Material de apoio 2 – Design de Interface e Usabilidade

Pense em design como criar algo para um propósito, ou seja, projetar algo para que realize uma tarefa específica. Quando nós voltamos especificamente para o design de interface estamos falando de construir o meio pelo qual uma pessoa irá dialogar com outra pessoa ou máquina. Tendo essa visão sobre o design de interface deixamos de lado aquele senso comum de ter o design como “embelezador de coisas”.

Uma interface mal projetada poderá gerar dúvidas no usuário de forma que ou ele ficará preocupado e focado em não cometer erros na utilização da aplicação ao invés de simplesmente utilizá-la para suprir suas necessidades ou ficará perdido e inseguro de forma que a experiência de uso o conduzirá a simplesmente não se aproximar mais daquele produto, interface ou sistema.

O papel do designer de interfaces do usuário (UI Designer) é projetar uma interface que não abra espaço para esses possíveis momentos de insegurança, deixar muito claro quais serão os resultados da interação com a interface (perceba que não estamos falando apenas do ambiente no qual o usuário está mas também de onde ele veio e para onde ele irá) e garantir que o usuário realize todas as ações e tarefas de forma simples e eficiente. Falando com outras palavras, o usuário “não precisará de um manual de instruções” e terá sua atenção totalmente voltada para realizar sua necessidade por meio daquela interface.

Ben Shneiderman, cientista da computação e professor do Laboratório de Interação Humano-Computador na Universidade de Maryland, College Park, desenvolveu uma lista chamada Eight Golden Rules of Interface Design que, como o nome já indica, apresenta oito regras essenciais para o Design de Interfaces as quais apresentarei a vocês a seguir.

8 regras de ouro para desenvolvimento de interfaces de Ben Shneiderman - Eight Golden Rules of Interface Design

#1 Mantenha a consistência: Utilizar ícones, cores, hierarquias, menus, call-to-actions e fluxos de uso similares em situações similares é essencial para construir uma boa interface e fornecer uma ótima experiência para o usuário. Quanto mais diferentes formas de interação maior a dificuldade que ele terá em utilizar uma interface por isso manter a consistência tem um papel fundamental, ela cria facilidade em assimilar aquele momento específico que o usuário está vivendo a outras situações previamente experienciadas, portanto, a decisão de qual ação tomar se torna muito mais simples e fácil.

#2 Permita que os usuários utilizem atalhos: Na medida que o nível de conhecimento que possuímos sobre uma interface cresce a necessidade de formas mais rápidas de interação para realizar uma tarefa começam a surgir. Um exemplo para isso são os famosos Ctrl+C e Crtl+V, atalhos que permitem ao usuário mais experiente realizar uma tarefa, que antes demandaria um tempo maior, em alguns poucos segundos.

#3 Responda o seu usuário quando ele perguntar: Como comentei anteriormente, criar uma interface é desenvolver um meio pelo qual irá ocorrer um diálogo entre um usuário e outro usuário ou uma máquina, portanto fornecer um feedback das ações do usuário é como responder a uma pergunta que ele tenha feito para a interface. O usuário precisar saber onde ele está, o que está acontecendo e para onde ele será direcionado após o término de sua ação, isso exige que o feedback apresentado a ele seja apropriado, ou seja, deve conter algum significado compreensível pelo usuário. Um exemplo simples para isso é a indicação de qual página ele está dentro de um questionário que possui dez páginas.

#4 Crie diálogos que indiquem o fim de uma ação: Dizer para o usuário que ele completou um ciclo é essencial. Permitir que o fluxo simplesmente acabe sem comunicar ao usuário causa a dúvida se a ação feita por ele foi a a ação correta. Não permita que seus usuários achem alguma coisa, dê a certeza de que a ação dele resultou em algo. Para exemplificar tenha em mente um fluxo de compra de um produto via e-commerce, ao final da compra é importante informar ao usuário que o item que ele desejava foi adquirido com sucesso.

#5 Mostre uma maneira de reparar um erro: As pessoas não gostam que digam a elas que estão erradas, principalmente seus usuários. Toda interface deve possuir mecanismos capazes de evitar o máximo possível que o usuários cometam erros (foolproof), mas quando erros impossíveis de serem evitados acontecem devemos apresentar ao usuário uma forma simples, passo-a-passo de como solucionar o erro ocorrido o mais rápido possível. Exemplo disso são os avisos que aparecem em formulários que nos indicam se precisamos corrigir algum campo de texto e qual deles corrigir.

