



DESIGN DE INTERFACES

Felipe Orsini Martinelli

DESIGN DE INTERFACES

1^a edição

São Paulo
Platos Soluções Educacionais S.A
2022

© 2022 por Platos Soluções Educacionais S.A.

Todos os direitos reservados. Nenhuma parte desta publicação poderá ser reproduzida ou transmitida de qualquer modo ou por qualquer outro meio, eletrônico ou mecânico, incluindo fotocópia, gravação ou qualquer outro tipo de sistema de armazenamento e transmissão de informação, sem prévia autorização, por escrito, da Platos Soluções Educacionais S.A.

Head de Platos Soluções Educacionais S.A

Silvia Rodrigues Cima Bizatto

Conselho Acadêmico

Alessandra Cristina Fahl

Camila Turchetti Bacan Gabiatti

Camila Braga de Oliveira Higa

Giani Vendramel de Oliveira

Gislaine Denisale Ferreira

Henrique Salustiano Silva

Mariana Gerardi Mello

Nirse Ruscheinsky Breternitz

Priscila Pereira Silva

Tayra Carolina Nascimento Aleixo

Coordenador

Henrique Salustiano Silva

Revisor

Gabrielly Del Carlo Richene

Editorial

Beatriz Meloni Montefusco

Carolina Yaly

Mariana de Campos Barroso

Paola Andressa Machado Leal

Dados Internacionais de Catalogação na Publicação (CIP)

M385d Martinelli, Felipe Orsini

Design de interfaces / Felipe Orsini Martinelli. – São Paulo:

Platos Soluções Educacionais S.A., 2022.

32 p.

ISBN 978-65-5356-174-8

1. Design. 2. Experiência do usuário. 3. Interface.

I. Título.

CDD 005

Evelyn Moraes – CRB 010289/O

2022

Platos Soluções Educacionais S.A
Alameda Santos, nº 960 – Cerqueira César
CEP: 01418-002— São Paulo — SP
Homepage: <https://www.platosedu.com.br/>

DESIGN DE INTERFACES

SUMÁRIO

Apresentação da disciplina	05
O Design de Interface e a Experiência do Usuário	07
Técnicas para Ideação e Identificação de Problemas	21
Prototipagem de Interfaces	34
Avaliações de Usabilidade	46

Apresentação da disciplina

Nesta disciplina, abordaremos as etapas necessárias para a concepção de uma interface gráfica, com o objetivo de compreender os principais conceitos que permeiam a área do design de interfaces. Muitas pessoas pensam que é um processo simples ou que envolve apenas o pessoal da área de computação. Criar softwares envolve muitos conceitos de Design, Psicologia, entre outras áreas.

Dito isso, no tema 1, começaremos lidando com os princípios de interface e interação, além de vermos a respeito dos campos de UX e UI, que são de extrema relevância na construção de projetos digitais. Aqui, podemos citar também conceitos de uso de cor e fontes tipográficas, e como isso interfere na interação do usuário com o sistema. Depois, passaremos para o campo do *design thinking*, que é uma abordagem muito útil para resolver problemas que envolvem a criação de aplicações digitais que requerem uma boa interface para realização de tarefas. Além disso, em todos os quatro temas, realizaremos atividades práticas de elaboração de projetos por meio da ferramenta FIGMA, muito utilizada no mercado de UX e UI.

No tema 2, partiremos para as técnicas de identificação de problemas e como nos organizar para resolvê-los. Abordaremos entrevistas, questionários, mapas mentais, entre outras técnicas que são utilizadas no mercado para isso. Entraremos também em algumas técnicas vindas da arquitetura da informação, que auxiliam, e muito, na organização de projetos. Veremos que é necessário muito planejamento antes de se criar um aplicativo, um software ou um site, porém, não é algo impossível.

No tema 3, veremos que, a partir dos dados levantados e organizados em um projeto, podemos dar início à criação da interface, começando pela criação de *wireframes* e prototipagem, onde você poderá aprender sobre várias técnicas e possibilidades de aplicação. Além disso, veremos que a prototipagem não é apenas criar uma tela visualmente agradável, mas também pensar na navegação que o usuário realizará para cumprir uma tarefa. Tudo isso acontecerá antes do produto ser, de fato, desenvolvido.

Por fim, no tema 4, veremos o que fazer depois do desenvolvimento, quando o produto está finalizado e precisamos testá-lo com o público. Neste ponto, as técnicas de avaliação de usabilidade ajudam a encontrar problemas que não tinham como ser detectados nas etapas anteriores.

Que seja uma disciplina muito proveitosa e que você consiga aplicar os conceitos em projetos reais.

Bons estudos!

O Design de Interface e a Experiência do Usuário

Autoria: Felipe Orsini Martinelli

Leitura crítica: Gabrielly Del Carlo Richene



Objetivos

- Compreender os princípios do design de interface para aplicações digitais.
- Analisar os principais pontos de usabilidade em uma interface.
- Assimilar a relação entre o design thinking e o design de interfaces.



1. Introdução

Usamos tecnologias da informação, como computadores e celulares, em diversas tarefas do nosso dia a dia. Tal uso implica em interagirmos com sistemas que foram programados, porém, não vemos tal código, mas interfaces gráficas. O termo interface é de uso cada vez mais comum na sociedade, principalmente, após a popularização dos smartphones. Além disso, para criarmos uma interface, há diversas etapas e técnicas que devem ser levadas em consideração.

Neste material, discutiremos sobre a definição de interface. Depois, veremos conceitos referentes aos tipos de interação, passando pelos conceitos de UX e UI, que servirão de ponto de partida para comentarmos a respeito do que é uma boa interação e o que fazer para conseguir em um projeto. Falaremos também sobre usabilidade e abordaremos os tópicos do uso de cores e fontes tipográficas em uma interface. Depois, abordaremos o conceito de *design thinking* e sua relação com projetos de interface. Por fim, abordaremos o Figma, ferramenta muito utilizada na criação de protótipos de interface.



2. O que é a interface?

O primeiro conceito a ser trabalhado é o de interface. Para isso, é importante entender dois pontos fundamentais. O primeiro deles é de que o termo interface remete a qualquer interação, ou intermédio entre dois elementos, porém, consideraremos o conceito dentro do campo das tecnologias digitais, em que se trata do meio por onde acontecerá a interação entre máquina e usuário. Isso significa todos os meios por onde passa informação. Assim, o teclado, o mouse, a caixa de som, a tela, o aplicativo, tudo faz parte da interface. Entre todos esses elementos, o que mais se destaca, novamente com a popularização dos smartphones, é a tela, que, hoje, possui sensibilidade a toques. Todos

os botões que vemos na tela, menus, links, aplicativos, fazem parte da chamada Interface Gráfica do Usuário, do inglês *Graphic User Interface*, ou GUI.

Um segundo ponto importante é que os computadores lidam com uma linguagem que nós, humanos, não compreendemos, a chamada linguagem de máquina, que trabalha com pulsos de energia (que podem ser exemplificados como zeros e uns). Enquanto isso, os computadores não compreendem a linguagem humana ou linguagem natural. O problema que isso traz é que os computadores funcionam com base em linguagem de programação, que não chegam ao usuário comum, apenas seu resultado visual.

Em outras palavras, podemos colocar a interface como um tradutor, que trabalhará para que as duas partes (homem e máquina) se entendam (JOHNSON, 2001) e, dessa maneira, o usuário possa enviar mensagens em aplicativos, compartilhar conteúdos em redes sociais, entre outras ações do dia a dia. Isso mostra a importância que a interface possui, merecendo grande atenção em sua elaboração por parte de profissionais de várias áreas do conhecimento, como Design, Psicologia, Computação, entre outras.

O campo do design de interfaces existe para que a interação seja feita de maneira muito fácil: não precisamos pensar em linguagens de programação, nem no funcionamento do hardware de um celular, para que possamos enviar uma mensagem de bom dia para alguém. Isso é muito diferente do que acontecia décadas atrás, quando era necessário escrever um código na tela de um computador para abrir um programa, para apenas depois conseguir enviar uma mensagem, torcendo para que o e-mail chegasse sem problemas. Assim, tal facilidade é recente no meio digital.

Quando os computadores surgiram, e eram grandes e extremamente lentos, a interação era bem mais complexa. Tivemos várias mudanças

significativas no uso dos computadores, como o surgimento das telas, do mouse e de softwares cada vez mais intuitivos, o que permitiu com que as ações comuns sejam feitas sem a necessidade de treino, principalmente, com a relação que se criou do espaço digital com a organização da informação (JOHNSON, 2001).

A partir disso, chegamos ao design de interfaces, que abrange a criação de todos os aspectos visuais e de navegação em um aplicativo, para que seja fácil e intuitivo ao usuário. É fundamental que a interface seja produzida tendo em mente seu utilizador e não quem produziu. Isso porque, conforme aponta Norman (2006, p. 11), “o design na verdade é um ato de comunicação, o que significa ter um profundo conhecimento e compreensão da pessoa com quem o design está se comunicando.”



3. UX e UI

Conforme o que foi exposto, a interface tem um papel de grande importância no uso de produtos digitais. Entre as várias áreas que circulam neste campo, podem ser citados, principalmente, os campos de design da Interface de Usuário, conhecido como UI (*User Interface*) e da Experiência do Usuário, conhecido como UX (*User Experience*). Enquanto a UI trabalha com a construção das interfaces, considerando princípios de design gráfico e interação, a UX lida com métodos para construir uma melhor experiência de uso. É importante destacar que o campo da UX é extremamente amplo, englobando áreas diversas, como: design visual, arquitetura da informação, design de interação, interação humano-computador, psicologia, usabilidade, entre outras. Além disso, são inúmeras possibilidades de atuação para tal profissional: arquitetura da informação, usabilidade, taxonomia, estratégias de design, pesquisa com usuários, entre outras (TEIXEIRA, 2014).

Experiências de usuário existem desde quando o ser humano começou a usar ferramentas para a execução de alguma tarefa. Podem ser consideradas, aqui, tarefas tanto funcionais, como acessar um aplicativo de banco, quanto emocionais, como acessar uma rede social (TEIXEIRA, 2014). É importante ressaltar que a experiência é subjetiva, o que quer dizer que cada usuário terá uma experiência diferente na realização de uma mesma tarefa. Conforme aponta Teixeira (2014), tomando o uso de um caixa eletrônico como exemplo, as experiências são impactadas por fatores humanos (a capacidade de entender o texto que está na tela, por exemplo) e fatores externos (o horário do dia ou o fato de haver uma fila de pessoas atrás de você).

3.1 Fundamentos para uma boa interação

Há diversos fundamentos que podem ser listados na construção de uma boa experiência no uso de interfaces. Podemos começar discutindo os princípios de interação propostos pelo professor Donald Norman, um dos nomes mais importantes da área. Para ele, podemos abordar alguns fundamentos fundamentais a serem verificados no uso de qualquer produto, não apenas os digitais. O primeiro deles é a visibilidade, que indica que “as peças corretas têm de estar visíveis e devem transmitir a mensagem correta” (NORMAN, 2006, p. 28).

Em uma interface gráfica, os botões precisam ser facilmente encontrados pelo usuário, seguindo uma prioridade visual (qual botão deve ter mais destaque, qual botão deve ser colocado primeiro), conforme a navegação pretendida. Assim, um botão que se espera que seja muito utilizado, deve ter prioridade na tela. Um exemplo disso é o destaque (tamanho ou cor diferentes) que o botão de enviar um e-mail possui nos principais aplicativos e sites de e-mail.

Depois, podemos falar do princípio de mapeamento, que é a relação entre controles, botões e interfaces e qual o resultado de seu uso.

Este conceito é fortemente relacionado com a visibilidade. Se há uma boa visibilidade, ocorre um bom mapeamento e o produto é usado sem maiores problemas. Quando o uso é intuitivo, é dado o nome de mapeamento natural. Norman (2006) dá um bom exemplo ao abordar uma tesoura: o formato do objeto, a posição e tamanho dos espaços para inserir os dedos, tudo é extremamente intuitivo, que se comprehende o funcionamento do objeto apenas pelo seu formato.

