

1 - Visibilidade do status do sistema. Ex: Ação concluída com sucesso? Feedback.

2 - Correspondência entre o sistema e o mundo real. Ex: Ícone de carrinho de compras é o carrinho de compras do app. Relação real/digital.

3 - Controle do usuário e liberdade. Ex: Poder desfazer ações, liberdade de clicar onde bem entenderem sem serem forçados a algum caminho específico indesejado.

4 - Consistência e padrões. Ex: O botão “confirmar”, em todos os lugares sempre terá a mesma função esperada. O ícone “carrinho” sempre será o carrinho de compras.

5 - Prevenção de erros. Ex: Evitar que mande um formulário com informações erradas, como um registro de login que só aceita datas de aniversário DIA/MÊS/ANO.

6 - Reconhecimento ao invés de memorização. Ex: Usuários reconhecem que o botão “login” faz o login. Não devem memorizar as informações, mas reconhecer.

7 - Flexibilidade e eficiência de uso. Ex: Sistema fácil de usar por novato ou veterano. No caso da interface que não precisa de muito tempo de aprendizado, como o Google. Tem uma aba e você pesquisa nela.

8 - Estética e design minimalista. Ex: Design com detalhes sem exagero, algo bonito que não desfoca da funcionalidade.

9 - Ajuda os usuários a reconhecer, diagnosticar e corrigir erros. Ex: Mensagens de erro informativas. login incorreto? mensagem é “login e/ou senha incorreto/s”

10 - Ajuda e documentação. Ex: Caso necessário o botão de ajuda sempre deve estar disponível. Acesso fácil à documentação.

- Visibilidade do status do sistema: O sistema deve manter os usuários informados sobre o que está acontecendo, fornecendo feedback apropriado em tempo hábil. Isso significa que os usuários devem saber o estado atual do sistema, como se uma ação foi concluída com sucesso, se uma tarefa está em andamento ou se ocorreu um erro. Feedback claro e imediato ajuda os usuários a entender e confiar no sistema.
- Correspondência entre o sistema e o mundo real: A linguagem e os conceitos usados na interface devem ser familiares e se alinhar com o mundo real dos usuários. Isso significa que os termos, ícones e metáforas usados na interface devem refletir as expectativas dos usuários com base em sua experiência no mundo real. Isso facilita a compreensão e a interação.
- Controle do usuário e liberdade: Os usuários devem ter a capacidade de desfazer ações indesejadas e navegar pelo sistema com facilidade. Isso evita erros irreversíveis e permite que os usuários escolham seu próprio caminho, sem ficarem presos em ações que não desejam executar.
- Consistência e padrões: Elementos da interface, como botões, ícones e menus, devem ser consistentes em todo o sistema e seguir padrões amplamente reconhecidos. Isso torna a interface previsível e familiar, permitindo que os usuários apliquem o conhecimento adquirido em uma parte do sistema em outras partes.
- Prevenção de erros: O sistema deve ser projetado de forma a prevenir erros comuns. Isso pode ser feito por meio de confirmações, restrições inteligentes, validações de entrada e feedback claro que ajude os usuários a evitar erros.
- Reconhecimento ao invés de memorização: Os usuários não devem ser forçados a memorizar informações desnecessárias. Em vez disso, a interface deve apresentar opções e ações claramente para que os usuários possam reconhecê-las facilmente, em vez de depender da memória.
- Flexibilidade e eficiência de uso: O sistema deve ser eficiente tanto para usuários novatos quanto para especialistas. Deve ser possível realizar tarefas com rapidez e eficiência após uma curta curva de aprendizado. Isso implica em oferecer atalhos, opções avançadas e recursos que atendam às necessidades de diferentes usuários.
- Estética e design minimalista: A interface deve ser esteticamente agradável, mas não deve conter elementos de design desnecessários que possam distrair ou confundir os usuários. O design deve ser limpo, organizado e focado nas tarefas e informações essenciais.
- Ajuda os usuários a reconhecer, diagnosticar e corrigir erros: Quando os usuários cometem erros, o sistema deve fornecer mensagens de erro claras e orientações sobre como corrigi-los. As mensagens de erro devem ser informativas e indicar as etapas necessárias para resolver o problema.
- Ajuda e documentação: Caso os usuários precisem de assistência, deve ser fácil acessar a documentação ou a ajuda. No entanto, o sistema deve ser projetado de forma a minimizar a necessidade de consultar essa ajuda. A documentação deve ser clara e estar disponível quando necessário.