#6 Deixe seu usuário reverter ações (famoso Ctrl+Z): Uma das principais funções que toda interface deve fornecer ao usuário é de permitir que ele retorne uma ação ou até mesmo um grupo de ações realizadas durante a utilização. A função Ctrl+Z remove a preocupação do usuário de cometer algum erro durante a navegação pois ele sabe que será possível reverter o resultado posteriormente.

#7 Dê ao usuário a sensação de controle: Criar a sensação para que o usuário se sinta no controle permite uma certa confiança entre ele e a interface. E isso se dá na medida que as transformações do ambiente ocorram da maneira como ele espera que ocorram.

#8 Reduza a carga de memória de curta duração: O ser humano possui uma “memória RAM” bem curta, o que quero dizer é que somos capazes de armazenar aproximadamente cinco informações diferentes de uma vez durante um período bem curto de tempo. Por conta disso toda interface deve possuir hierarquias muito bem desenvolvidas de forma que permita ao usuário encontrar as informações que procura de forma rápida sem a necessidade de ficar anotando dados de uma página para comparar com outros dados de outras páginas “perdidas” na aplicação.

Conclusão: Entendendo a importância de uma interface e alguns dos pontos que devem ser levados em conta ao criar uma, percebemos que eles de fato determinarão se a experiência será boa ou ruim. Não se trata de deixar algo mais bonito, mas de tornar a experiência de interação entre uma pessoa e uma máquina a mais natural e

tranquila possível. As oito regras de ouro de Ben Shneiderman são diretrizes essenciais para o design de interfaces, e sua aplicação pode melhorar significativamente a usabilidade de um software ou aplicativo. Vamos explorar cada uma delas, incluindo para que servem, quando utilizá-las e exemplos práticos.

Exemplos de utilização:

1. Consistência: Para que serve: A consistência ajuda os usuários a entenderem e preverem o comportamento da interface. Interfaces consistentes reduzem a curva de aprendizado.

Quando utilizar: Sempre que possível, especialmente em aplicativos grandes ou complexos.

Exemplo: Em um aplicativo de gerenciamento de tarefas, os ícones para "adicionar" e "remover" devem ser os mesmos em diferentes seções do aplicativo. Se o botão de "salvar" é verde em uma parte, deve ser verde em todas as partes.

2. Atalhos: Para que serve: Atalhos são essenciais para usuários experientes que desejam realizar tarefas rapidamente, aumentando a eficiência.

Quando utilizar: Quando um software é destinado a usuários que já possuem conhecimento prévio ou quando se deseja acelerar a interação em tarefas frequentes.

Exemplo: Em um editor de texto, permitir que usuários utilizem Ctrl+B para negrito ou Ctrl+S para salvar facilita a edição rápida, melhorando a produtividade.

3. Feedback: Para que serve: O feedback proporciona clareza sobre as ações realizadas, aumentando a confiança do usuário na interação com o sistema.

Quando utilizar: Sempre que uma ação é realizada, especialmente em operações críticas como enviar formulários ou excluir itens.

Exemplo: Após o envio de um formulário de contato, o usuário deve ver uma mensagem de confirmação, como "Seu formulário foi enviado com sucesso!" Isso ajuda a garantir que a ação foi concluída.

4. Prevenção de erros: Para que serve: Minimizar erros ajuda a reduzir frustrações e melhora a experiência do usuário.

Quando utilizar: Em qualquer situação onde o erro pode resultar em consequências significativas, como perda de dados ou ações irreversíveis.

Exemplo: Ao tentar excluir um arquivo, um diálogo de confirmação deve aparecer, perguntando "Você realmente deseja excluir este arquivo?". Isso ajuda a evitar exclusões acidentais.

5. Exploração: Para que serve: Permitir que os usuários explorem a interface livremente ajuda na descoberta de funcionalidades e aumenta a satisfação.

Quando utilizar: Em aplicações que têm múltiplas funcionalidades ou configurações, permitindo uma interação mais intuitiva.

Exemplo: Em um software de design gráfico, incluir um modo de pré-visualização onde o usuário pode experimentar diferentes efeitos sem alterar o projeto original permite que ele explore sem medo.

6. Controle do usuário: Para que serve: Oferecer controle ao usuário aumenta a sensação de autonomia e satisfação, levando a uma melhor experiência geral.