O terceiro princípio é o de feedback ou retorno. Trata de receber uma informação com base em algo que foi feito. Se não há feedback, é comum que uma ação seja repetida, ocasionando em efeitos indesejados, como abrir um software duas vezes ou reiniciar um computador que parece ter travado, perdendo informação que não foi salva (NORMAN, 2006). Um feedback pode ser feito de uma maneira visualmente simples, desde mudar a cor de um botão para indicar que foi clicado, até ações mais elaboradas, como uma barra de carregamento que é parcialmente preenchida conforme um site é carregado, ou uma mensagem de confirmação quando um formulário é preenchido.

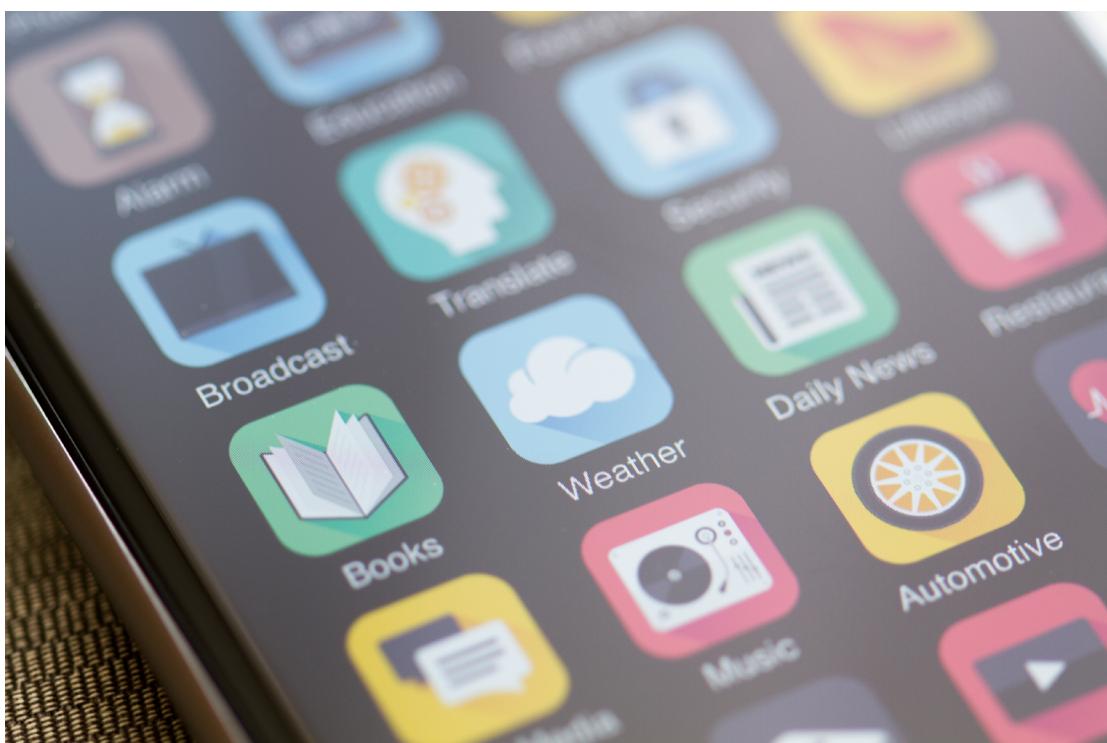
O quarto conceito é o de restrições. Uma interface deve ser restrita em alguns pontos, de modo que o usuário possa fazer apenas aquilo que esteja relacionado às funções do produto (NORMAN, 2006). Um exemplo é o botão de instalar um software, o qual só pode ser clicado depois que foi selecionada a opção de aceitar os termos de uso. As restrições ajudam, e muito, a evitar a frustração do usuário, já que impedem que erros sejam cometidos no uso de um produto.

Por fim, mas não menos importante, há o conceito de *affordance*, que remete às propriedades de um objeto que mostram como deve ser usado (NORMAN, 2006). Um exemplo simples disso é uma maçaneta, que dá uma ideia de girar ou ser puxada, ou o ícone para abrir a câmera do celular, que costuma ter o desenho de uma máquina fotográfica, ou algo parecido. Por conta do *affordance*, ao vermos um ícone, sabemos intuitivamente para que aquele aplicativo serve. Outro exemplo comum

são os botões em um site, que trazem uma ideia visual de que podem ser pressionados. Não há a necessidade de explicar que isso é um botão clicável.

Pode-se dizer que a ideia de *affordance* está intimamente ligada ao conceito de metáforas visuais, que remetem ao uso de ícones e funções no computador de modo similar ao mundo real (JOHNSON, 2001). Ao olhar o desktop, ou área de trabalho, é possível ver uma lixeira, acumulada com arquivos excluídos, sendo esvaziada de tempos em tempos, do mesmo modo que uma lixeira do mundo real. A organização de arquivos em pastas também é uma associação muito importante com o mundo real. Conforme aponta Johnson (2001), tais metáforas tiveram papel de grande importância na popularização da tecnologia. A Figura 1 mostra alguns ícones em uma tela de celular. É possível saber a funcionalidade de vários aplicativos apenas pelo seu ícone, conforme os conceitos de *affordance* e metáforas de interface.

Figura 1 – Exemplo de ícones em uma interface gráfica



Fonte: denkcreative/ iStock.com.

No exemplo da Figura 1, entre os vários aplicativos exibidos, podemos, por exemplo, observar dois deles: um aplicativo de previsão do tempo, intitulado *Weather*, cujo ícone é uma nuvem e, logo abaixo, um aplicativo de notícias, com o título *Daily News*, cujo ícone apresenta uma ilustração de um jornal. A relação é compreendida de maneira direta: se o ícone possui um elemento de nuvem ou sol, é grande a probabilidade de que lidará com a previsão do tempo, assim como um ícone que ilustra um jornal traz uma ideia de que lidará com notícias. Um ícone mal pensado pode levar a uma grande frustração do usuário, ao clicar em um ícone que remete a uma ideia e seria um aplicativo com outra funcionalidade.

3.2 Usabilidade

Um outro ponto importante, na interação, é a usabilidade. De maneira resumida, pode ser definida como a área que visa criar produtos que sejam fáceis de usar. Uma boa usabilidade indica cinco pontos em um produto, conforme aponta Nielsen (1993):

- Aprendizagem: o produto deve ser fácil de aprender a usar. Quanto mais intuitiva for uma função, mais fácil de aprender. Podemos relacionar este conceito com os princípios de visibilidade e mapeamento, apontados anteriormente.
- Eficiência: deve permitir um uso eficiente com alta produtividade na execução de tarefas. Dessa forma, o usuário não deve perder tempo tentando entender as funcionalidades, ou realizar várias ações para executar uma tarefa que pode ser feita com poucos passos.
- Memoriabilidade (ou reminiscência): as funções são fáceis de memorizar para usá-las novamente, ou seja, quando uma função é fácil de se executar, é facilmente memorizada.

- Evitar erros: o produto deve ajudar o usuário a cometer a menor quantidade de erros possíveis. Esse conceito pode ser relacionado ao princípio de restrições, discutido no tópico anterior.
- Satisfação: o uso deve ser agradável, o que traz satisfação ao usuário. Tal ideia é simples: um usuário tenderá a não usar mais um produto se não for agradável de ser usado.



4. Cores e tipos

Para otimizar a navegação em interfaces, existem vários fatores que impactam diretamente no processo, como as cores e as fontes tipográficas utilizadas. As cores podem direcionar o olhar por uma tela, tendo como funções “separar áreas visuais, organizar o conteúdo textual, promover a legibilidade, destacar itens, orientar a interação, entre outros” (SOBRAL, 2019, p. 43). Dessa forma, as cores podem ser associadas a ideias positivas ou negativas na navegação, além de interferir na legibilidade e causar cansaço quando utilizadas de qualquer jeito. Um ponto importante a ser considerado é que em telas as cores ficam mais luminosas, “o que normalmente dificulta a leitura caso o contraste com o fundo não se mostre suficiente ou seja saturado e intenso demais” (GUERRA; TERCE, 2019, p. 28).

Um ponto importante a ser considerado no uso de cores é que possuem um aspecto psicológico nas ideias que transmitem, que é da ordem cultural e subjetiva. Tomando como exemplo a cor branca, muito utilizada para o fundo de sites para web, pode trazer uma ideia de paz em algumas culturas e de morte em outras (LUPTON; PHILLIPS, 2008, p. 71). Tal exemplo demonstra a importância de levar em conta a cultura na qual o público-alvo de um produto está inserida.

Outro fator relevante para interfaces é o uso de fontes tipográficas, que traz o conceito de legibilidade ou a facilidade com que um texto

é lido. Esse conceito está intimamente ligado às propriedades de um texto, como o espaçamento entre letras e entre linhas, a largura dos caracteres, o tipo de fonte utilizado, entre outros (SOBRAL, 2019). Neste ponto, é importante compreender que algumas fontes possuem uma aparência mais sóbria, enquanto outras possuem uma aparência mais informal. Isso interfere diretamente em seu uso. Imagine um site de uma empresa de seguros, que utiliza a fonte *Comic Sans*, ou um site de brinquedos infantis, que utilize a *Times New Roman*. O Quadro 1 exemplifica tal diferença.

Quadro 1 – Exemplo de usos inadequados de fontes tipográficas

Fonte tipográfica	Texto 1	Texto 2
Times New Roman	Seguro de vida	Brinquedos
Arial	Seguro de vida	Brinquedos
<i>Comic Sans MS</i>	<i>Seguro de vida</i>	<i>Brinquedos</i>

Fonte: elaborado pelo autor.

Pela imagem, é possível perceber uma grande diferença do mesmo texto apresentado em fontes diferentes. Por isso, deve-se sempre utilizar fontes que estejam de acordo com a proposta. Outro ponto a ser considerado é que, entre as várias maneiras em que podemos classificar fontes, uma das principais se refere ao uso de serifas ou não, que são os prolongamentos que as letras possuem em suas hastes. No Quadro 1, anteriormente apontado, a fonte *Times New Roman* possui serifas, enquanto a fonte Arial, não. O uso, ou não, de fontes serifadas depende da situação: fontes serifadas facilitam a leitura em meios impressos, porém, podem dificultar a leitura em telas (SOBRAL, 2019).

5. Design thinking

No processo de elaboração de um projeto, independente da área situada, é importante planejar o que será feito. A falta de planejamento pode tornar um projeto demorado, aumentando o seu custo de

produção. Aqui, diversas analogias podem ser feitas. Por exemplo, imagine construir uma casa empilhando tijolos sem saber qual deveria ser o tamanho das paredes, ou pintar um quadro sem ter feito esboços, estudos de cor, tinta e texturas. O mesmo se passa com a criação de interfaces, em que a posição dos botões, as ações que serão disponibilizadas ao usuário. É preciso haver planejamento para otimização do processo.

Tal planejamento pode ser feito seguindo várias abordagens. Entre elas, pode ser citado o *design thinking*, que traz para diversas áreas o que designers já faziam ao juntar as necessidades humanas com os recursos disponíveis para a solução do problema junto ao processo de viabilização econômica de um projeto (BROWN, 2017). Em outras palavras, trata-se de lidar com a solução de um problema, colocando o ser humano como peça central da solução, criando uma experiência agradável em um serviço.

O *design thinking* está intimamente relacionado com a experiência de usuário, principalmente, se considerarmos seu foco no ser humano na produção de um projeto. Traz conceitos interessantes para pensarmos na construção de uma interface. Brown (2017) aponta três elementos complementares, que devem estar presentes em qualquer projeto de design. O insight, que é a ação de sair pelo mundo para observar as pessoas e tentar compreender suas necessidades. Depois, há a observação, o que envolve observar clientes fora do padrão, como alguém que coleciona bonecas ao invés de alguém que compra apenas uma. No entanto, não adianta ter levantado vários dados se não há o terceiro fator, que é a empatia, ou seja, se colocar no lugar do usuário. Em outras palavras, criar “conexões com as pessoas que estamos observando” (BROWN, 2017, p. 55).

