



Material do Aluno

Técnico em Desenvolvimento *Web e Cibersegurança*

UX e UI

Fundamentos de UX e UI - PARTE 2



**Instituto
Telles**

Parcerias:

SEDUC
Secretaria de Estado
da Educação

SECTI
Secretaria de
Estado de Ciência,
Tecnologia e Inovação

GOVERNO DE
GOIÁS
O ESTADO QUE DÁ CERTO

Sumário

Capítulo 01: USABILIDADE E INTERAÇÃO:

COMPREENSÃO DAS NECESSIDADES DO CONSUMIDOR	4
1.1 Compreensão da usabilidade e necessidades do consumidor	4
1.1.1 A Definição de Usabilidade	5
1.2 Padrões de interface: em busca de facilitar a compreensão do usuário	10
1.2.1 Alguns exemplos de boas práticas no design	12
1.3 Design centrado no usuário e em suas necessidades	16
1.4 Introdução ao wireframing e à prototipagem	19
1.4.1 Ferramentas de criação para prototipagem e wireframing	20
1.4.2 Tipos de protótipos e suas definições	21
1.4.3 Vamos testar protótipos de baixa fidelidade?	23

Capítulo 02: DO DESIGN À IMPLEMENTAÇÃO: ESTRATÉGIAS TÉCNICAS PARA COMUNICAR, ADAPTAR E REFINAR INTERFACES RESPONSIVAS

	27
2.1 Comunicando design para desenvolvedores e partes interessadas	28
2.2 Design responsivo: adaptação de layouts para dispositivos móveis	30
2.2.1 Ferramentas do desenvolvimento	32
2.2.2 Entendendo a estrutura de uma página e sua responsividade	34
2.3 A utilização de media queries	38
2.4 Conhecendo e utilizando ferramentas de depuração para testes em diferentes tamanhos de tela	43
2.4.1 Operando e entendendo o DevTools	45
2.5 Otimizando experiências: testes de usabilidade, métricas UX e feedback do usuário na iteração de design	49
Referências	53

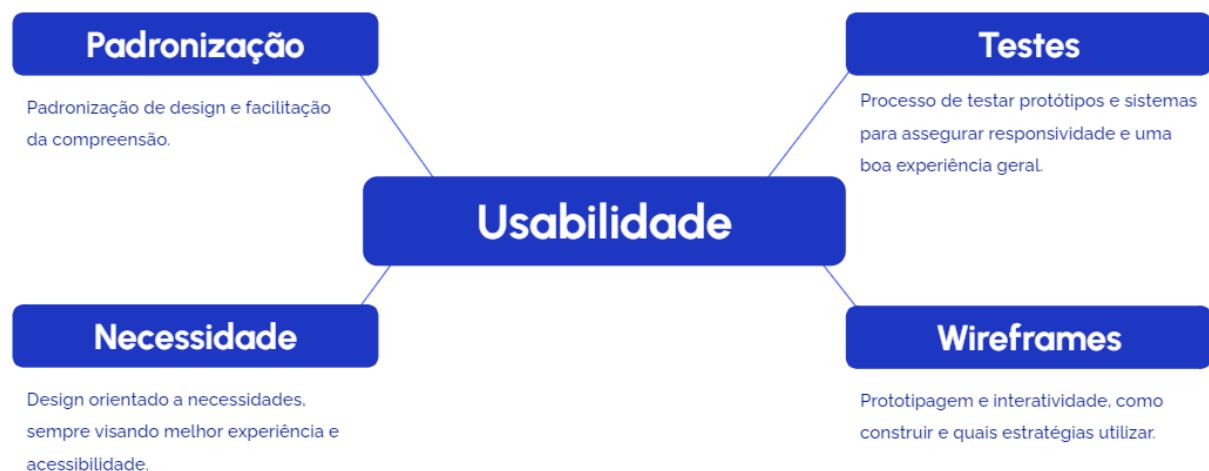
CAPÍTULO 01

USABILIDADE E INTERAÇÃO: COMPREENSÃO DAS NECESSIDADES DO CONSUMIDOR

O que esperar deste capítulo:

- Definir a usabilidade e a sua importância no *design* de produtos digitais;
- Manter padrões ao longo da interface, buscando facilitar a compreensão;
- Explorar o processo de *design* centrado no usuário, compreendendo as suas necessidades, características e preferências;
- Conceituar a prototipagem e o *wireframing*, ferramentas para criar protótipos interativos e *wireframes*;
- Testar a usabilidade com protótipos de baixa fidelidade.

1.1 Compreensão da usabilidade e necessidades do consumidor



Antes de nos aprofundarmos no assunto, vamos, juntos, explorar a questão a seguir.

Vamos refletir

Para você, o que é um objeto útil?



De forma geral, um objeto é considerado **útil** quando **serves** a um propósito específico ou facilita determinada tarefa ou atividade. Por exemplo, uma chave de fenda é **útil** para apertar ou soltar parafusos, enquanto um garfo pode ser utilizado para **pegar** e **manusear alimentos sólidos**. Nem todos os objetos são igualmente úteis para todo mundo, mas, geralmente, eles seguem um padrão específico que tende a facilitar a vida cotidiana das pessoas.

Quando o tema é utilidade, é comum refletirmos sobre **o tipo de finalidade** de cada objeto, fenômeno que é intitulado **usabilidade**. A usabilidade nos ajuda a refletir sobre os diferentes contextos nos quais um mesmo material pode ser aplicado.

1.1.1 A definição de usabilidade

Usabilidade refere-se à **facilidade de uso** e **eficácia de um produto** ou **sistema** na realização de uma tarefa específica, levando em consideração o usuário. Desse modo, um produto ou sistema é considerado "usável" quando é intuitivo, eficiente e satisfatório para quem o utiliza.

Para entender melhor como isso ocorre na prática, observe a imagem.

Analise o exemplo a seguir

Uma máquina de café automática é considerada usável se os usuários conseguirem preparar café de maneira simples, ou seja, sem a necessidade de muitas instruções ou esforço. Leva-se em consideração, também, se o café preparado atende às expectativas de sabor e qualidade esperadas.

Após analisar a imagem da máquina de café ao lado, você a classifica como usável?



Como é possível ver, a estrutura de determinado produto afeta, de forma direta, o modo como o consumidor interage com ele.