Quando utilizar: Sempre que a personalização ou a escolha do usuário for relevante, especialmente em softwares criativos ou interativos.

Exemplo: Em um aplicativo de música, permitir que o usuário crie e organize suas próprias playlists dá a ele controle sobre sua experiência auditiva.

7. Minimização da carga de memória: Para que serve: Reduzir a carga de memória ajuda os usuários a se concentrarem nas tarefas, em vez de se lembrarem de informações que podem ser apresentadas.

Quando utilizar: Em interfaces que exigem navegação frequente entre diferentes seções ou que possuem muitos recursos.

Exemplo: Em um software de gerenciamento de projetos, usar menus suspensos para exibir opções em vez de exigir que o usuário memorize cada uma delas reduz a carga mental.

8. Apoio à interação: Para que serve: Oferecer suporte e orientação ajuda os usuários a navegar pela interface e a aprender suas funcionalidades, tornando a experiência mais fluida.

Quando utilizar: Ao introduzir novos recursos ou quando o usuário pode estar menos familiarizado com o sistema.

Exemplo: Um novo software pode incluir um tour interativo que guia o usuário pelas principais funcionalidades, explicando o que cada botão faz.

10 Heurísticas de Nielsen

As 10 heurísticas de usabilidade de Jakob Nielsen são princípios gerais que ajudam a avaliar a usabilidade de interfaces. Aqui estão elas:

1. Visibilidade do status do sistema: O sistema deve sempre informar os usuários sobre o que está acontecendo, através de feedback apropriado.

2. Correspondência entre o sistema e o mundo real: A interface deve usar linguagem e conceitos familiares aos usuários, evitando jargões técnicos.

3. Controle e liberdade do usuário: Os usuários frequentemente escolhem funções por engano e precisam de uma maneira clara de desfazer e refazer ações.

4. Consistência e padrões: Os usuários não devem ter que se perguntar se diferentes palavras, situações ou ações significam a mesma coisa. Siga convenções de plataforma.

5. Prevenção de erros: Melhor do que boas mensagens de erro é um design cuidadoso que previne problemas antes que eles ocorram.

6. Reconhecimento em vez de memorização: Minimize a carga de memória do usuário tornando objetos, ações e opções visíveis.

7. Flexibilidade e eficiência de uso: A interface deve atender tanto usuários novatos quanto experientes, oferecendo atalhos e personalização.

8. Estética e design minimalista: Diálogos não devem conter informações irrelevantes ou raramente necessárias. O conteúdo deve ser conciso.

9. Ajuda aos usuários a reconhecer, diagnosticar e recuperar de erros: As mensagens de erro devem ser expressas em linguagem clara, indicando o problema e sugerindo uma solução.

10. Ajuda e documentação: Embora o ideal seja que o sistema possa ser usado sem documentação, pode ser necessário fornecer ajuda e documentação acessíveis.

Essas heurísticas são úteis para identificar problemas de usabilidade e melhorar a experiência do usuário.

Exemplo de utilização

Aqui estão exemplos de como as 10 heurísticas de Nielsen podem ser aplicadas em design de interface:

1. Visibilidade do status do sistema: Em um aplicativo de e-commerce, uma barra de carregamento informa ao usuário que o pagamento está sendo processado.

2. Correspondência entre o sistema e o mundo real: Um aplicativo de reservas de hotel usa termos como "check-in" e "check-out", em vez de termos técnicos, tornando a navegação mais intuitiva.

3. Controle e liberdade do usuário: Um editor de texto permite que os usuários desfaçam e refaçam ações com facilidade, usando comandos como Ctrl+Z e Ctrl+Y.

4. Consistência e padrões: Em um sistema operacional, os ícones de "configurações" e "ajuda" são sempre os mesmos, independentemente do aplicativo.

5. Prevenção de erros: Um formulário online destaca campos obrigatórios e fornece exemplos de preenchimento, evitando entradas inválidas.

6. Reconhecimento em vez de memorização: Um menu suspenso em um software de design gráfico apresenta opções de ferramentas, evitando que o usuário precise lembrar atalhos.

7. Flexibilidade e eficiência de uso: Um aplicativo de planilhas oferece tanto menus de opções quanto atalhos de teclado para usuários avançados.

8. Estética e design minimalista: Um site de notícias apresenta uma página inicial limpa, mostrando apenas os principais artigos e evitando distrações visuais.