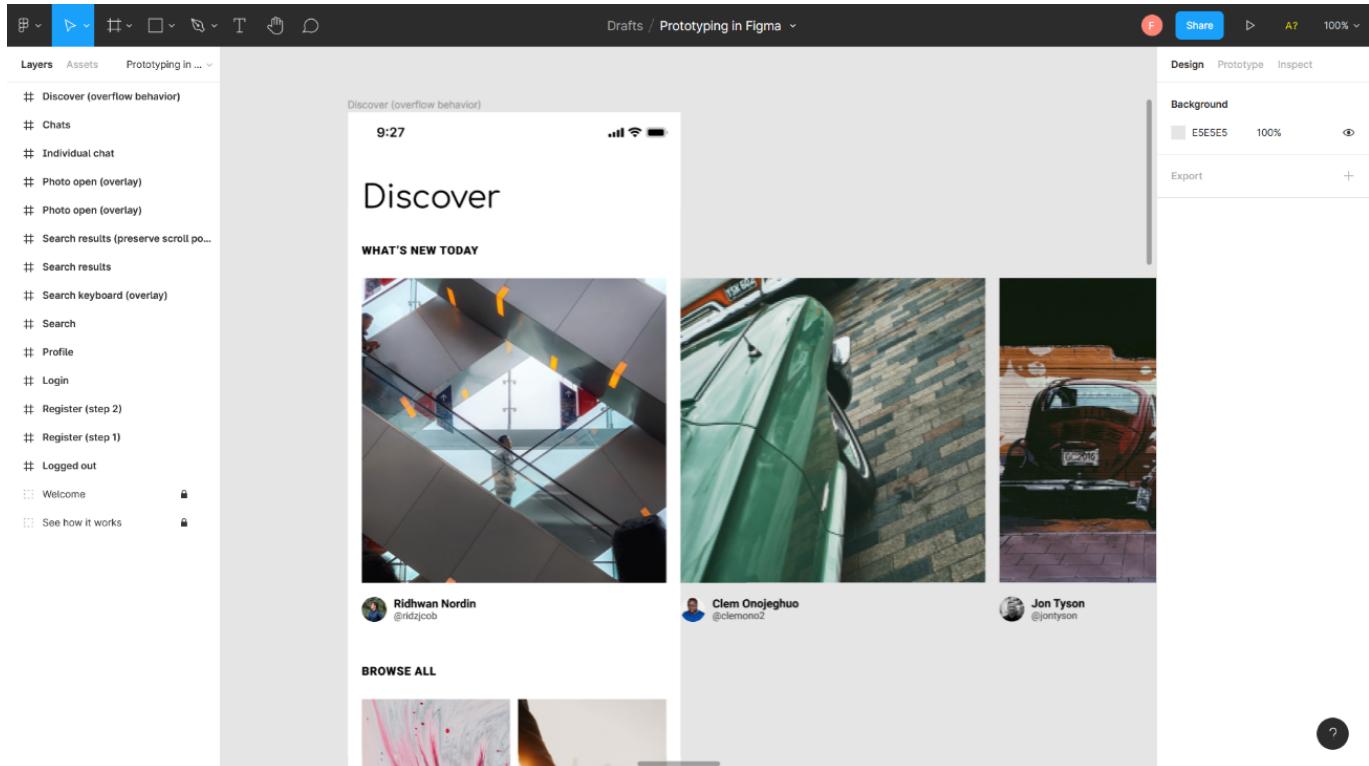
Além disso, dois pontos importantes trazidos por essa abordagem são a experimentação e a prototipagem, que estão fortemente ligados e ajudam muito no processo de elaboração de interfaces gráficas. Pela

experimentação, se entende a abertura a novas possibilidades em um projeto e novas soluções para um problema. Já a prototipagem, é a criação de modelos rápidos e baratos para uma solução. Ambos os conceitos servem para tirar ideias do papel e verificar se são factíveis ou não na prática.

6. Usando o FIGMA

Há inúmeras ferramentas gráficas que podem ser usadas na construção de interfaces, desde esquemas em papel, durante o planejamento, até protótipos de alta fidelidade em softwares específicos para tal finalidade. Entre esses softwares, um que possui grande destaque no mercado, é o FIGMA. É uma ferramenta que pode ser usada de modo on-line (pelo navegador, sem precisar instalar nada) ou offline (baixando um aplicativo, disponível para Windows, Mac e Linux). Além disso, a aplicação também está disponível para celular, o qual espelha o projeto que está no computador, sendo de grande utilidade no teste dos projetos para dispositivos móveis. É importante também considerar que o FIGMA possui vários recursos para se trabalhar de maneira compartilhada dentro de um mesmo projeto. Por fim, possui uma versão gratuita e três modelos pagos por meio de assinaturas, conforme as necessidades de uma equipe: plano *Professional*, *Organization* e *Enterprise*. A Figura 2 mostra a tela inicial de um modelo para prototipagem no Figma, que foi aberto na versão para navegador.

Figura 2 – Interface da versão web do aplicativo Figma



Fonte: print de tela do sistema operacional Windows 10.

A ferramenta é muito útil na criação, tanto de modelos gráficos, pela função Design, quanto na criação de fluxos de navegação, o que é feito pela função de prototipagem, ou Prototype. Dessa forma, é possível aplicar todos os conceitos que foram vistos nos tópicos anteriores na criação de projetos de interface e protótipos digitais.

Referências

- BROWN, T. **Design Thinking**. Rio de Janeiro: Alta Books, 2017.
- GUERRA, F.; TERCE, M. **Design Digital**: conceitos e aplicações para websites, animações, vídeos e webgames. São Paulo: Senac, 2019.
- JOHNSON, S. **Cultura da interface**: como o computador transforma nossa maneira de criar e comunicar. Rio de Janeiro: Zahar, 2001.
- LUPTON, E.; PHILLIPS, J. C. **Novos fundamentos do Design**. São Paulo: Cosac Naify, 2008.
- NIELSEN, J. **Usability Engineering**. Burlington: Morgan Kaufmann, 1993.

NORMAN, D. **O Design do dia a dia.** Rio de Janeiro: Rocco, 2006.

SOBRAL, W. **Design de interfaces.** São Paulo: Érica, 2019.

TEIXEIRA, F. **Introdução e Boas Práticas em Ux Design.** São Paulo: Casa do Código, 2014.

Técnicas para Ideação e Identificação de Problemas

Autoria: Felipe Orsini Martinelli

Leitura crítica: Gabrielly Del Carlo Richene



Objetivos

- Analisar as principais técnicas para levantamento de dados para projetos de interface.
- Compreender os principais conceitos que abrangem a identificação de problemas.
- Assimilar a relação entre o design de interfaces e as etapas de elaboração.



1. Introdução

A criação de interfaces é um processo que passa por várias etapas. Dentro delas, temos várias técnicas disponíveis, tendo em vista as necessidades dos usuários, requisitos para uma navegação intuitiva, tipo de público com que se trabalhará, entre outros fatores. Pode-se definir tais etapas em uma fase inicial de definição do problema, para depois passar para a fase de ideação, em que as ideias serão organizadas. A partir daí, começa, de fato, a se projetar um produto, lidando com o design de interface propriamente dito, o que resultará nos primeiros protótipos com os quais terão algum grau de interação. Apenas depois, partiremos para a elaboração efetiva do produto da maneira como que chegará ao público.

É importante ressaltar que as ferramentas que serão explicadas não são utilizadas em todos os projetos de interface, assim como seria demasiado complexo tentar enumerar todas as técnicas existentes e suas variações, considerando que vários métodos possuem mais de um possível uso. Além disso, algumas ferramentas possuem definições variadas, a depender da maneira com que são empregadas e por qual empresa, além de sofrerem modificações com o passar do tempo. Outro fator importante a ser considerado é que várias ferramentas podem ser usadas em mais de uma fase de desenvolvimento. Por exemplo, o grupo focal pode ser usado tanto para definir um problema quanto para levantar dados a respeito de um produto, ou até mesmo verificar a satisfação de um grupo de usuários. Por conta disso, “conhecer esses métodos e saber qual a hora certa de usar cada um deles é uma das habilidades mais procuradas em profissionais de UX no mercado.” (TEIXEIRA, 2014, p. 18). Dessa maneira, daremos início à discussão abordando algumas das principais ferramentas disponíveis para a fase de identificação de problemas em projetos de design de interfaces.



2. Identificação de problemas

Considerando a criação de interfaces como um processo de design, é necessário que, em primeiro lugar, sejam definidos quais problemas um projeto tentará sanar. É relevante pontuar que, em muitos casos, um cliente lida com um problema de maneira restrita, pedindo por uma solução que pode não ser a melhor. Como aponta Lupton (2013), um cliente que quer um folheto poderia aumentar seu público com um site ou uma campanha de marketing organizada. O mesmo pode acontecer com uma aplicação digital que possua uma interface gráfica. Por isso, é importante que a identificação do problema seja feita de modo organizado, com levantamento de dados, necessidades e requisitos, entre outros fatores, além de simplesmente partir de uma premissa simples dada pelo cliente.

Para definir um problema, podem ser utilizadas várias técnicas em que dados são coletados com os usuários. Tais técnicas devem ser escolhidas conforme o acesso disponível às informações e aos usuários (BARBOSA; SILVA, 2010). Isso significa que é necessário verificar qual acesso que teremos ao público, com quantas pessoas se trabalhará, entre outros fatores. Além disso, é importante verificar se o projeto será uma melhoria em um produto já existente ou se um produto novo será criado.

2.1 Entrevista e grupo focal

Entre as técnicas, podemos começar pela entrevista, em que se conversa com os clientes para saber sobre seus desejos e necessidades (LUPTON, 2013). As entrevistas podem ser com perguntas abertas, onde não há nenhuma restrição de tamanho ou tipo de resposta que pode ser dada, ou fechadas, em que se pode responder conforme opções pré-estabelecidas. Cada tipo de pergunta possui suas próprias vantagens e desvantagens.

As perguntas abertas permitem respostas diversificadas, além de aprofundar nos temas abordados, porém, levam mais tempo para serem realizadas e analisadas posteriormente. Enquanto isso, as perguntas fechadas permitem com que análises quantitativas sejam feitas de maneira rápida, porém, limitam as variações de resposta. Isso significa o que pode ou não ser respondido. Aqui, podemos citar também o grupo focal, técnica na qual se lida com uma entrevista ou discussão em grupo, ao invés de apenas uma pessoa por vez. Os grupos focais permitem com que consigamos múltiplos pontos de vista sobre um tópico de maneira rápida (BARBOSA; SILVA, 2010).

2.2 Questionários quantitativos e qualitativos

Outro recurso que pode ser usado é o questionário, que pode ser impresso ou aplicado de maneira on-line. A vantagem dessa ferramenta é que permite a coleta de dados de um número grande de pessoas, em comparação à entrevista. Além disso, o questionário também oferece a facilidade de ser feito por meio de ferramentas digitais, que já realizam toda a organização dos dados coletados (BARBOSA; SILVA, 2010). Também é possível utilizá-lo depois de uma entrevista, a fim de aprofundar em tópicos levantados durante a etapa de entrevista.

Outro ponto importante a ser considerado é que o levantamento de dados pode trazer informações importantes do ponto de vista quantitativo e qualitativo. Por exemplo, o tanto de horas por dia em que o público-alvo acessa Internet e de qual dispositivo são informações quantitativas. Já as impressões que o público-alvo possui da aplicação que estão tentando usar, são informações qualitativas.

