exemplo disso pode ser a indicação do espaço livre no disco rígido em uma aplicação de gerenciamento de arquivos.

O feedback é especialmente importante quando a ação em curso possui um tempo de resposta maior que o tempo padrão.

Em geral, o tempo de resposta deve ser o mais rápido possível, mas pode ser que o computador demore a responder.

Nesse caso, é sempre bom **manter o usuário informado** sobre o estado da execução da tarefa solicitada, de modo que não pense, por exemplo, que o computador travou ou que não está executando o que lhe foi solicitado.

Manter o usuário informado

Durante a exclusão de muitos arquivos, por exemplo, o MS Windows exibe uma caixa de diálogo na qual mantém o usuário informado sobre o progresso da tarefa.

Também é preciso preocupar-se com o **feedback** em caso de falha no sistema.

Muitos sistemas simplesmente param de responder às solicitações do usuário sem dar qualquer informação de que se encontram indisponíveis. Infelizmente, não fornecer nenhum tipo de feedback é muito ruim, pois faz com que o usuário tenha de adivinhar o que está errado.

Feedback

É uma boa ideia, por exemplo, que mensagens de alerta sejam exibidas quando o usuário está prestes a executar urna ação irreversível. Imagine que o usuário está prestes a copiar uma pasta para um destino em que já existe urna pasta com o mesmo nome. É importante que o usuário seja alertado sobre o problema e que tenha a opção de escolher o que fazer - se deseja cancelar a operação, substituir a pasta-destino ou mesclar o conteúdo de ambas.

1.8 Heurística 8: Reduza a carga de memória do usuário

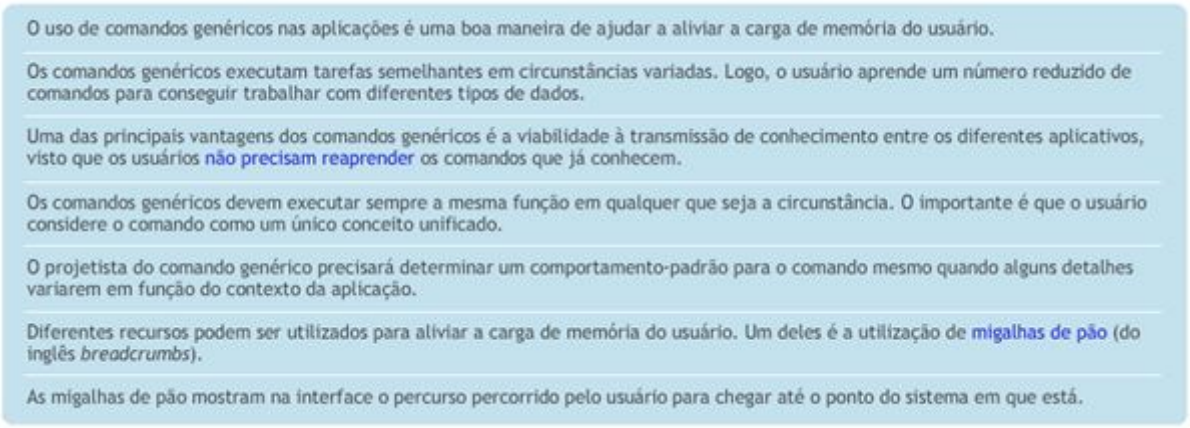
Os computadores devem aliviar a carga de **memória** dos usuários o máximo que puderem. As interfaces pautadas no reconhecimento baseiam-se na **visibilidade** de objetos que sejam de interesse do usuário.

Memória

Infelizmente, a exibição de muitos objetos em tela acaba resultando em uma poluição visual que vai de encontro ao que sugere a heurística: a criação de diálogos simples e naturais. Portanto, use o bom senso e mapeie a exibição de objetos com as necessidades pontuais do usuário.