9. Ajuda aos usuários a reconhecer, diagnosticar e recuperar de erros: Em um sistema de e-mail, se um envio falha, uma mensagem explica o problema e sugere ações, como "Verifique sua conexão com a internet".

10. Ajuda e documentação: Um software complexo de edição de vídeo inclui um tutorial interativo para guiar os novos usuários através das principais funcionalidades.

Esses exemplos ilustram como as heurísticas podem ser aplicadas para melhorar a usabilidade e a experiência do usuário em diferentes contextos.

Usabilidade

O que é Usabilidade? Usabilidade refere-se à facilidade com que um usuário pode interagir com um sistema, produto ou serviço. É um aspecto crucial do design de interfaces que determina a eficácia, eficiência e satisfação do usuário ao utilizar uma aplicação. Em termos práticos, usabilidade envolve a criação de produtos que são intuitivos e agradáveis de usar, permitindo que os usuários realizem suas tarefas de forma rápida e sem frustrações.

Para que Serve a Usabilidade?

A usabilidade serve para:

- Melhorar a Experiência do Usuário (UX): Uma boa usabilidade aumenta a satisfação do usuário, promovendo uma interação positiva com o produto.
- Reduzir Erros: Interfaces bem projetadas ajudam a minimizar a ocorrência de erros, facilitando a realização de tarefas.
- Aumentar a Eficiência: Usabilidade otimizada permite que os usuários completem suas tarefas mais rapidamente, economizando tempo.
- Aumentar a Adoção do Produto: Produtos com alta usabilidade são mais propensos a serem adotados, uma vez que os usuários percebem facilidade em utilizá-los.

Como Aplicar os Conceitos de Usabilidade?

Para aplicar os conceitos de usabilidade, é necessário seguir algumas etapas:

- Pesquisa de Usuário: Realizar entrevistas, questionários e observações para entender as necessidades e comportamentos dos usuários.
- Desenvolvimento de Personas: Criar perfis representativos dos usuários típicos, ajudando a guiar o design de acordo com suas expectativas.
- Prototipagem: Desenvolver protótipos de baixa e alta fidelidade para testar ideias e receber feedback antes do desenvolvimento final.

- Testes de Usabilidade: Conduzir testes com usuários reais para identificar problemas de usabilidade e áreas de melhoria.
- Iteração: Basear-se no feedback coletado para refinar e melhorar o design, repetindo o ciclo até alcançar um resultado satisfatório.

Ferramentas

Existem diversas ferramentas que podem ajudar a melhorar a usabilidade, incluindo:

- Figma: Uma plataforma de design colaborativo que permite a criação de protótipos interativos.
- Adobe XD: Software para design de interfaces e prototipagem, com recursos para testes de usabilidade.
- UsabilityHub: Um serviço que permite realizar testes rápidos para obter feedback sobre designs.
- Lookback: Uma ferramenta para realizar testes de usabilidade remotos, permitindo a gravação de sessões de usuários.
- Google Analytics: Embora não seja uma ferramenta de design, pode fornecer dados valiosos sobre o comportamento dos usuários, ajudando a identificar áreas problemáticas.

Exemplos de Utilização

- Aplicativos de E-commerce: Muitas plataformas de e-commerce investem em usabilidade para facilitar a navegação, a busca de produtos e o processo de checkout. Um exemplo é a Amazon, que utiliza feedback constante para otimizar a experiência do usuário.
- Softwares de Produtividade: Aplicativos como o Trello e o Asana implementam conceitos de usabilidade ao permitir que os usuários organizem suas tarefas de forma intuitiva, utilizando arrastar e soltar.
- Sites de Informação: Websites de notícias, como o G1, utilizam uma hierarquia clara de informações e navegação simples, garantindo que os leitores encontrem rapidamente o que buscam.
- Sistemas de Gestão: Sistemas ERP (Enterprise Resource Planning), como o SAP, investem em usabilidade para garantir que usuários possam acessar informações complexas de maneira eficiente e sem confusões.
- Jogos Digitais: Jogos como "The Legend of Zelda: Breath of the Wild" utilizam tutoriais e dicas que se adaptam ao nível de habilidade do jogador, melhorando a experiência e a usabilidade.

A usabilidade é um componente essencial no design de interfaces que impacta diretamente a satisfação e a eficiência do usuário. Ao aplicar conceitos de usabilidade de forma estruturada e utilizando as ferramentas adequadas, é possível criar produtos que não apenas atendem, mas superam as expectativas dos usuários, resultando em maior adoção e sucesso no mercado.