2.3 *Brainstorming*: atacando o problema

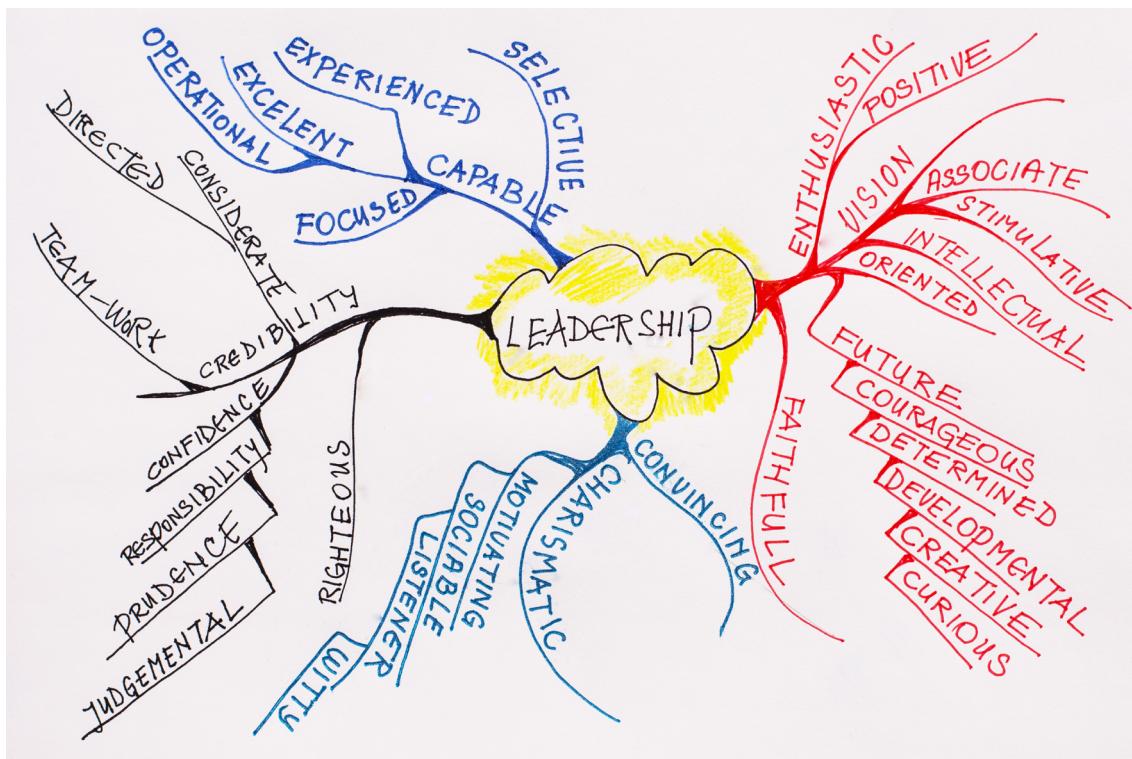
Entre as ferramentas, podemos citar também o *brainstorming*, que pode ser definido como “atacar um problema a partir de todas as direções

possíveis de uma só vez, bombardeando-o com perguntas rápidas para chegar a soluções viáveis" (LUPTON, 2013, p. 16). De modo sucinto, é uma técnica que serve para, em um primeiro momento, gerar ideias e, em um segundo momento, organizá-las. Assim, em um primeiro momento, é dado um tema central ou uma pergunta. A seguir, todos os participantes dão ideias sem filtrá-las, que são anotadas por um moderador. Um ponto importante, nesta etapa, é que todas as ideias são válidas, por mais absurdas que pareçam. Por conta desse ponto, é importante que os participantes não se restrinjam em suas ideias, o que pode impedir que propostas interessantes sejam abordadas. Depois disso, em uma segunda etapa, as ideias são organizadas, filtrando quais podem ser mais úteis ao projeto ou não. Assim sendo, uma vantagem desse tipo de técnica é de que qualquer pessoa pode contribuir, sem a necessidade de conhecimentos técnicos no campo de interfaces.

2.4 Mapa mental

Outra ferramenta interessante é o mapa mental, que consiste em mapear ideias, imagens e propostas que se relacionam a um tópico central. Lida com ramificações e subdivisões, de modo que se aproxima da maneira que o cérebro humano lida com ideias. Em outras palavras, "o mapa mental apresenta os conteúdos da mesma forma como eles são compreendidos pelo cérebro, estimulando a criação das sinapses, o que torna a forma de apresentar e lembrar mais intuitiva e fácil." (GUERRA; TERCE, 2019, p. 49). A Figura 1 mostra um exemplo de mapa mental.

Figura 1 – Exemplo de mapa mental.



Fonte: leuviah/ iStock.com.

É possível ver, na Figura 1, que há uma ideia principal, centralizada na imagem e em tamanho maior, da qual se originam ramificações que levam a outras ideias relacionadas à primeira. Um mapa mental não possui um limite, sendo possível criar ramificações menores nas ramificações maiores.

2.5 Estudos de campo e investigação contextual

Pode ser dado também como exemplo de ferramenta, o estudo de campo, em que um pesquisador visita o usuário para fazer observações, entrevistas e investigação contextual. O objetivo principal é entender o comportamento do usuário em seu ambiente natural de atuação, observando distrações e interrupções, além de outros fatores que não são verificados em um laboratório (BARBOSA; SILVA, 2010).

Por fim, há a investigação contextual, em que se parte “da hipótese de que, quando boa parte do trabalho não pode ser articulada adequadamente por aqueles que o praticam, é necessário que vejamos o trabalho.” (BARBOSA; SILVA, 2010, p. 166). Isso significa que são verificados todos os aspectos que englobam o trabalho que o usuário realiza, por meio da observação dele em seu ambiente de trabalho, realizando as ações que podem trazer dados relevantes para o projeto que desenvolvemos.

3. Ideação

Os dados levantados são úteis para traçarmos possibilidades de criação, levando em conta o perfil do público. Por exemplo, o tipo de dispositivo utilizado. É necessário pensar se o perfil de usuário utiliza celular ou computador, e em qual plataforma seria melhor criar o projeto, conforme o problema a ser trabalhado. Uma aplicação para pedir um transporte particular não seria bem utilizada, caso funcionasse apenas em computadores, por exemplo, assim como não seria uma boa ideia criar uma aplicação para celulares para um público-alvo que não está habituado a utilizá-lo.

A partir deste ponto, chegamos à etapa de ideação, na qual trabalharemos as ideias para possíveis resoluções ao problema em questão. Conforme foi citado no início do texto, aqui, podem entrar técnicas já apresentadas na etapa anterior, como *brainstorming* e mapas mentais. Além destas, podem ser citadas a construção de um perfil de usuário ou o uso de personas, que são ótimas ferramentas para realizar uma análise de usuários para o produto.

A primeira delas, o perfil de usuário, “é uma descrição detalhada das características dos usuários cujos objetivos devem ser apoiados pelo sistema sendo projetado.” (BARBOSA e SILVA, 2010, p. 174). Assim,

tomam-se os dados coletados e cria-se um perfil seguindo uma característica em comum, como faixa etária, nível escolar ou cargo.

Outro recurso é o uso de personas, que é facilmente criado a partir dos perfis de usuário. Entretanto, diferente destes, nas personas, teremos a construção de um personagem fictício que representa um grupo de usuários. Dessa forma, não se trabalhará com um perfil de 30 a 40 anos, que possui afinidade com tecnologia, mas com a persona da Maria, que possui 35 anos, grande afinidade com tecnologia e quer um aplicativo mais fácil para enviar e-mails corporativos, apenas para dar um exemplo. Essa persona possui um objetivo, que deve ser levado em conta no projeto (BARBOSA; SILVA, 2010). O uso de personas é muito útil para criar afinidade da equipe de desenvolvimento do projeto com o público-alvo do produto (TEIXEIRA, 2014). O Quadro 1 mostra um exemplo de persona, com todas as características pensadas.

Quadro 1 – Exemplo de persona

Maria Andrade  <p>Profissão: contadora. Educação: graduada em Ciências Contábeis. Perfil: possui 35 anos, é solteira, mora sozinha em um apartamento na região central de São Paulo.</p>	<p>Atributos:</p> <ul style="list-style-type: none"> Adora ler livros de ficção e sobre tecnologia. Possui grande afinidade com tecnologia, usando celular e notebook, ambos para uso pessoal e profissional. <p>Objetivos:</p> <ul style="list-style-type: none"> Quer um aplicativo mais fácil para enviar e-mails corporativos. Quer economizar tempo ao realizar atividades repetitivas no trabalho. <p>Frustações:</p> <ul style="list-style-type: none"> Precisou trancar uma especialização por conta da alta demanda do trabalho. Precisa fazer hora extra quase todo dia, o que tira seu tempo livre.
--	---

Fonte: adaptado de https://thefader-res.cloudinary.com/private_images/w_1800,c_limit,f_auto,q_auto:best/smile-2072907_1920_hh7tqr/this-bexit-photo-also-comes-from-pixabay.jpg. Acesso em: 10 maio 2022.

Repare, no Quadro 1, que algumas informações não lidam diretamente com o projeto que é feito. No entanto, tais informações ajudam na construção da persona enquanto possível usuário, facilitando a abordagem desta técnica para se pensar as possibilidades de interação.

Além das personas, pode ser citado também o recurso de *moodboard*, que é “uma coleção de imagens e referências que eventualmente se transformarão no estilo visual do produto em questão.” (TEIXEIRA, 2014, p. 26). É muito útil para a equipe de criação mostrar para os colegas de outras áreas, além de para os clientes, qual a linha visual pretendida para o projeto.

4. Planejamento do produto

Com as ideias organizadas e dados analisados, é possível trabalhar com base em como o usuário utilizará o produto, começando, de fato, o planejamento do que será o produto. Aqui, entram técnicas que lidam com conceitos de arquitetura da informação. Trata-se de uma área cuja origem remete à biblioteconomia (TEIXEIRA, 2014), que organiza a grande quantidade de informação em um site ou sistema, de modo que facilite o uso pelo público. Embora seja uma área diferente da UX e da UI, trabalha junto com elas para que a interface tenha o conteúdo e sua distribuição hierárquica e visual estejam organizados e, em consequência, traz uma experiência mais fácil e agradável ao usuário.

Um exemplo de ferramenta para organização é a classificação de cartões, muito útil para organizar a distribuição de informações e facilitar a navegação em um projeto. É um processo simples, em que um conjunto de cartões com amostras de conteúdo é dado aos participantes para que classifiquem em categorias (BARBOSA; SILVA, 2010). Podem ser dados inúmeros exemplos, como a organização dos itens no menu de um portal para web ou a distribuição de elementos em uma interface

gráfica. Para exemplificar melhor, imagine que o projeto chegou a uma proposta de um site que terá trinta páginas diferentes disponíveis ao público. Como devem ser organizadas no menu de navegação? Quais categorias fazem mais sentido para o usuário? Os cartões também são úteis para auxiliar a organizar o conteúdo em si, entre as páginas, respondendo a questionamentos do tipo: o campo de contato deve ter uma página própria ou deve estar embutido em outra página?

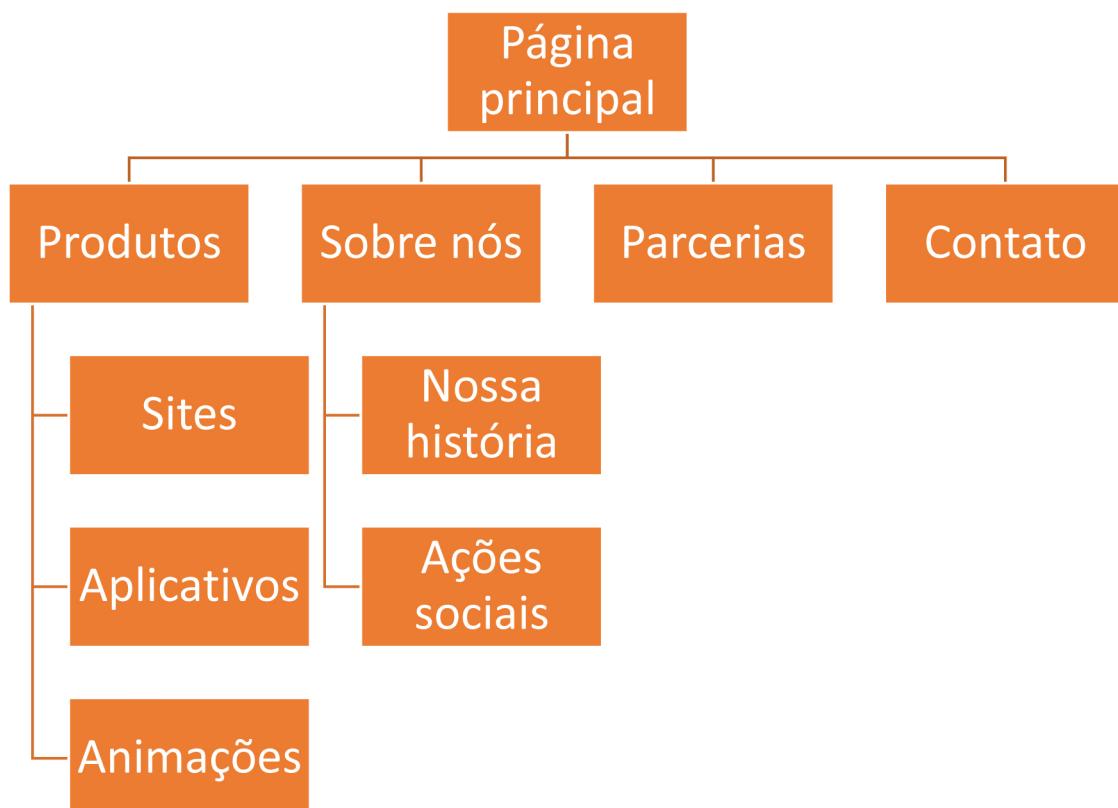
Outro ponto importante é que, dentro do uso dos cartões, pode ser feita uma classificação fechada, em que os participantes recebem também cartões com categorias definidas, ou uma classificação aberta, em que os próprios participantes definem as categorias em que as informações serão organizadas. Assim, essa é uma ferramenta importante para aprendermos como o público pensa e entende o produto que estamos desenvolvendo.