A usabilidade é vital nos tempos atuais, especialmente por conta do avanço da tecnologia. Imagine lidar com um aplicativo ou *site* confuso, cheio de botões e menus complicados. Seria frustrante, certo? Por isso, a usabilidade é um princípio fundamental no desenvolvimento de produtos, sistemas e interfaces. Seu objetivo principal é **aprimorar a interação entre usuários e tecnologia**, tornando-a mais **eficaz, eficiente e satisfatória**. É sobre facilitar essa relação para que, assim, as pessoas alcancem os seus objetivos sem cometer erros ou gastar esforços desnecessários, o que garante uma experiência positiva.



Hora do desafio!

Pense em algum *site* ou aplicativo que você utiliza com frequência.
Depois, reflita sobre as seguintes questões:

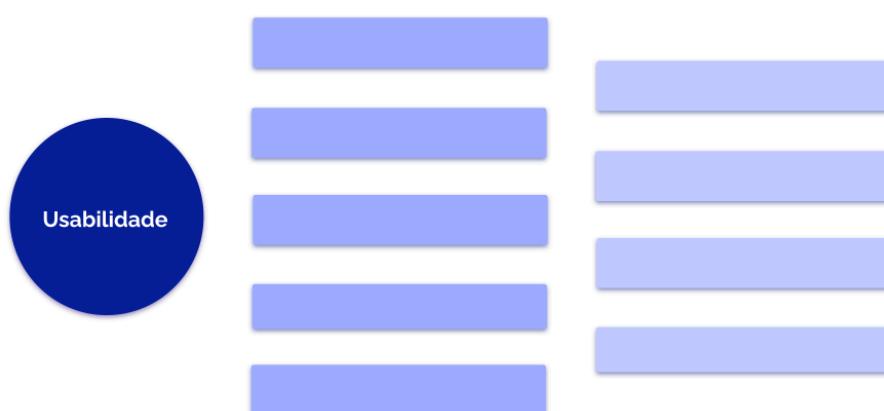
- 1 Qual é o propósito principal desse *site* ou aplicativo?
- 2 As cores e fontes utilizadas são legíveis e acessíveis?
- 3 O *site* fornece uma resposta clara sobre as ações do usuário?
- 4 O *site* ou aplicativo é acessível para pessoas com deficiências, incluindo aquelas que possuem deficiência visual, auditiva ou motora?

Percebeu como a interface de um site ou aplicativo afeta a interação do usuário?
Para facilitar a sua análise, observe alguns princípios da usabilidade:

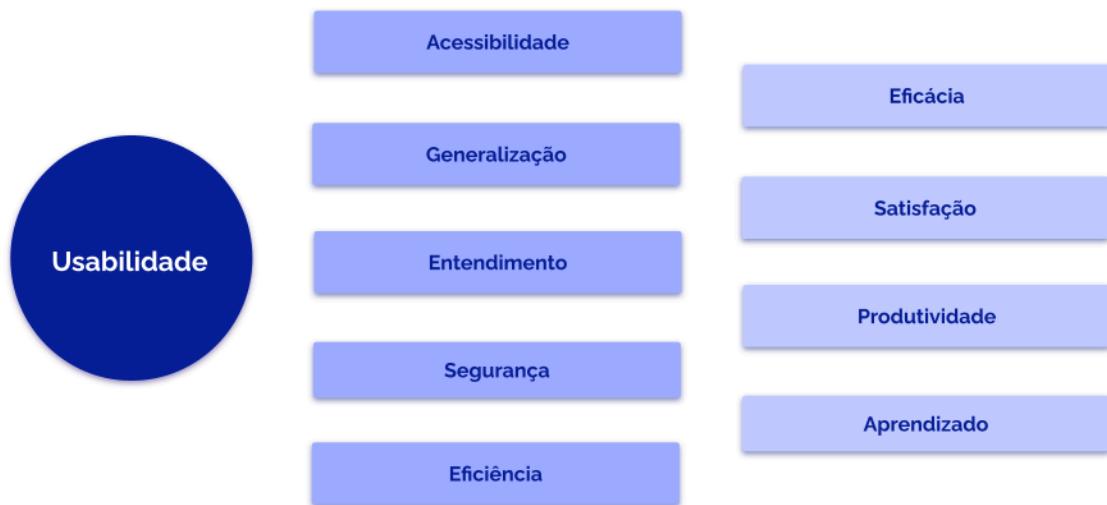
Entendimento	Garantir que os elementos da interface sejam comprehensíveis e intuitivos para os usuários, mantendo padrões visuais e de interação para facilitar a compreensão do sistema.
Eficácia	Garantir que os usuários consigam atingir os seus objetivos de forma clara e sem obstáculos. Aqui, a eficácia é medida por meio da taxa de sucesso dos usuários na realização de tarefas.
Acessibilidade	Certificar-se de que o sistema é acessível para usuários com diferentes habilidades, incluindo necessidades especiais, e garantir que o produto seja compatível com tecnologias que auxiliam usuários possuidores de deficiências.
Aprendizado	Facilitar o aprendizado inicial do usuário, tornando a curva de aprendizado mais suave. Além disso, disponibilizar documentação e tutoriais que auxiliem os usuários nesse processo.
Segurança	Projetar sistemas que minimizem a ocorrência de erros por parte dos usuários e assegurar que suas informações estejam protegidas, implementando práticas robustas de segurança.
Eficiência	Criar caminhos lógicos e eficazes para que os usuários realizem suas tarefas de maneira eficiente. Com isso, busca-se reduzir a quantidade de interações necessárias para completar uma ação, tornando o processo mais ágil.

Satisfação	Buscar criar interfaces que proporcionem uma experiência agradável e satisfatória, incorporando elementos que forneçam <i>feedback</i> positivo aos usuários durante as suas interações.
Produtividade	Fornecer funcionalidades e ferramentas que aumentem a produtividade dos usuários, projetando sistemas que minimizem o tempo necessário para realizar tarefas específicas.
Generalização	Projetar interfaces que possam ser facilmente adaptadas a diferentes contextos de uso. Assim, permite-se que os usuários as personalizem de acordo com suas preferências e necessidades.

E aí, está preparado(a) para criar uma interface que investe na usabilidade e que se preocupa com a interação do usuário? Que tal começar a observar situações do seu dia a dia nas quais esses atributos podem ser aplicados? Considerando o que você aprendeu até aqui, tente elaborar uma nuvem de palavras associadas ao conceito de usabilidade. Use o esquema abaixo para elaborar o exercício.