Visibilidade

Em geral, as pessoas conseguem-se lembrar mais facilmente daquilo que lhes é mostrado do que de itens que estão em sua memória e que não contam com nenhuma dica visual para recuperação. Isso ocorre quando aprendemos uma língua estrangeira, por exemplo. É muito mais fácil recordar o vocabulário ensinado através de imagens do que aquele relacionado a conceitos abstratos. Sendo assim, os computadores devem exibir elementos visuais de diálogo com o usuário.



Não precisam reaprender

Observe o exemplo da opção imprimir: em diferentes aplicativos e com diferentes tipos de arquivo, seu comportamento é sempre o mesmo. Nos diferentes aplicativos do Windows, por exemplo, a opção colar terá sempre o mesmo comportamento: inserir o conteúdo da Área de Transferência no arquivo em que se está trabalhando.

Migalhas de pão

O interessante do recurso migalhas de pão é que as opções listadas no percurso do usuário funcionem como um atalho para retorno às páginas visitadas, visto que se trata de links para as mesmas.

1.9 Heurística 9: Exiba mensagens claras de erro

As situações de erro são graves para a usabilidade por dois motivos principais:

- Primeiro, porque tais situações representam circunstâncias nas quais o usuário está com problemas e, possivelmente, não conseguirá usar o sistema para executar a tarefa desejada.
- Segundo, porque tais situações representam oportunidades para ajudar o usuário a compreender melhor o sistema, visto que, no momento do problema, o usuário estará mais motivado a prestar atenção ao conteúdo das mensagens de erro.

As mensagens de erro devem seguir quatro regras básicas, quais sejam:

1-As mensagens de erro devem ser escritas de forma clara e devem evitar códigos pouco claros. Os usuários devem ser capazes de compreender a mensagem sozinhos, sem ter de consultar um manual. Pode ser necessário incluir códigos do sistema ou códigos que ajudem o responsável pelo sistema a rastrear o problema, mas essa informação deve vir acompanhada de uma mensagem que situe o usuário com relação ao ocorrido.

No caso de entradas textuais, métodos de correção de ortografia podem ser especialmente rápidos e precisos quando as possíveis entradas do usuário estão restritas a um conjunto definido de termos, tais como nome de arquivos e comandos.

Além de exibir boas mensagens de erros, as aplicações também devem ser capazes de se recuperar do erro ocorrido.

Por exemplo, os usuários devem ser capazes de desfazer o efeito de comandos errados e devem poder editar e revisar comandos anteriores sem precisar refazê-los desde o princípio.

2-As mensagens de erro devem ser precisas. Por exemplo, em vez de dizer simplesmente que não foi possível abrir um arquivo, o sistema deve informar o nome do arquivo cuja tentativa de abertura falhou.

No caso de entradas textuais, métodos de correção de ortografia podem ser especialmente rápidos e precisos quando as possíveis entradas do usuário estão restritas a um conjunto definido de termos, tais como nome de arquivos e comandos.

Além de exibir boas mensagens de erros, as aplicações também devem ser capazes de se recuperar do erro ocorrido.

Por exemplo, os usuários devem ser capazes de desfazer o efeito de comandos errados e devem poder editar e revisar comandos anteriores sem precisar refazê-los desde o princípio.

3-O sistema deve ajudar o usuário a resolver o problema. Se, por exemplo, um editor de textos não consegue abrir um arquivo por conta de uma incompatibilidade de formatos, a mensagem de erro pode avisar sobre o ocorrido e sugerir outra aplicação na qual o arquivo possa ser aberto (caso exista uma instalada no computador).

No caso de entradas textuais, métodos de correção de ortografia podem ser especialmente rápidos e precisos quando as possíveis entradas do usuário estão restritas a um conjunto definido de termos, tais como nome de arquivos e comandos.

Além de exibir boas mensagens de erros, as aplicações também devem ser capazes de se recuperar do erro ocorrido.