A partir daqui, podemos lidar com vários tipos de fluxograma, como o diagrama de serviços (ou *blueprint* de serviços), o mapa de serviços do usuário (ou *customer journey map*) e o mapa de experiências. Em geral, estes três tipos de diagrama lidam com a maneira com que a interação ocorrerá. No entanto, cada um deles lida de uma maneira específica (KALBACH, 2016). O diagrama de serviços mostra como ocorre a interação do usuário com o serviço em todos os pontos de contato, sendo de grande utilidade para vermos “o caminho que os consumidores percorrem em múltiplos canais (site, serviço de atendimento ao consumidor, loja física etc.) e para identificar oportunidades de melhoria” (TEIXEIRA, 2014, p. 19). Já o mapa de serviços do usuário é derivado do diagrama de serviços, com a diferença de que costumam mostrar a escolha de uma ação como, por exemplo, comprar um produto. Por fim, o mapa de experiência mostra a interação por outro ponto de vista, indicando como que a empresa se encaixa na vida do usuário (KALBACH, 2016).

Outra técnica que pode ser usada é o fluxo de usuário, que é uma técnica em que se cria uma representação visual da navegação do usuário para realizar uma tarefa determinada em uma interface (TEIXEIRA, 2014). O fluxo de usuário está relacionado ao fluxo de navegação, criado na tentativa de indicar os possíveis comportamentos do usuário em uma aplicação (por exemplo, qual caminho deve ser feito para enviar um e-mail).

Pode-se falar também do *sitemap* (ou mapa do site), que é uma técnica em que se monta um diagrama com todas as páginas de um site, ou áreas de um aplicativo de maneira hierárquica (TEIXEIRA, 2014). O fluxo, entre elas, é indicado por linhas, o que auxilia a ver a possível navegação na aplicação. No caso dos sites para web, é comum que o mapa seja inserido na versão publicada do site, no rodapé ou em uma página específica para isso. A Figura 2 mostra um exemplo de *sitemap*.

Figura 2 – Exemplo de *sitemap* genérico



Fonte: elaborada pelo autor.

Repare o quanto a organização do *sitemap* segue uma ordem lógica, em que os elementos estão organizados de maneira fácil e intuitiva. Por exemplo, quero saber sobre a história da empresa, logo vou para a opção *sobre nós*.

O storyboard é uma técnica que vem do meio audiovisual, em que, de maneira sucinta, cria-se uma sequência de imagens, indicando as ações que acontecerão no filme ou, no caso de um projeto de interfaces, no uso da aplicação digital. Assim, nas áreas de UX e UI, o storyboard mostrará a ação de uma persona diante do produto, permitindo lidar com o lado emocional de uma experiência específica no uso de um produto (KALBACH, 2016), de maneira mais tangível e auxiliando a equipe a criar empatia com o público-alvo (TEIXEIRA, 2014). Além disso, é um tipo de ferramenta em que vários membros da equipe podem colaborar.

Junto ao storyboard, existe o uso de cenários, em que é feita uma descrição detalhada da experiência do usuário a partir de seu ponto de vista, diferindo do storyboard por ser uma ferramenta textual (KALBACH, 2016). Semelhante ao storyboard, o cenário permite com que sejam trabalhadas experiências específicas em um projeto, por meio do uso de personas.

A partir de tais recursos, é possível verificarmos quais funções um aplicativo terá, ou quais páginas um site terá, quais as principais funções que um usuário fará na aplicação, entre outros pontos fundamentais para dar início à construção visual da aplicação.

Referências

BARBOSA, S. D. J.; SILVA, B. S. da. **Interação humano-computador**. Rio de Janeiro: Elsevier, 2010.

GUERRA, F.; TERCE, M. **Design digital**: conceitos e aplicações para websites, animações, vídeos e webgames. São Paulo: Senac, 2019.

KALBACH, J. **Mapping experiences**: a complete guide to creating value through journeys, blueprints e diagrams. Boston: O'Reilly, 2016.

LUPTON, E. (org). **Intuição, ação, criação**: Graphic Design Thinking. São Paulo: Gili, 2013.

TEIXEIRA, F. **Introdução e Boas Práticas em Ux Design**. São Paulo: Casa do Código, 2014.

Prototipagem de Interfaces

Autoria: Felipe Orsini Martinelli

Leitura crítica: Gabrielly Del Carlo Richene



Objetivos

- Compreender os principais elementos das interfaces e sua organização visual.
- Diferenciar wireframes, mockups e protótipos no processo de elaboração de interfaces.
- Elaborar protótipos de baixa e alta fidelidade para interfaces gráficas.



1. Construção da interface gráfica

A partir do momento em que se tem todos os requisitos, funcionalidades e fluxos de navegação, pode-se dar início à construção visual da interface de uma aplicação ou site para web. Tal processo também possui várias etapas, o que torna o projeto não apenas organizado, mas também traz melhor fluxo de trabalho para toda a equipe. Imagine o quanto de tempo seria perdido, caso um botão precisasse ser deslocado na interface, o que faria com que outros elementos precisassem de ajuste, depois que o projeto já foi programado e está funcionando corretamente.

Assim, passamos para o processo de construção da interface gráfica, partindo de rascunhos e ideias, o que pode surgir por meio de várias técnicas de processo criativo. Não existe uma regra para criar um rascunho, principalmente se levarmos em conta que se trata de uma representação inicial e bem simples da interface. Podemos dar a isso o nome de *sketches*, ou rabiscos, que ajudarão na validação rápida de conceitos e propostas para um produto (TEIXEIRA, 2014). Tais ideias darão base para a construção do *wireframe*. A partir daí, temos o *mockup*, para depois começarmos a etapa de prototipação, que traz várias possibilidades, que devem ser escolhidas conforme o projeto, o tempo de execução e o orçamento da equipe.



2. Estilos de interação

O usuário pode interagir de diversas maneiras, com uma aplicação para realizar as tarefas que pretende. Desde as maneiras mais complexas, como as linguagens de comando, até as mais simples, como as telas sensíveis ao toque. A linguagem de comando requer treino e o uso de termos que estejam no vocabulário dos usuários (BARBOSA; SILVA, 2010), não sendo uma interação intuitiva. Podemos citar também o uso

de linguagem natural, que lida com termos mais próximos da linguagem humana (e mais distantes da linguagem de máquina). Outros dois tipos de interação são o uso de menus e caixas de diálogos, muito utilizados em softwares e portais web, e o uso de formulários, mais específico para coleta de dados e para realização de cadastros.

Podemos trabalhar também com a chamada *What You See Is What You Get* (WYSIWYG), que pode ser traduzida como *o que você vê é o que obtém*. Esta técnica é muito comum em editores de texto e de imagens, pois o que é feito, na tela, é o que realmente sairá no arquivo final (SOBRAL, 2019). Um exemplo simples disso é o próprio editor de texto: um texto escrito na folha digital será impresso com o mesmo tamanho em uma folha de papel de verdade.

Outro tipo de interação é o uso de ícones, ou representação icônica para representar um objeto, ferramenta ou propriedade. Aqui, é importante que os ícones sigam cinco “princípios básicos: imediação, generalidade, coerência, caracterização e comunicabilidade” (SOBRAL, 2019, p. 118).

Depois, temos a manipulação direta, aproximando o uso de aplicações à maneira com que lidamos com objetos no mundo real. Um exemplo disso é o clicar e arrastar documentos em um computador, ou mover ícones com os dedos em uma tela de celular. A manipulação direta traz menor chance de erro na interação e uma aprendizagem mais rápida, em comparação ao uso de comandos, por exemplo (BARBOSA; SILVA, 2010). Podemos comparar alguns pontos entre tais estilos, conforme o que é mostrado no Quadro 1.

Quadro 1 – Comparativo de estilos de interação

	Tempo de aprendizagem	Velocidade de uso	Predisposição a erros	Extensibilidade	Habilidade de digitação requerida
w	Baixo.	-	Baixa.	Baixa.	-
Manipulação direta.	Baixo.	Média.	Baixa.	Baixa	Nenhuma
Seleção de menus.	Médio.	Média.	Baixa.	Média.	Nenhuma.

Preenchimento de formulários.	Baixo.	Alta.	Baixa.	Média.	Alta.
Linguagem de comandos.	Alto.	Alta.	Alta.	Alta.	Alta.
Linguagem natural.	Baixo.	Média.	Alta.	Alta.	Nenhuma.
Diálogo de perguntas e respostas.	Baixo.	Baixa.	Baixa.	Alta.	Alta.

Fonte: adaptado de Sobral (2019, p. 121).



3. Princípios de design em interfaces

Alguns princípios de design são muito úteis para pensarmos onde inserir elementos em uma interface, de modo que fiquem visualmente agradáveis, além de permitir boa usabilidade. Podemos dizer que “são relativamente constantes e foram concebidos ao longo de vários anos a partir do estudo da cognição e do comportamento humano” (LOWDERMILK, 2013, p. 97).

O primeiro princípio é o da proximidade, também abordado como princípio de agrupamento, que diz que estabelecemos uma relação entre elementos que estão visualmente próximos, e uma separação quando estão distantes (LOWDERMILK, 2013). Dessa maneira, botões em um aplicativo que possuam funções próximas, devem estar próximos na interface, o que torna a interface mais fácil de ser compreendida. O mesmo se aplica à distribuição de itens em menus. Lowdermilk (2013) dá um ótimo exemplo do Microsoft Office. A partir de 2007, foi implementado um menu com abas, em que as ferramentas foram organizadas por meio da semelhança entre si. A Figura 1 mostra um exemplo presente no Microsoft Word, do pacote Office 365, versão mais recente da suíte de aplicativos.

Figura 1 – Menu do software Word



Fonte: print de tela no sistema operacional Windows 10.

Podemos abordar também o princípio de visibilidade, que “corresponde a tudo o que você utilizar para concentrar o foco visual em um elemento ou uma ação na interface de usuário de seu aplicativo” (LOWDERMILK, 2013, p. 99). Aqui, entram características que afetam a visibilidade de um elemento na tela, como tipo de letra, cor, opacidade, relação entre tamanhos e status no sistema. Tal conceito vai de encontro também à ideia de feedback visual, que indica a importância de haver feedback para as interações feitas pelo usuário. Um simples círculo girando, como ocorre em vários aplicativos, ou uma animação de carregamento, como ocorre no Gmail, é o suficiente para indicar que a aplicação está carregando. Apesar de parecer algo simples, esse tipo de recursos é o suficiente para tirar uma possível dúvida por parte do usuário, se o clique feito funcionou ou se o aplicativo não travou.

Outro princípio que podemos aplicar também é o de hierarquia visual, que “estabelece que os aplicativos devem fornecer indicadores visuais para ajudar o usuário a perceber como o aplicativo está organizado” (LOWDERMILK, 2013, p. 102). Este princípio pode ser aplicado com o uso de menus e outras opções de organização e possui forte relação com o primeiro princípio explicado, de proximidade. A hierarquia serve também para indicar a importância de alguns elementos gráficos em cima de outros, de acordo com sua função ser mais utilizada ou menos utilizada. Um exemplo disso é a disposição mais fácil de ser vista e um maior destaque para botões que são mais utilizados, e o contrário para os menos usados.