Veja, na imagem a seguir, uma possível resolução da atividade.



De forma geral, a usabilidade é essencial para o sucesso de produtos digitais, pois afeta, de forma direta, a experiência do usuário, a eficiência operacional e os resultados comerciais. Produtos com boa usabilidade são mais propensos a atenderem às expectativas dos usuários, resultando em benefícios para os consumidores e, também, para as empresas que os desenvolvem.

Para exercitar o que você aprendeu até aqui, resolva o desafio:

Pense fora da caixa

Considere que a rede social denominada gOLD deve ser projetada, **exclusivamente, para pessoas idosas**, proporcionando um espaço virtual **seguro e acolhedor** para elas. Nessa rede, os mais velhos podem se **conectar, compartilhar experiências, interesses** e, além disso, se **envolver em atividades sociais**.

Com base no que foi aprendido até aqui, responda: quais elementos você considera essenciais na plataforma, já que o intuito é torná-la acessível?



1.2 Padrões de interface: em busca de facilitar a compreensão do usuário

Observe a imagem a seguir.

Vamos refletir?

Tente imaginar os padrões que acontecem no seu cotidiano. Por exemplo, é provável que, após acordar todas as manhãs, você tome um banho, se vista e tome, também, um café reforçado. Essas ações **repetidas** compõem uma parte da sua rotina e te ajudam a manter a **estabilidade** e a **eficiência**.



Quando analisamos tudo aquilo que fazemos diariamente, fica claro que seguir uma rotina organizada é importante em vários aspectos. Estruturar o cotidiano não só nos ajuda a fazer as coisas de forma mais eficiente, como reduz o estresse, melhora a nossa saúde e facilita a adoção de hábitos saudáveis, nos tornando mais disciplinados e autocontrolados.

Os padrões não se limitam apenas à nossa rotina diária. Na verdade, eles desempenham papel **crucial** em diversos aspectos da sociedade. No *design* de interface, por exemplo, eles são considerados praticamente **imprescindíveis** para a organização de páginas na internet. Vamos falar mais sobre isso?

Os padrões de interface possuem o importante papel de tornar tudo mais fácil para o entendimento dos usuários. Isso inclui *layouts*, elementos visuais, interações e fluxos de trabalho, objetivando experiências **coesas** e **intuitivas** na web. Essas diretrizes ajudam os usuários a interagirem com sistemas e aplicativos de maneira mais eficiente, reduzindo a curva de aprendizado e melhorando a experiência geral.

Para saber mais sobre essa questão, assista ao vídeo a seguir:



Vídeo no YouTube *Consistência e uso de Padrões #U03 // UXNOW*, do canal UXNOW / Daniel Furtado (21 abr. 2017). Disponível em: <https://abre.ai/iPhB>. Acesso em: 5 fev. 2024.

A partir do que foi observado no vídeo acima, reflita sobre as questões a seguir.

1. Ao projetar um sistema ou aplicativo, quais são os benefícios de seguir padrões de interface?
2. O que é estado de *flow* (fluidez)?
3. Faz sentido haver diversos padrões em um produto? Dê um exemplo de boa prática de consistência em um *layout* de página.

Como discutido no vídeo, é crucial manter coesão e padrões bem definidos ao construir um produto digital para, desse modo, garantir consistência e facilitar a compreensão do usuário. Ter um menu diferente em cada página do sistema ou dividir a tela em duas cores distintas, por exemplo, pode ser confuso, o que pode, ainda, afetar a experiência geral do produto de forma negativa.

Para observar como isso ocorre na prática, confira:



Saiba mais

Nem sempre os sites famosos adotaram designs fáceis e intuitivos. Quando surgiram, algumas plataformas famosas, como Google e Facebook, apresentavam interfaces bastante distintas das que possuem hoje.

Para ver o antes e depois desses sites, acesse o QR code ao lado.



1.2.1 Alguns exemplos de boas práticas no design

O *design* permeia todos os aspectos da vida moderna, seja a interface de um aplicativo ou o *layout* de uma cidade, e a sua qualidade pode impactar a experiência individual das pessoas **significativamente**.

Por exemplo, uma cidade deve promover a conectividade entre suas diferentes partes, buscando facilitar o tráfego de pedestres, ciclistas e veículos. Isso pode incluir a **criação de ciclovias, calçadas largas, passagens subterrâneas ou elevadas** e, principalmente, uma **rede de transporte público eficiente**.

A seguir, veja o exemplo da cidade de Edimburgo, na Escócia, Reino Unido, que reúne várias dessas características:



Disponível em: <https://encurtador.com.br/gIX1q>. Acesso em: 16 fev. 2024.

Por isso, em uma página da internet, é essencial manter boas práticas no *design*. Isso ocorre não apenas por questão estética, mas para garantir que os produtos e serviços atendam às necessidades e expectativas dos usuários.

Confira algumas dicas importantes:

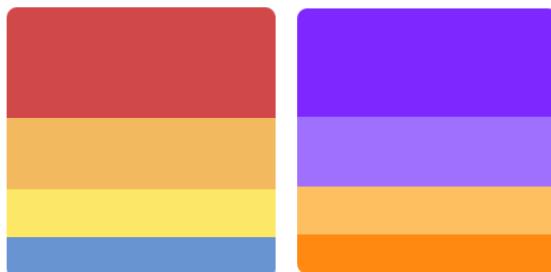
- **Paletas de cores que fazem sentido:** selecionar paletas de cores coesas é essencial no *design*, pois as cores devem harmonizar entre si de modo a manter a consistência de uso. Como, porém, escolher cores que se complementam?

Uma solução viável é utilizar o *website* colorhunt.co, que oferece opções para criar paletas de cores que fazem sentido juntas, o que inclui tons quentes e frios. Tudo depende das preferências e escolhas de cada um.

The screenshot shows the homepage of Color Hunt. At the top, there's a navigation bar with a logo, a search bar labeled "Search palettes", and an "Add to Chrome" button. Below the navigation, there are two columns of color palettes. The left column has three palettes: a yellow/green palette with 136 likes (18 hours old), a blue/purple palette with 173 likes (Yesterday), and a teal/green palette with 501 likes (2 days old). The right column has three palettes: a purple/pink palette with 136 likes (18 hours old), a dark blue palette with 173 likes (Yesterday), and a dark blue/purple palette with 501 likes (2 days old). On the far right, there's a section titled "Color Palettes for Designers and Artists" with a subtext about discovering the newest hand-picked palettes. A QR code is located on the right side of the page.