Por exemplo, os usuários devem ser capazes de desfazer o efeito de comandos errados e devem poder editar e revisar comandos anteriores sem precisar refazê-los desde o princípio.

4-As mensagens de erro devem ser **educadas** e não devem intimidar os usuários ou culpá-los, explicitamente, pelo ocorrido. Os usuários já se sentem suficientemente mal quando encontram erros. A aplicação não precisa pode piorar a situação, com mensagens do tipo O USUÁRIO EXECUTOU UMA OPERAÇÃO ILEGAL, em caixa alta, como se gritasse com o usuário. As mensagens devem evitar termos extremos como fatal ou ilegal. Em geral, as mensagens de erro podem ser construídas de modo a sugerir que o problema é do computador, já que a interface deveria ter sido projetada de modo a impedir que erros ocorressem.

Educadas

Um modo útil de gerar mensagens de erro construtivas é tentar adivinhar o que o usuário realmente queria dizer ou fazer.

No caso de entradas textuais, métodos de correção de ortografia podem ser especialmente rápidos e precisos quando as possíveis entradas do usuário estão restritas a um conjunto definido de termos, tais como nome de arquivos e comandos.

Além de exibir boas mensagens de erros, as aplicações também devem ser capazes de se recuperar do erro ocorrido.

Por exemplo, os usuários devem ser capazes de desfazer o efeito de comandos errados e devem poder editar e revisar comandos anteriores sem precisar refazê-los desde o princípio.

1.10 Heurística 10: Marque com clareza as saídas

Os usuários não gostam de se sentir encurralados pelo computador!

Para aumentar no usuário o sentimento de controle do diálogo com a aplicação, o sistema deve oferecer a ele maneiras simples de sair dos diferentes ambientes nos quais pode entrar através da interface.

É importante, por exemplo, que todas as aplicações para instalação de software – que, normalmente, é um processo demorado – disponibilizem um botão Cancelar ou um botão que permita ao usuário retornar à etapa anterior.

Outro recurso importante e no qual os usuários aprendem rapidamente a confiar é o de desfazer as ações. Quando possível e pertinente ao contexto da aplicação, é essencial que esse comando esteja disponível ao longo do sistema.

Quando as opções de cancelamento e de desfazer ações estão disponíveis, os usuários se sentem encorajados a explorar a aplicação e a experimentar novas tarefas, o que lhes ajuda a aprender mais sobre a interface com a qual estão interagindo.

Um princípio básico do projeto de interface é assumir que os usuários cometerão erros independentemente do que seja feito para melhorar a aplicação. É importante, portanto, viabilizar a recuperação desses erros e torná-la fácil.

O tempo de resposta do sistema deve ser o mais rápido possível!

Nos casos de a aplicação não conseguir concluir o processamento dentro do tempo-limite durante o qual o usuário consegue manter sua atenção (normalmente, 10 segundos), o usuário deve poder interromper o computador e cancelar a operação.

As interfaces devem sempre manter um alto poder de resposta, de modo que permaneçam atentas às novas ações do usuário e deem a elas prioridade maior do que concluir as ações em curso.

Os diferentes mecanismos de saída e cancelamento devem estar visíveis na interface e não devem depender da habilidade do usuário para lembrar-se de códigos especiais ou combinações de teclas.

A visibilidade é um princípio geral da usabilidade, mas ela é especialmente crucial para a situação de saída.

Os usuários precisam da visibilidade quando se percebem em território desconhecido, no qual podem sentir medo de causar danos ou perder dados caso executem alguma ação indevida.

O que vem na próxima aula

- Os critérios ergonômicos de avaliação de interfaces;
- Os problemas relacionados a esses critérios;
- Avaliação e proposição de soluções para problemas ergonômicos.

CONCLUSÃO

Nesta aula, você:

- Conheceu as heurísticas de usabilidade;
- Identificou problemas relacionados às heurísticas;
- Soube como propor soluções para problemas de usabilidade com base nas recomendações propostas pelas heurísticas.