4. Wireframes e mockups

Os termos *wireframe*, *mockup* e protótipo podem ser facilmente confundidos, inclusive usados como sinônimos. Embora não seja conceitualmente errado indicar um *wireframe* como um tipo de protótipo de baixa fidelidade, abordaremos de maneira separada.

No campo do design de interfaces, o *wireframe* pode ser apontado como uma estrutura gráfica que indica onde ficarão organizados os elementos visuais em uma interface, sem preocupação com cores ou fontes. Isso significa que não estamos preocupados com as cores ou quais imagens serão utilizadas, mas com qual será a posição de um botão específico, se o menu será lateral, horizontal, fixo, flutuante, entre outras opções (MEMÓRIA, 2005). O *wireframe* é uma ferramenta importante para a equipe, pois permite que os diretores de arte tenham noção de como lidarão com os elementos gráficos, assim como, a partir do *wireframe*, a equipe de programação consegue “estimar o esforço de trabalho, começar a modelagem de dados e até a testar soluções de *front-end* que se adequem à solução proposta pelo UX designer” (TEIXEIRA, 2014, p. 42). A Figura 2 mostra um exemplo de *wireframe*.

Figura 2 – Exemplo de wireframe



Fonte: Chaosamran_Studio/ iStock.com.

Repare, na Figura 2, que há uma padronização das telas: os quadrados com linhas cruzadas, como se fosse um x, indicam onde teremos imagens. Já as linhas, indicam campos de texto. As setas vermelhas indicam possíveis fluxos de navegação. Ao clicar em um elemento específico, a aplicação levará para a tela indicada com as linhas. Assim, podemos trabalhar, ainda no papel, possibilidades de navegação, já estabelecendo alguns pontos de como o usuário fará uso do produto.

É importante também que se pense nos métodos de navegação nesta etapa. Em outras palavras, como o usuário navegará, pelo aplicativo ou site, para realizar uma tarefa específica. Enquanto alguns sistemas simples terão uma navegação mais linear, outros como lojas on-line e sites institucionais terão uma navegação mais elaborada (TEIXEIRA, 2014). Assim, entre outros questionamentos que a equipe julgar importantes, é relevante considerar questões como: um menu é necessário? Um campo de busca facilitará o acesso ao conteúdo?

Um ponto que também deve ser considerado desde a concepção do *wireframe* é em relação aos dispositivos de acesso ao aplicativo ou site e tamanhos de tela. Alguns aplicativos podem se comportar de maneiras diferentes, conforme a disposição da tela em modo paisagem ou retrato (TEIXEIRA, 2014). No caso de sites, já se tornou um requisito mínimo que o site seja pensado, primeiro, em como ficará na tela de celulares, para, depois, criar a interface para computadores, ao que se deu o nome de *mobile first* (MULLINS, 2015). Dessa forma, podemos criar um site responsivo, o que quer dizer que o site se adaptará ao tamanho da tela e ao tipo de dispositivo no qual é acessado. Aplicações para criação de sites, trabalham com três tamanhos para testar responsividade: celular, tablet e computadores.

A depender de como a equipe se organiza, depois que um *wireframe* é organizado e aprovado, podemos criar *mockups*, que são imagens da interface, mais fiéis visualmente ao produto final, porém, também sem aspectos de interação (MEMÓRIA, 2005). Os *mockups* são úteis

para mostrar ao cliente como a interface, ou o site, ficará visualmente, já considerando, aqui, a escolha de cores, fontes e disposição dos elementos gráficos na tela. Um ponto importante é que nem sempre são feitos *mockups*, algo que está mais relacionado com a área de sites do que aplicativos. No entanto, pode ser de grande utilidade na comunicação com clientes. A partir disso, podem ser criados os protótipos.

5. Protótipos

Tendo já definido como os elementos ficarão, damos início à prototipação. O protótipo é a criação de um modelo da interface interativa, podendo possuir aspectos gráficos bem elaborados ou não. O protótipo é uma melhor opção tanto para mostrar ao cliente quanto para testar com usuários. Diferente do *wireframe*, não possuirá várias telas estáticas em um arquivo PDF, por exemplo, mas um arquivo mais simples, já com um modelo de navegação (TEIXEIRA, 2014). É comum a prototipação em projetos de tecnologia, por uma questão de economia de recursos e tempo da equipe. Isso porque verifica possíveis problemas na navegação, que, caso fossem vistos apenas quando a interface estivesse pronta e codificada, demorariam muito mais tempo para serem ajustados. Imagine criar um aplicativo e descobrir que a navegação está confusa para a maioria dos usuários, sendo necessário trabalhar com alterações espalhadas por entre cinco mil linhas de código.

Os protótipos podem ser abordados de acordo com sua proximidade com o projeto final. De um modo bem simplificado, existem os protótipos de baixa e alta fidelidade. O conceito de fidelidade remete ao protótipo ser visualmente parecido ou não com o produto final. Dessa forma, um protótipo de baixa fidelidade é visualmente mais simples, ao mesmo tempo em que é bem diferente do que virá a ser o projeto

finalizado. Apesar da simplicidade, os de baixa fidelidade permitem com que várias avaliações sejam feitas, tanto com a equipe quanto com usuários, de maneira rápida e barata, o que traz maior qualidade para o produto final (SOBRAL, 2019). Por outro lado, os protótipos de alta fidelidade são bem próximos do produto final, mas requerem mais tempo para realizar ajustes. O Quadro 2 traz um comparativo interessante entre os dois tipos de prototipagem.

Quadro 2 – Comparativo entre protótipos

Tipo protótipo.	Fator mais positivo.	Aplicação no ciclo de desenvolvimento.	Custo de alterar a aparência (look and feel).	Custo de alterar a sequência – comportamento.
Baixa fidelidade.	Flexibilidade, facilidade de alterar sequência, comportamento geral.	Início.	Quase nenhum.	Baixo.
Alta fidelidade.	Fidelidade da aparência (look and feel).	Final.	Baixo.	Alto.

Fonte: adaptado de Sobral (2019, p. 76).

Um ponto importante, a ser considerado, é que alguns autores ainda classificam os protótipos como baixa, média e alta fidelidade. Neste caso, podemos considerar os de baixa fidelidade como *wireframes*, feitos digitalmente ou no papel, de maneira rápida, para compreender as necessidades dos usuários e os requisitos que uma aplicação deve ter (COUTO, 2019). Os de média fidelidade são mais elaborados do que os de baixa, já com indicações dos fluxos de navegação, enquanto os de alta fidelidade são também visualmente similares ao aplicativo final, tendo vários fluxos de navegação completos para testar.



6. Aplicações para prototipagem

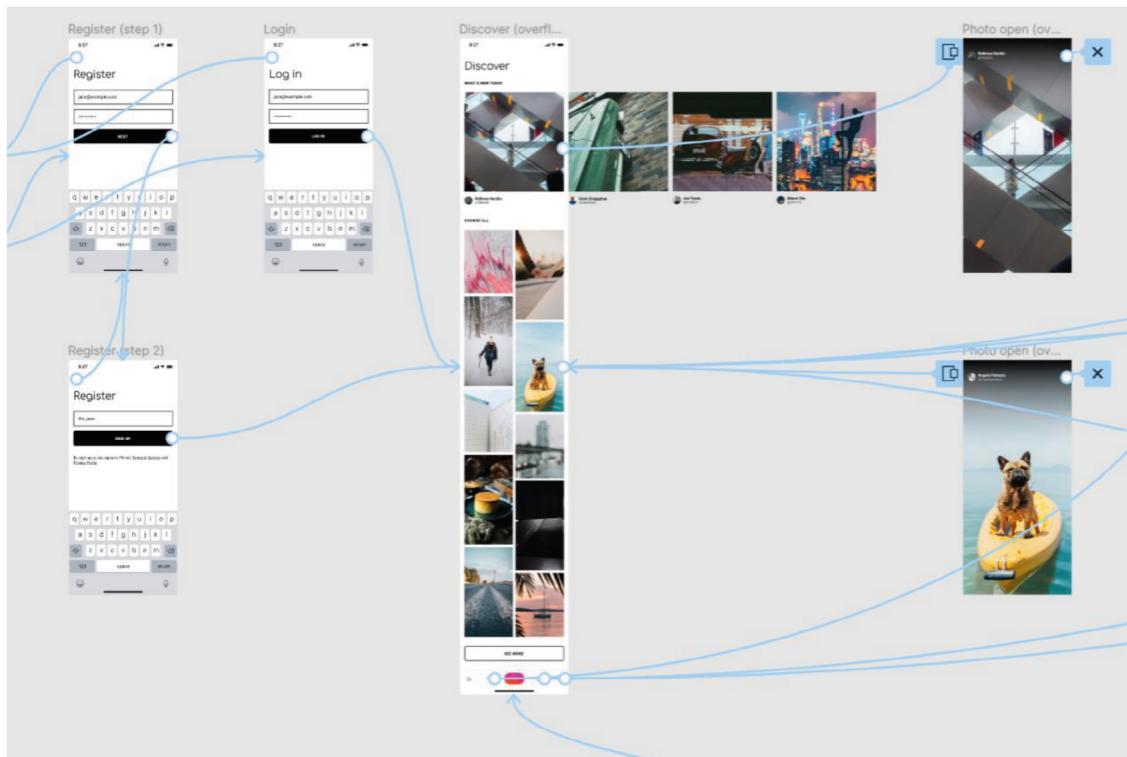
A criação de protótipos requer softwares específicos para sua elaboração. Softwares de edição gráfica, como o Adobe Photoshop, Adobe Illustrator ou o Corel Draw, não possuem recursos de interação, que são de extrema importância para pensar na navegação que será feita na interface. No entanto, tais softwares podem ser de grande utilidade na construção de *wireframes* e *mockups*. Já para que possamos trabalhar com fluxos de navegação, podem ser utilizados softwares específicos para a criação de *wireframes* e protótipos, que oferecem a construção gráfica e interativa de interfaces de maneira bem simples e rápida, sem a necessidade de elaborar o código que funcionará por trás da interface (COUTO, 2019, p. 49).

Há inúmeros softwares e aplicações que funcionam on-line, sem a necessidade de instalação, para tal finalidade. Para *wireframes* simples, podem ser usadas ferramentas como o Draw.io, que funciona on-line e permite a criação fácil de *wireframes* simples. Outras ferramentas, como Adobe XD e Figma, já trazem recursos de interação para prototipagem, além de várias destas ferramentas possuírem recursos, como a criação de um projeto em equipe e a visualização em um aparelho celular do projeto.

Essas ferramentas são disponibilizadas como aplicações on-line, *plugins* ou versões desktop. Além disso, a maioria delas possui uma versão comercial com vários planos disponíveis, além de uma versão de testes. Couto (2019) lista as ferramentas Balsamiq, Draw.io, Mockflow, Figma, Adobe XD e Sketch como de grande relevância no mercado. Outros exemplos famosos disponíveis são: Axure, Marvel e InVision. Um ponto importante a ressaltar é que não existe uma ferramenta melhor do que a outra, principalmente, se levarmos em consideração que a maioria delas possuem recursos parecidos e trazem resultados bem próximos.

A Figura 3 mostra uma captura de tela da versão on-line do Figma, com um arquivo, para demonstração de ferramentas, aberto.

Figura 3 – Exemplo de projeto no aplicativo Figma



Fonte: print de tela no sistema operacional Windows 10.

Como é possível constatar, as telas da Figura 3 possuem uma proximidade muito alta com uma possível aplicação para celulares, diferente de um protótipo de baixa fidelidade. Além disso, o uso de tais softwares é de grande facilidade, sem a necessidade de programação ou algum outro recurso complexo. Repare também nas setas azuis, que indicam o fluxo de navegação ao interagir com alguns dos botões. Tal fluxo é facilmente testado ao clicar na opção de apresentação do projeto, o que pode ser visto no computador ou em um celular, por meio de um navegador ou do aplicativo do próprio Figma instalado.