Disponível em: <https://colorhunt.co/>

Por exemplo, ao criar uma paleta de cores para o público infantil, é possível utilizar as cores a seguir.



- **Padrão de layout único:** no campo do desenvolvimento de *software*, é comum adotar apenas uma ferramenta de cada categoria. Um caso típico é o dos ícones, pois, atualmente, há uma variedade de opções disponíveis. Pode ser tentador utilizar ícones de diferentes bibliotecas, mas é importante sempre manter a consistência e padronização.



Disponível em: <https://encurtador.com.br/ivKT4>. Acesso em: 16 fev. 2024.

- **Coesão nos formulários:** em um *site*, o formulário é uma área na qual os usuários podem inserir informações ou fazer escolhas, com o objetivo de, geralmente, enviar dados de volta para o servidor. Ao desenvolver uma plataforma, procure evitar criar formulários com elementos diversos e *layouts* incompatíveis com outros formulários presentes em diferentes módulos do sistema.

Formulario com Bootstrap

Nome	
------	--

Formulario usando o padrão do bootstrap

Form com materializeCSS

	Nome		Telefone
--	------	--	----------

O que achou dos exemplos? Conseguiu perceber como pequenos detalhes fazem grande diferença quando se trata de padronização? Agora, reflita sobre os próximos tópicos.

1. Observe os exemplos anteriores. Você já identificou algumas dessas inconsistências em algum *website* ou aplicativo que utilizou?
2. Se você fosse contratado por uma empresa e se tornasse o responsável por desenvolver a interface de um produto de forma mais consistente, como aplicaria as técnicas acima?
3. Você já enfrentou dificuldades em encontrar funcionalidades específicas devido à falta de padronização em produtos digitais? Se sim, quais?

Para exercitar cada vez mais o seu conhecimento, veja a imagem.



Hora do desafio!

Analise algum aplicativo de sua escolha e siga os passos a seguir.

- 1 Observe atentamente a interface do aplicativo.
- 2 Identifique padrões de interface, como cores de botões, ícones de ação ou posição de elementos na página.
- 3 Interaja com os diferentes formulários da página (espaços para inserir informações).
- 4 Sugira possíveis melhorias ou ajustes para manter a consistência na interface do usuário.

1.3 Design centrado no usuário e em suas necessidades

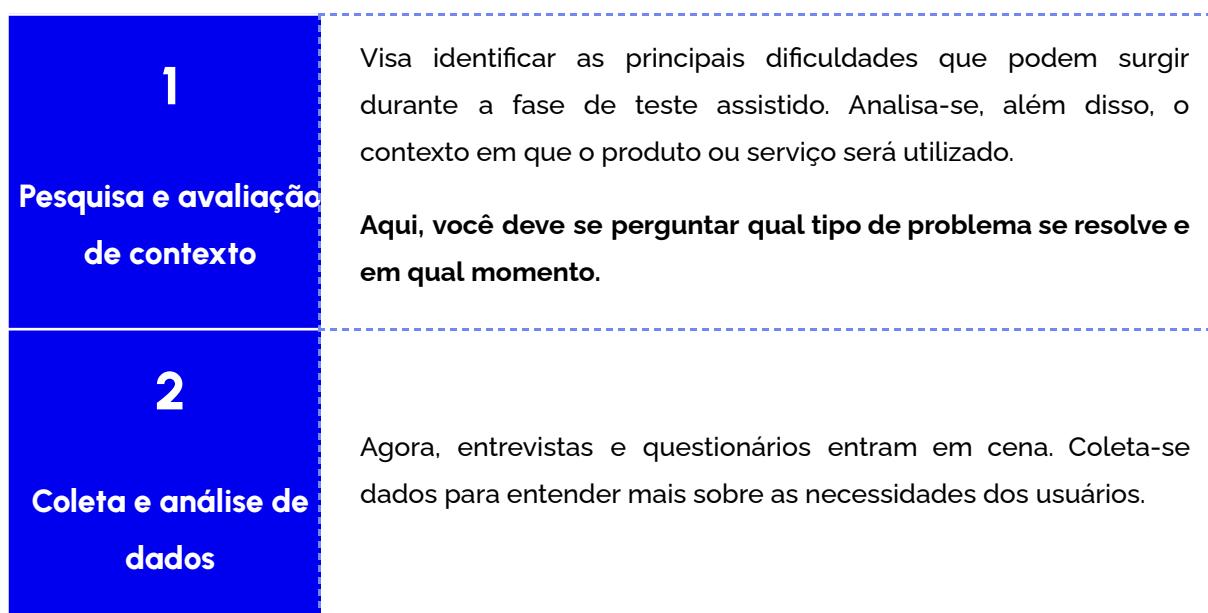
Ao desenvolver um produto, mais importante do que pensar na **maneira** como você prefere o material, é essencial refletir sobre a forma como o **usuário** se relacionará com ele. Embora pareça simples, essa abordagem tem grande importância, pois coloca as necessidades e experiências do usuário como foco principal do processo de design.

Os **sistemas operacionais** são exemplo disso. Como são feitos para **boa parte** das pessoas utilizarem, eles precisam ser fáceis e de uso simples. Por isso, apostam em ícones na tela inicial, gestos de toque e menus de navegação.

Dessa forma, o **Design Centrado no Usuário** (DCU) emerge como uma abordagem fundamental na concepção e aprimoramento de interfaces digitais e produtos, colocando os **usuários no centro do processo criativo**.

Esse método, fundamentado no profundo entendimento das necessidades, características e preferências dos usuários, busca não apenas satisfazer, mas, também, superar as expectativas. Desde a fase inicial de pesquisa até a implementação, passando, ainda, pelas iterações contínuas, o objetivo é proporcionar experiências significativas, intuitivas e adaptáveis.

Agora, vamos explorar as **etapas cruciais desse processo**, revelando os passos que transformam dados em soluções envolventes. Isso tudo resulta em interfaces que vão além da funcionalidade, pois se tornam genuinamente atrativas.