Referências

- BARBOSA, S. D. J.; SILVA, B. S. da. **Interação humano-computador**. Rio de Janeiro: Elsevier, 2010.
- COUTO, J. M. C. Projeto de aplicativo via *mockups*. In: SIMAS, Victor Luiz *et al.* **Desenvolvimento para Dispositivos Móveis**, v. 2. Porto Alegre: SAGAH, 2019.
- LOWDERMILK, T. **Design centrado no usuário**: um guia para o desenvolvimento de aplicativos amigáveis. São Paulo: Novatec, 2013.
- MEMÓRIA, F. **Design para a Internet**. Projetando a experiência perfeita. Rio de Janeiro: Elsevier, 2005.
- MULLINS, C. Responsive, Mobile App, Mobile First: Untangling the UX Design Web in Practical Experience. In: **Proceedings of the 33rd Annual International Conference on the Design of Communication**, p. 1-6, [s. l.], 2015.
- SOBRAL, W. **Design de interfaces**. São Paulo: Érica, 2019.
- TEIXEIRA, F. **Introdução e Boas Práticas em Ux Design**. São Paulo: Casa do Código, 2014.

Avaliações de Usabilidade

Autoria: Felipe Orsini Martinelli

Leitura crítica: Gabrielly Del Carlo Richene



Objetivos

- Analisar técnicas para a realização de avaliações de interfaces.
- Compreender diferentes heurísticas para projetos de interface gráfica.
- Aprofundar nas heurísticas de usabilidade de Nielsen.

1. Introdução

Depois que criamos um aplicativo, software ou site, precisamos saber como os usuários lidarão com o produto. Para isso, trabalharemos com diversos tipos de testes e análises, verificando se a interface está fácil de usar, se a interação ocorre de maneira intuitiva, entre outros fatores.

A partir dos conceitos de acessibilidade, comunicabilidade e usabilidade, podemos realizar avaliações de usabilidade, seguindo heurísticas gerais da área, ou próprias de uma plataforma, conforme o que será discutido a seguir.

2. Acessibilidade, comunicabilidade e usabilidade

Para lidar com a interação do usuário em um sistema, existem três conceitos-chave a serem trabalhados. O primeiro deles é a acessibilidade, que consiste em remover empecilhos que impeçam usuários de utilizarem um sistema, tendo algum tipo de deficiência ou não. É importante ressaltar que a “acessibilidade atribui igual importância a pessoas com e sem limitações na capacidade de movimento, de percepção, de cognição e de aprendizado. (BARBOSA; SILVA, 2010, p. 33). Existem vários exemplos do que pode ser feito para remover esses empecilhos, como o uso de leitores de tela em sites, além de adaptar os sites para que funcionem corretamente com esses leitores. Um outro ponto interessante, a respeito da acessibilidade, é de que pode trabalhar também com limitações temporárias, como é o caso de uma pessoa dirigindo um carro, que precisa usar o GPS do celular. Esse usuário não poderá manusear o aparelho, nem ler instruções na tela. Dessa forma, o aparelho lida com indicações sonoras de como prosseguir no trajeto (BARBOSA; SILVA, 2010).

O segundo conceito é o de comunicabilidade, que remete ao designer comunicar ao usuário qual é a estratégia de uso para resolver uma tarefa na interface desenvolvida, de modo que o usuário comprehenda tal lógica e consiga usar uma interface sem maiores problemas. Uma observação relevante é que compreender a lógica envolvida no design não quer dizer adquirir um conhecimento aprofundado do projeto. Não precisamos saber com profundidade sobre o funcionamento mecânico de um carro, por exemplo, para dirigirmos, assim como não precisamos saber como foi todo o desenvolvimento de um software para fazermos bom uso dele (BARBOSA; SILVA, 2010).

Por fim, o terceiro conceito é o de usabilidade, que pode ser definido como um conjunto de técnicas que visam a criação de produtos fáceis de usar e que tragam satisfação para o usuário ao utilizá-los (NIELSEN, 1993). Aqui, devem ser levados em conta fatores cognitivos, que interferirão na maneira que o usuário usa um sistema, além de fatores emocionais e sentimentais (BARBOSA; SILVA, 2010).

A usabilidade indica que um produto deve oferecer: fácil aprendizagem para ser utilizado; eficiência no uso, o que traz alta produtividade; memorabilidade, ou seja, facilidade em memorizar as ações que podem ser feitas; satisfação de uso ao usuário; e prevenção de erros, o que quer dizer que deve auxiliar o usuário a cometer a menor quantidade de erros possível (NIELSEN, 1993).

A partir desses conceitos, é possível adentrar aos processos avaliativos de usabilidade em um projeto de interface.

3. Avaliações de usabilidade

Lidamos com diversos tipos de avaliações no desenvolvimento de um produto. Podemos lidar durante o desenvolvimento, em especial para

tirarmos dúvidas que surgem no processo de design, de modo que as avaliações se mesclam ao processo de desenvolvimento (ROCHA; BARANAUSKAS, 2003). Aqui, entram diversas técnicas de coleta de dados e criações de protótipos, que já permitem que encontremos problemas de usabilidade antes do projeto ser construído. Além dessas técnicas, durante o desenvolvimento, podemos trabalhar com avaliações depois que o produto já está pronto, que pontos importantes relativos à interação e satisfação do usuário, que não conseguiríamos verificar em etapas anteriores.

É relevante ressaltarmos que as avaliações, depois que um produto foi finalizado, devem ser feitas com frequência, de modo a melhorar constantemente um projeto. Um software dificilmente deixa de passar por avaliações de usabilidade depois que foi lançado, havendo várias mudanças de interface e implementação de novos recursos a cada nova versão. Como aponta Lowdemilk (2013), os softwares não acertam todos os pontos de usabilidade na versão 1.0.

Entre essas avaliações, temos as relativas à usabilidade, que consistem em diversas técnicas para analisar o uso de um produto, tendo os testes com usuários como parte central. Aqui, entram questionários, entrevistas e a observação dos usuários. Um ponto a ser destacado é que as avaliações de usabilidade possuem grande importância, pois trazem dados quantitativos para análise, ao invés de informações subjetivas dadas pelos usuários (LOWDERMILK, 2013). Esta quantificação auxilia em testes, principalmente, quando queremos medir a performance dos usuários na execução de uma tarefa, como, por exemplo, quantos segundos demorarão, em média, para a realização de login em uma aplicação.

A performance é um ponto de grande importância nos testes de usabilidade, considerando que a eficiência de uso é um dos pontos fundamentais para um produto ter boa usabilidade (ROCHA; BARANAUSKAS, 2003). Dessa maneira, podem ser medidas ações como:

quantidade de erros realizados pelo usuário; frequência de acesso à documentação da aplicação; quais funções não foram usadas em nenhum momento; a razão entre a quantidade de ações bem-sucedidas e a quantidade de erros; entre várias outras opções.

Em relação à observação de usuários, pode ser feita de duas maneiras. A primeira é observá-los no ambiente de uso normal, ou seja, de uso cotidiano, como em um escritório. A este tipo de ambiente, se dá o nome de ambiente natural. A segunda maneira seria em um laboratório específico para esse tipo de avaliação, que contará com equipamentos específicos para os testes (PREECE; ROGERS; SHARP, 2013). Quando feita em um laboratório, costuma-se gravar vídeos e áudio da pessoa utilizando o equipamento, além de captura de tela, para verificar como foi feita a navegação. Costuma-se traçar uma série de tarefas a serem realizadas, o que torna possível analisar se o sistema está confuso para tais tarefas, se os usuários acharam as funções que precisavam, entre outros fatores.

Um ponto a ser considerado é que o uso de laboratório não é um requisito para os testes, mas apenas um facilitador, pois estes podem ser feitos em vários lugares diferentes (ROCHA; BARANAUSKAS, 2003). Nos dias atuais, há também várias ferramentas disponíveis para a realização de testes de maneira remota, como sites que captam a movimentação do mouse pela tela, apenas para dar um exemplo simples.

É importante que se atente a dois fatores nos testes de usabilidade: a confiabilidade, que pode ser definida como o grau de certeza de que teremos sempre o mesmo resultado com o teste repetido várias vezes; e a validade, que aponta os resultados como um reflexo do que queremos realmente avaliar (ROCHA; BARANAUSKAS, 2003). A quantidade de usuários a realizarem esse tipo de teste, contribuirá com esses dois fatores. No entanto, não são necessários muitos usuários em um teste, se forem realizados de maneira iterativa. Conforme aponta Nielsen

(2000), cinco usuários são o suficiente. Isso porque a cada usuário a mais, você aprenderá menos de seu comportamento, pois haverá muitas repetições de comportamentos já vistos nos usuários anteriores. Assim, depois do quinto usuário, a quantidade de coisas novas a serem aprendidas é mínima, tornando-se perda de tempo dentro dos testes.

Um outro ponto a destacarmos é de que um teste pode ser separado em 4 etapas, seguindo o que apontam Rocha e Baranauskas (2003). A primeira delas é a preparação, principalmente, quando realizado em um laboratório específico de usabilidade: verificar se os equipamentos estão ajustados, se o ambiente está limpo, entre outros fatores. A segunda etapa é a introdução, na qual a situação é apresentada aos usuários. Aqui, é importante que se deixe os usuários à vontade, indicando que o teste é relativo ao sistema, e não a ele. Além disso, explicar que podem dar feedback e fazer perguntas a qualquer momento. Por fim, deve-se explicar também como será realizado o teste, quais técnicas serão aplicadas e que, caso queiram, podem parar com a participação a qualquer momento. A terceira etapa remete à aplicação do teste em si. Depois, a quarta etapa é a da sessão final, em que os usuários podem responder a um questionário e fazer perguntas gerais sobre o teste que foi feito.

Aqui, devemos considerar que alguns usuários podem evitar dar feedbacks por receio de ofender a equipe de avaliação, o que é importante de ser explicado que não acontecerá. Apesar disso, aprender a ouvir feedbacks é uma aptidão de extrema relevância a quem quer trabalhar em qualquer etapa da elaboração de interfaces, considerando que muitos feedbacks serão desagradáveis de se escutar, porém necessários (LOWDERMILK, 2013).



4. Heurísticas

As avaliações heurísticas não focam em uma tarefa em particular, mas sim avaliam um produto a partir de um conjunto de regras ou diretrizes (LOWDERMILK, 2013). Algumas empresas de tecnologia oferecem essas diretrizes, em especial quando serão desenvolvidos projetos para suas plataformas. Por exemplo, a Apple possui diretrizes específicas para a criação de aplicativos para o iOS, assim como a Google possui suas próprias para criarmos aplicativos para o Android. Ao criarmos um aplicativo para ambas as plataformas, precisamos que cada versão siga um conjunto diferente de regras e diretrizes. Seguir tais regras ou não, determina a aprovação, ou posterior banimento, do aplicativo da plataforma.

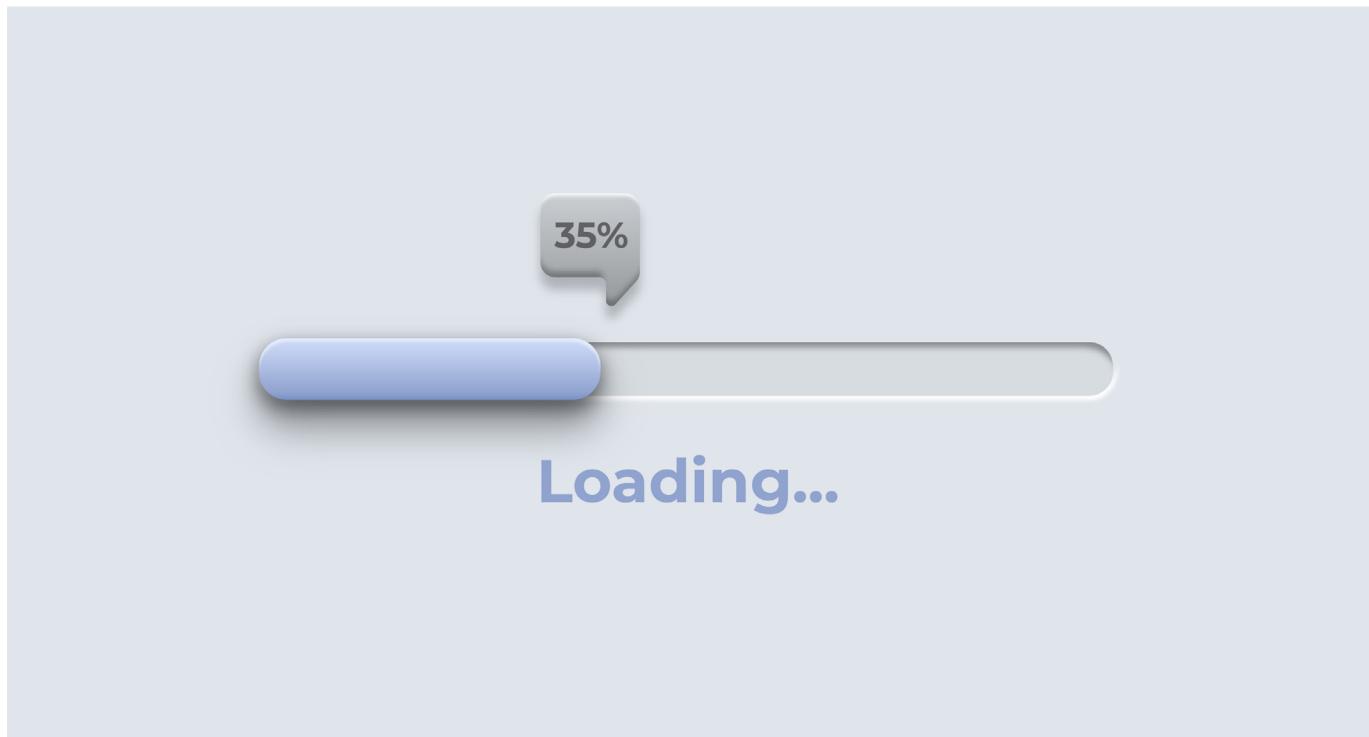
Podemos também realizar testes usando heurísticas de usabilidade, ou seja, um conjunto de regras que devem ser verificadas em um produto para que se ateste uma boa usabilidade. Nielsen (2020) determina dez heurísticas, a partir dos pontos relativos à usabilidade que foram tratados ao final do item 2.

4.1 Visibilidade do status do sistema

Este princípio é relacionado ao feedback do sistema, e diz que o sistema deve informar constantemente ao usuário como está o andamento de uma tarefa. A falta de feedback pode trazer frustração durante o uso, pois fica difícil saber se um comando funcionou e algo está carregando ou se o aplicativo travou. Aqui, entram diversos exemplos, como: barras de carregamento; mensagens pop-up, que confirmam se algo aconteceu ou não; mensagens de confirmação para uma ação; entre outros. Outro ponto a ser considerado, neste princípio, é a importância de indicar, por exemplo, de maneira clara, qual arquivo está selecionado em uma pasta, ou qual música está tocando dentre as diversas existentes em uma

lista. A Figura 1 mostra um exemplo de barra de carregamento, item relacionado à ideia proposta nessa heurística.

Figura 1 – Exemplo de barra de carregamento



Fonte: merovingian/ iStock.com.

Pode-se ver, na Figura 1, uma barra indicando o carregamento de alguma ação, que pode ser um arquivo sendo baixado, copiado ou excluído. Esse tipo de feedback visual permite verificarmos que alguma ação acontece gradualmente, sendo o valor em porcentagem alterado com o passar do tempo. Neste caso, uma animação simples, como uma pequena movimentação das reticências, auxilia no feedback, ao indicar que o sistema não travou.

4.2 Correspondência entre o sistema e o mundo real

O design de um produto deve falar a linguagem do usuário e deve também trabalhar conforme convenções do mundo real. Dessa forma, o uso de alguns conceitos ou termos específicos podem parecer facilmente compreensíveis para nós, mas confusos para os usuários.

Além disso, é importante que sejam feitas relações com o mundo real, de modo que um produto se torne intuitivo.

Neste item, também podem ser discutidas as questões relacionadas ao uso de metáforas, como o uso de ícones relativos a funções específicas dentro de um sistema (NORMAN, 2006). Exemplos disso são os diversos aplicativos para envio de e-mails, que possuem um desenho de envelope, ou para o envio de mensagens, que possuem um símbolo que traz uma ideia de mensagem, como é o caso dos aplicativos WhatsApp e Telegram.

4.3 Controle e liberdade ao usuário

Devemos oferecer aos usuários opções de emergência, na execução de tarefas por engano, como o comando de desfazer com o Ctrl + Z, ao apagar um pedaço de texto sem querer (NIELSEN, 2020). Esse tipo de recurso traz maior controle do sistema para o usuário, além de trazer uma sensação de liberdade e confiança ao executar tarefas. Além disso, evitam a frustração de perder um trabalho por conta de algum erro simples.

Essa heurística também está relacionada com as possibilidades de navegação em um site ou aplicativo, em que temos o controle de voltar e avançar por entre as janelas da aplicação. Por exemplo, caso tenhamos feito uma escolha e queiramos alterá-la, basta clicar no botão de voltar e o site continuará como estava antes de tomarmos a primeira ação.

4.4 Consistência e padrões

Uma aplicação deve ter consistência entre suas telas, assim como deve seguir padrões já estabelecidos na indústria e na plataforma para a qual foi desenvolvida. Isso quer dizer que, caso se queira inserir um comando de copiar e colar, é preciso seguir os comandos de atalho já

estabelecidos em outros softwares no mesmo sistema operacional. Caso contrário, poderá trazer frustração e confusão ao usuário, que já estão acostumados com os mesmos atalhos em praticamente todos os outros softwares. Da mesma forma, produtos de uma mesma empresa costumam seguir um mesmo padrão de interface, o que facilita o uso, como é o caso dos principais softwares do pacote Office, da Microsoft (Word, Excel e Power Point) ou do pacote Adobe Creative Cloud (Photoshop, Illustrator, Premiere, entre outros).

4.5 Prevenção de erros

Erros devem ser evitados o máximo possível. Nielsen (2020) aponta dois tipos possíveis de erros: deslizes e equívocos. Os deslizes acontecem por falta de atenção, e podem ser evitados ao limitar a execução de algumas tarefas dentro do sistema, estabelecendo padrões bem definidos para seu uso (o que podemos relacionar à heurística anterior). Já os equívocos, ocorrem por incompatibilidade entre o modelo mental do usuário e do design de um produto. Isso significa que o usuário acredita estar fazendo uma ação, porém, entende errado as estratégias de design do sistema e realiza outra ação. Os equívocos podem ser evitados ou corrigidos, oferecendo a opção de desfazer uma tarefa e exibindo avisos de confirmação.

4.6 Reconhecimento ao invés de memorização real

Elementos, ações e opções devem ser facilmente visíveis na interface, ou facilmente recuperáveis quando necessárias. Dessa maneira, uma informação deve ser reconhecida ao invés de memorizada. Uma dica importante para evitar a memorização é de oferecer pequenos blocos de ajuda conforme as tarefas são realizadas, no lugar de oferecer um tutorial longo a ser lido.

4.7 Flexibilidade e eficiência de uso

O sistema deve ser de fácil uso, tanto por usuários leigos quanto por experientes, que podem utilizar atalhos que são desconhecidos para os leigos. Isso quer dizer que o sistema precisa ser flexível, oferecendo várias maneiras de realizar uma mesma tarefa, sendo uma escolha do usuário a maneira que achar melhor para si. Nesse ponto, é útil que sejam oferecidas possibilidades de personalizar atalhos e funções do software, como, por exemplo, a ordem dos botões ou qual formato de arquivo deve ser salvo como padrão.

4.8 Estética e design minimalista

“Interfaces não devem conter informação que é irrelevante ou raramente necessária” (NIELSEN, 2020, [n. p.]). Essa é uma heurística importante de ser pensada, pois cada elemento gráfico, que é adicionado na tela, competirá pela atenção do usuário, o que pode atrapalhar na navegação. Dessa forma, a interface deve ser simples, limpa e focar nos objetivos primários do usuário.

4.9 Ajudar usuários a reconhecer, diagnosticar e se recuperar de erros

Conforme a quinta heurística (NIELSEN, 2020), de prevenção de erros, estes devem ser evitados. Entretanto, mesmo tentando evitar o máximo de erros possíveis, ainda algum deles acontecerá. Caso ocorram, é importante informar ao usuário o que, de fato, aconteceu, e não dar apenas uma mensagem de erro genérica, ou um código de erro que não possa ser utilizada para a compreensão do usuário. Além disso, mensagens de erro devem trazer uma possível solução para o usuário, além de deixar explicado de maneira clara qual erro que aconteceu, evitando termos técnicos. Isso significa que se deve evitar uma

mensagem que diga, por exemplo, apenas “erro 0xc00000f”, pois, para o usuário, esse código não quer dizer nada.

4.10 Ajuda e documentação

Esta última heurística diz que um sistema deve ser o mais intuitivo possível, de modo que o usuário não precise de ajuda para utilizá-lo. No entanto, caso necessite, a ajuda deve ser oferecida de maneira simples e de fácil compreensão. Um exemplo disso são tutoriais com animações, indicando qual botão deve ser pressionado para realizar uma ação. Além disso, é importante que seja possível realizar buscas por termos específicos dentro da documentação oferecida ao usuário.

5. Demais técnicas

Outra técnica relevante é a de pensar em voz alta, que consiste em pedir ao usuário verbalizar tudo que pensar enquanto realiza as tarefas. Trata-se de uma técnica interessante para verificar alguns aspectos da navegação, mas não é uma boa técnica para verificar falhas nesta navegação, ou para medir a performance para executar uma tarefa (ROCHA; BARANAUSKAS, 2003). Além disso, talvez seja necessário realizar perguntas para o usuário durante o teste, incentivando-o a falar. Podemos ressaltar que “a principal força dessa técnica é mostrar *o que* os usuários estão fazendo e *porque* estão fazendo *enquanto* estão fazendo, evitando as racionalizações posteriores” (ROCHA; BARANAUSKAS, 2003, p. 204).

Podemos falar também dos testes A/B, muito úteis para a escolha entre duas opções de design, verificando, com os usuários, qual opção preferem ou qual opção funciona melhor para a execução de determinada tarefa. Esse tipo de teste é feito com dois grupos de usuários, sendo que cada grupo receberá um projeto para testar com

algum ponto diferente entre as duas versões. Pode ser, por exemplo, a posição de um botão para compra, a distribuição de todos os elementos na tela, ou apenas a mensagem em uma chamada em um site.

É importante ressaltar que se trata de um teste às escuras, pois os usuários não sabem que há duas versões sendo testadas (LOWDERMILK, 2013). A partir disso, podemos levantar dados, como: qual opção fez com que os usuários realizassem uma tarefa mais rápido, ou qual opção de botão permitiu com que uma compra fosse feita de maneira mais agradável, entre inúmeros outros exemplos que podem ser dados. Apesar das possibilidades que esse tipo de teste oferece, é importante tomar cuidado para não focar apenas nos detalhes, deixando de lado a experiência do usuário com o produto em sua totalidade.



Referências

- BARBOSA, S. D. J.; SILVA, B. S. da. **Interação humano-computador**. Rio de Janeiro: Elsevier, 2010.
- LOWDERMILK, T. **Design centrado no usuário**: um guia para o desenvolvimento de aplicativos amigáveis. São Paulo: Novatec, 2013.
- NIELSEN, J. **Usability Engineering**. Burlington: Morgan Kaufmann, 1993.
- NIELSEN, J. Why You Only Need to Test with 5 Users. In: **Nielsen Norman Group, 2000**. Disponível em: <https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/> Acesso em: 10 maio 2022.
- NIELSEN, J. 10 Usability Heuristics for User Interface Design. In: **Nielsen Norman Group, 2020**. Disponível em: <https://www.nngroup.com/articles/ten-usability-heuristics/> Acesso em: 10 maio 2022.
- NORMAN, D. **O Design do Dia-a-Dia**. Rio de Janeiro: Rocco, 2006.
- PREECE, J.; ROGERS, Y.; SHARP, H. **Design de interação**: além da interação humano-computador. Porto Alegre: Bookman, 2013.
- ROCHA, H. V. da; BARANAUSKAS, M. C. C. **Design e Avaliação de Interfaces Humano-Computador**. São Paulo: NIED/UNICAMP, 2003.



BONS ESTUDOS!