3 Criação de personas e cenários de uso	<p>Começa-se segmentando os usuários em grupos para entender suas diferentes necessidades e comportamentos. Os cenários de uso ilustram as diversas formas como esses usuários interagem com a interface.</p>
4 Idealização e geração de conceitos	<p>É a hora de nos reunirmos para compartilhar o que aprendemos. Em sessões de <i>brainstorming</i>, exploramos ideias para resolver os problemas e atender às necessidades identificadas.</p>
5 Início da prototipação e testes de usabilidade	<p>A equipe técnica entra em ação para criar protótipos de baixa fidelidade e conduzir testes de usabilidade com usuários reais. Uma vez validados, é o momento de avançar para os protótipos de alta fidelidade.</p>
6 Avaliação heurística e revisão de design	<p>Criar caminhos lógicos e eficazes para que os usuários realizem suas tarefas de maneira eficiente, reduzindo a quantidade de interações necessárias para completar uma ação e tornando o processo mais ágil.</p>
7 Testes-piloto e validação	<p>Antes de iniciarmos o desenvolvimento completo, realizamos testes-piloto finais com um grupo menor de usuários. Assim, obtemos <i>feedbacks</i> finais.</p>
8 Desenvolvimento e implementação	<p>Agora, trabalhamos em estreita colaboração entre desenvolvedores e <i>designers</i> para garantir a consistência na execução e construção do produto final.</p>

9

Lançamento e monitoramento pós-implementação

O produto está pronto para ser lançado! Aqui, monitoramos as reações dos usuários de perto, garantindo que tudo esteja conforme o planejado.

10

Iteração contínua e melhorias

O trabalho não acaba com o lançamento. Coletamos dados constantemente, visando garantir o sucesso do produto, e aplicamos melhorias contínuas, tomando como base o *feedback* e o desempenho real dos usuários.

E aí, gostou? Com o *Design* Centrado no Usuário, é possível priorizar a compreensão das suas necessidades, envolvendo-os, de forma ativa, no processo de *design*. É assim que são desenvolvidos serviços que verdadeiramente fazem a diferença na vida das pessoas.

A partir do que você estudou, responda às perguntas a seguir. Ponha a mão na massa e exerçite tudo o que viu até então!



Qual fase desse processo mais te interessou? Você sabia que existia tanta coisa envolvida no desenvolvimento de um *software*?



Consegue identificar a fase mais importante nesse processo?



Qual a importância de monitorar o desempenho pós implementação?



O que é análise de contexto?



Com quem são feitos os testes piloto?

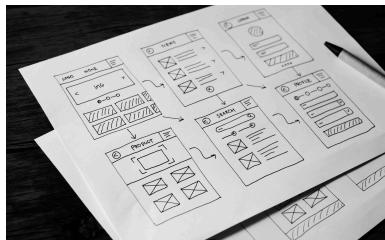
1.4 Introdução ao wireframing e à prototipagem

Você sabe qual é a importância do *wireframing* e da prototipagem no desenvolvimento de interface? Para simplificar os dois conceitos, analise o exemplo a seguir.

Imagine que você está construindo uma casa. Antes de começar a construção, é útil ter um esboço básico do *layout* da residência para saber onde ficarão os quartos, a sala de estar, a cozinha etc.

Agora, imagine que você possui o esboço básico da casa, mas que deseja ter uma ideia mais clara de como será o seu lar antes de começar a construção propriamente dita. Você decide, então, construir um modelo em escala reduzida da moradia, usando materiais mais simples.

A partir do exemplo, veja como o *wireframing* e o protótipo se relacionam.



VS

Wireframing

É o esboço do *layout* da casa. É o processo de criar um **rascunho básico** ou uma **representação visual** de uma página da web, aplicativo ou produto digital. Geralmente, é feito usando lápis e papel ou software de *design* gráfico.

Protótipo

É o modelo da casa reduzida. No *design*, é o processo de criar uma **versão simplificada e interativa** de um produto digital que permite testar funcionalidades, fluxos de usuário e interações, mas antes do desenvolvimento completo.

Para se aprofundar ainda mais no assunto, veja o próximo vídeo.



Vídeo no YouTube *O que é um Wireframe? //UXNOW*, do canal UXNOW / Daniel Furtado (22 set. 2016).

Disponível em: <https://bit.ly/47Zqzvq>. Acesso em: 30 jan. 2024.

A partir do que você observou no vídeo, tente responder:

1. Qual é o objetivo de um *wireframe* e quais tipos de informações podemos extrair dele?
2. Quais são os principais elementos que devemos observar para construirmos um *wireframe* de qualidade?
3. Por que *wireframes* não possuem cores?
4. O que é hierarquia?

1.4.1 Ferramentas de criação para prototipagem e *wireframing*

Agora que você conheceu um pouco mais sobre a definição de prototipagem e *wireframing*, chegou a hora de conhecer as principais ferramentas para desenvolvê-los.

Confira cada uma delas a seguir.

1. Com planos gratuitos

Figma: é uma ferramenta de *design* colaborativo baseada na web. Tem uma versão gratuita com recursos robustos para prototipagem e *wireframing*.

Balsamiq: é conhecido por seu foco em *wireframing* rápido e fácil, e têm uma versão gratuita *on-line* que oferece recursos básicos.

Adobe XD: é uma ferramenta de *design* e prototipagem da Adobe. Eles oferecem uma versão gratuita com recursos limitados, mas ainda úteis para projetos simples.

Sketch Cloud: é amplamente utilizado no *design* de interfaces, e possui recursos limitados em comparação à versão paga.

2. Ferramentas pagas

Axure RP: é uma ferramenta poderosa para criação de *wireframes* e protótipos interativos. É uma opção paga, mas oferece recursos avançados.

Sketch Premium: embora tenha uma versão gratuita *on-line*, o Sketch oferece uma versão paga com recursos mais avançados e é muito popular entre os *designers* de interface.

Adobe XD Premium: a versão paga do Adobe XD inclui recursos mais avançados, como colaboração em equipe e integração com outras ferramentas da Adobe.

InVision: oferece planos pagos que incluem recursos mais avançados para prototipagem e colaboração em equipe.

Sketch: oferece uma ampla gama de recursos e funcionalidades que permitem aos *designers* criarem *wireframes* de alta qualidade de forma eficiente

1.4.2 Tipos de protótipos e suas definições

Na criação de um produto, não basta **apenas criar um protótipo**. Durante o seu desenvolvimento, é essencial analisar **de que modo** ele será feito.

Existem diversos tipos de protótipos, cada um com suas próprias características e finalidades específicas, adaptados para atender às necessidades e estágios diferentes do processo de *design* e desenvolvimento de produtos digitais. Neste tópico, vamos examinar os diferentes tipos de protótipos e suas definições, destacando como cada um deles pode ser utilizado para facilitar o processo de *design* e o desenvolvimento de produtos digitais. Para facilitar a compreensão, utilizaremos exemplos para **ilustrar** cada um dos estágios.

	Alta fidelidade	Baixa fidelidade	Viabilidade
Definição	São representações visuais e interativas de um produto digital. Assemelham-se muito à aparência e ao comportamento final do produto.	São representações simplificadas e abstratas de um produto digital, destacando, principalmente, a estrutura e o fluxo. Não há preocupação com detalhes visuais refinados.	Implementações mínimas de um produto ou funcionalidade. Frequentemente, são conhecidos como MVPs (Produto Mínimo Viável).
Características	Incluem elementos gráficos, cores, fontes e imagens semelhantes ao <i>design</i> final. Permitem simular interações complexas e proporcionam uma experiência próxima da realidade.	Geralmente, são desenhados à mão ou criados rapidamente em ferramentas de wireframing. São úteis para validar conceitos, fluxos e hierarquia de informações.	Incluem apenas as características mais críticas para avaliação e validação, visando economizar tempo e recursos.

Para compreender um pouco mais sobre como isso é feito na prática, observe como os diferentes tipos de protótipo atuariam no desenvolvimento de uma **plataforma de streaming** de séries e filmes.

E se fosse em uma plataforma de séries e filmes?

Alta fidelidade	Baixa fidelidade	Viabilidade
Protótipo digital criado no Adobe XD ou Figma que mostraria elementos visuais mais detalhados, como ícones, cores e tipografia. Transições suaves entre telas e interações simuladas, como clique em botões e rolagem de páginas, estariam inclusos.	Esboço rápido em papel mostrando a disposição geral da interface, incluindo algumas seções, como página inicial, categorias de conteúdo, barra de pesquisa e reprodução de vídeo.	O protótipo teria uma interface de usuário simples. Haveria os recursos mínimos necessários para assistir a vídeos para, então, verificar a taxa de possibilidade de desenvolvimento da ideia.

Após compreender um pouco melhor a **importância dos protótipos**, você pode se perguntar qual é o mais adequado a se escolher. A resposta é que escolher o melhor tipo de protótipo para a sua ideia depende de vários fatores, incluindo o estágio do projeto, os objetivos específicos de *design* e as necessidades do público-alvo.

Entretanto, alguns tipos de protótipos possuem **diversas características**. Confira:



Alta fidelidade

- Semelhantes ao produto final;
- Custam mais caro e demoram mais para serem construídos.



Baixa fidelidade

- Não se assemelha ao produto final;
- Criados para servir como base para protótipos de alta fidelidade;
- Custam menos.



Viabilidade

- Novas tecnologias;
- São constantemente aprimorados;
- Tratam de riscos durante a descoberta de produtos.

1.4.3 Vamos testar protótipos de baixa fidelidade?

Gostou do que aprendeu até aqui? Qual foi o tópico que mais lhe chamou a atenção? Ao longo deste tema, você pôde observar a **importância dos protótipos** no desenvolvimento de determinado produto. Com eles, é possível testar e validar as suas ideias antes de investir recursos no desenvolvimento completo do produto.

Entretanto, mais do que isso, é essencial haver investimento na **avaliação da satisfação do público**. Para validar algumas ideias dos protótipos de baixa fidelidade, é possível realizar pesquisas com possíveis futuros usuários ou *stakeholders* (colaboradores em geral do projeto). Assim, busca-se legitimar as nossas ideias e verificar, com o auxílio de opiniões externas, o que agregará ou não valor ao nosso produto.

A seguir, observe um exemplo de pesquisa de satisfação a respeito de um produto que está sendo desenvolvido e que pretende ser inovador em relação às demais **opções de assinaturas** existentes no mercado.

Formulários Google



Formulário desenvolvido pelo autor (2024).

Disponível em: <https://forms.gle/FPJa2drQzJKYARP56>. Acesso em: 29 fev. 2024.

Essas informações são muito importantes para as empresas que criam produtos digitais, porque ajudam a melhorar a experiência dos usuários. Ao analisarmos os problemas que enfrentamos no dia a dia, podemos ter ideias valiosas. Muitas vezes, ao resolver esses problemas comuns, surgem ideias para novos produtos que podem surpreender e melhorar a vida das pessoas.



Chamada para ação

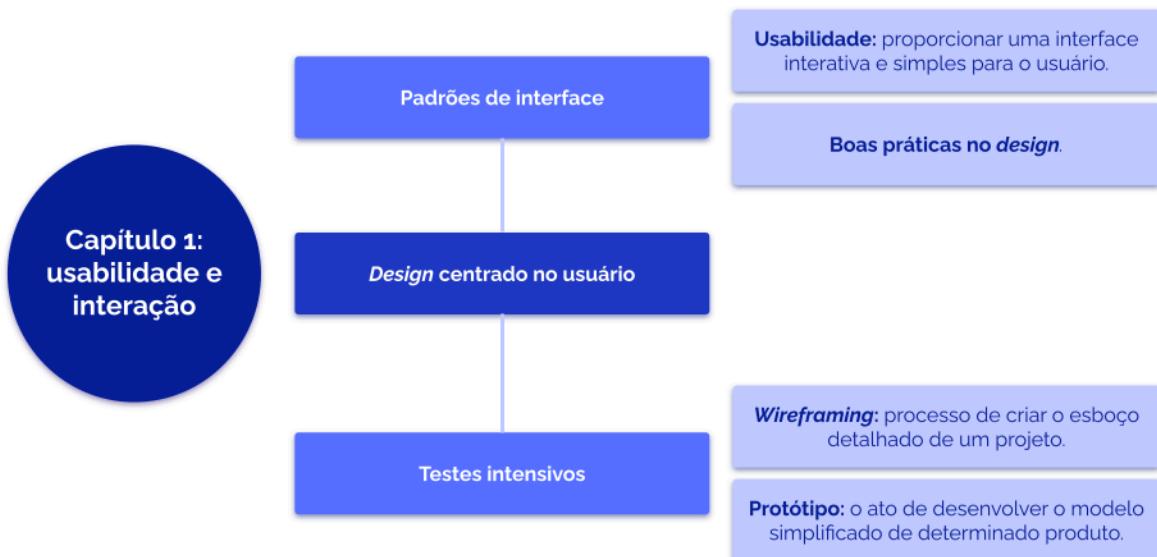
Que tal um desafio?

Nesta atividade, será preciso desenvolver uma ideia para um **aplicativo de entrega especializada em café**. Para isso, você deverá criar o esboço inicial e, se conseguir, aplicar os elementos em uma ferramenta de *wireframing* de sua escolha!



Neste tema, abordamos elementos fundamentais do *design* de produtos digitais. Inicialmente, foi apresentada a definição de **usabilidade** e a sua **crucial** importância na concepção de produtos digitais, destacando a necessidade de criar interfaces que proporcionem experiências de uso eficientes e satisfatórias. Enfatizamos, além disso, a importância de manter padrões ao longo da interface, promovendo a **facilidade de compreensão** por parte dos usuários. Ao estabelecer consistência visual e de interação, busca-se otimizar a navegação e tornar a experiência do usuário mais intuitiva.

Exploramos o processo de **Design Centrado no Usuário**, que se concentra em compreender as suas **necessidades, características e preferências**, e, além disso, passamos pela conceituação de duas ferramentas essenciais: prototipagem e *wireframing*. Ambas desempenham papéis cruciais no desenvolvimento de produtos digitais, permitindo a criação de modelos interativos e esboços estruturais que ajudam na visualização e avaliação do *design*. Agora, veja como todos esses conceitos se relacionam.





ATIVIDADE DE FIXAÇÃO

- 1.** Você já percebeu que a prevenção de erros pode ajudar a usabilidade das interfaces que você utiliza? Como, na sua opinião, isso afeta cada experiência?
- 2.** Para você, o que significa inclusividade no contexto de usabilidade? Você já teve alguma experiência positiva ou negativa em relação ao assunto?
- 3.** Por que é importante que os *designs* sigam padrões específicos? Você já teve alguma experiência na qual a falta de padronização prejudicou a sua experiência enquanto usuário?
- 4.** Você já se viu frustrado ao preencher formulários *on-line*? Por que os formulários são tão importantes nos sistemas que utilizamos?
- 5.** Concorda que é mais eficiente desenvolver, primeiro, um *software* e, então, verificar se ele atende às necessidades dos clientes? Justifique a sua resposta.
- 6.** Você acredita que a comunicação constante entre desenvolvedores e *designers* é fundamental para o sucesso de um projeto? Por quê?
- 7.** Durante a fase de coleta e análise de dados do *Design Centrado no Usuário*, que tipo de informações você considera mais importantes de serem absorvidas, visando garantir um *design* eficaz?
- 8.** Qual é a diferença entre protótipos de alta e baixa fidelidade? Forneça um exemplo para cada tipo e explique quando cada um deles é mais útil.
- 9.** Forneça uma característica específica de um protótipo de viabilidade que o torna útil para testar a ideia de determinado produto.
- 10.** Como você acha que podemos garantir que os nossos protótipos atendam às necessidades dos usuários? Quais, na sua opinião, são os métodos de validação mais eficazes?

CAPÍTULO 02

DO DESIGN À IMPLEMENTAÇÃO: ESTRATÉGIAS TÉCNICAS PARA COMUNICAR, ADAPTAR E REFINAR INTERFACES RESPONSIVAS

O que esperar deste capítulo:

- Comunicar o *design* para desenvolvedores e partes interessadas;
- Princípios do *design* responsivo: adaptando *layouts* e elementos para dispositivos móveis;
- Uso de *media queries* em CSS;
- Testes em diferentes tamanhos de tela: adaptando *designs* para dispositivos móveis;
- Testes de usabilidade e análise de métricas de UX;
- Coleta de *feedback* dos usuários;
- Iteração de *design* com base em *feedbacks*.

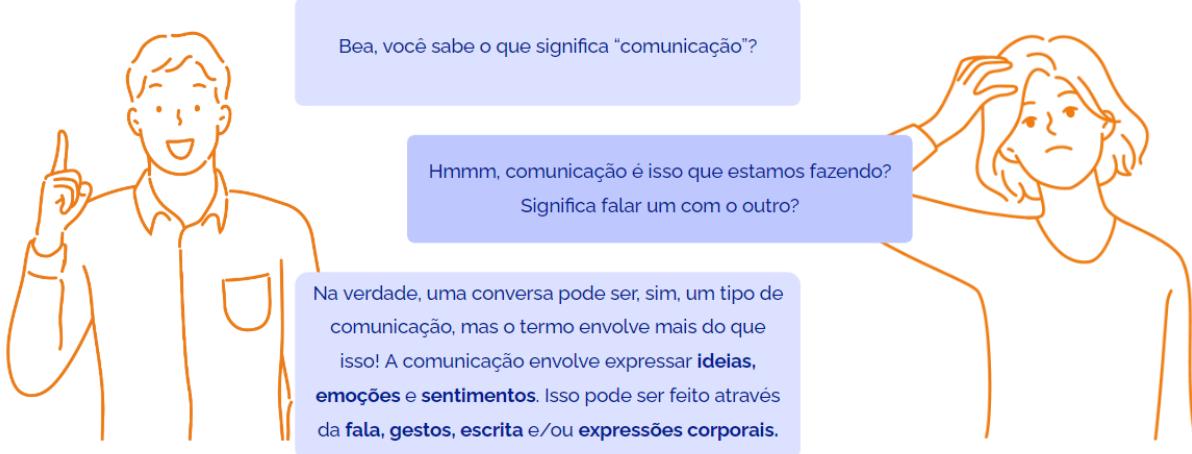
O processo de desenvolvimento de produtos digitais

Ao desenvolver a ideia de um aplicativo, você já sentiu dificuldades em criar um estilo único e interativo para o seu projeto? Se sim, como lidou com o problema?

Para superar essa e outras dificuldades associadas à criação de projetos digitais, vamos tratar do assunto de maneira prática. Aqui, abordaremos temas acerca da responsividade, *layouts*, análise de métricas e CSS avançado. Ao aplicar esses conhecimentos em nossos próprios projetos, faremos questão de enfatizar a importância da colaboração entre equipes de desenvolvimento e *design* para assegurar a excelência do produto final.

2.1 Comunicando design para desenvolvedores e partes interessadas

Observe a interação a seguir.



Considerando a imagem anterior, você consegue indicar qual é a diferença entre "comunicação" e "fala"? Pensando no dia a dia, quais situações corriqueiras envolvem a comunicação?

Embora seja essencial para o funcionamento das nossas vidas em sociedade, a comunicação pode ser encontrada em diversos setores, sendo imprescindível durante o desenvolvimento de projetos digitais. Mas por que é dada tanta importância a ela, afinal?

Reiteradamente, é importante garantir que todos na equipe entendam o que está sendo comunicado, não importa quanto conhecimento técnico os membros do time tenham. Quando falamos sobre o *design* para desenvolvedores e outras pessoas interessadas, usar uma **linguagem simples** é fundamental. Isso ajuda as pessoas entenderem a ideia por trás do *design* de maneira consistente, mesmo aqueles que não são especialistas em tecnologia. É por isso que a comunicação é indispensável!

É crucial criar um ambiente onde todos se sintam à vontade para compartilhar suas ideias, já que cada pessoa na equipe possui muito a contribuir. Além disso, usar documentos visuais torna mais fácil discutir e entender o *design*, especialmente considerando aqueles que não estão familiarizados com termos técnicos.

Veja como a boa comunicação pode ser **a chave** para resolver desafios encontrados por equipes.

Durante a criação de um aplicativo de entrega de alimentos, a equipe percebe que os usuários estão tendo problemas com a opção "Adicionar ao carrinho", o que resulta em uma perda significativa e cada vez mais frequente de clientes. Diante dessa situação, o que fazer?

1 Resolver o problema o mais rápido possível para diminuir gastos.

Não garante uma boa comunicação, pois pode resultar em confusões entre a equipe.

2 Criação de vários protótipos, mesmo que leve tempo.

Garante uma boa comunicação, pois cada um pode explorar diferentes abordagens.

3 Reunião com a equipe para discutir os problemas encontrados.

Garante uma boa comunicação, pois dá espaço para o compartilhamento de ideias.

Ficou curioso e quer se manter por dentro da importância da comunicação entre a equipe de *designers* e desenvolvedores? Confira o vídeo a seguir.



Vídeo no YouTube *Designer, você PRECISA aperfeiçoar a sua comunicação!*, do canal Chief of Design (15 set. 2021).

Disponível em: <https://abre.ai/iQac>. Acesso em: 5 fev. 2024.

A partir do que foi assistido, responda:

1. Como a falta de comunicação clara pode impactar os relacionamentos interpessoais e o desempenho no trabalho?
2. Quais são os desafios enfrentados ao comunicar ideias complexas ou conceitos técnicos para um público não especializado?
3. Quais são os benefícios de uma comunicação aberta e transparente em uma equipe?

2.2 Design responsivo: adaptação de layouts para dispositivos móveis

Agora que já aprendemos um pouco mais sobre comunicação entre equipes, vamos nos aprofundar no *design* responsivo. Você sabe o que é isso? Para entender mais, veja o exemplo a seguir.

Imagine que você possui um livro. Agora, pegue-o e abra em uma mesa grande. É fácil ler o que está escrito, certo? Depois disso, leve o mesmo livro até uma mesa pequena e tente abri-lo. É um pouco mais difícil fazer a leitura, não é?



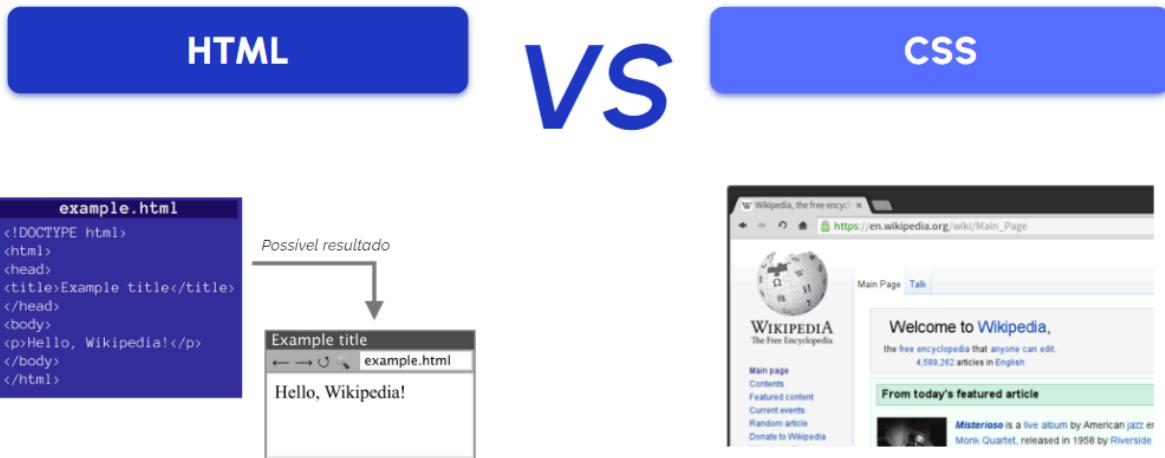
Disponível em: <https://encurtador.com.br/cjsuZ>. Acesso em: 19 fev. 2024.

O *design* responsivo pode ser comparado a um livro que se adapta automaticamente ao tamanho da mesa em que é colocado. Se a mesa é grande, o livro se abre confortavelmente, facilitando a leitura. Por outro lado, se a mesa é pequena, o livro ajusta-se de forma automática, garantindo que você ainda possa ler sem dificuldades. Um site com *design* responsivo funciona da mesma maneira: não importa se você vai acessá-lo em um computador grande, um *tablet* médio ou um *smartphone* pequeno, porque a página se adapta ao tamanho da tela de forma automática. Assim, você pode ver e interagir com o site sem que o dispositivo prejudique a experiência.

Conseguiu compreender como o processo ocorre? Com o *design* responsivo, os usuários podem usufruir da melhor experiência ao navegar por diferentes plataformas. Para nos aprofundarmos no tema, vamos falar um pouco sobre as ferramentas que utilizaremos: **HTML e CSS**.

 HTML (Hypertext Markup Language - Linguagem de Marcação de Hipertexto)	É a linguagem fundamental para a criação e estruturação de páginas <i>web</i> . O HTML é utilizado para definir a estrutura e os elementos de uma página, permitindo a criação de links , imagens , parágrafos , formulários e outros elementos interativos. Ele age como o esqueleto de uma página <i>web</i> , fornecendo a base sobre a qual o CSS e JavaScript podem atuar para criar uma experiência completa.
 CSS (Cascading Style Sheets - Cascata de Folhas de Estilo)	É uma linguagem de estilo utilizada para descrever a apresentação de documentos HTML (ou XML). Com o CSS, os desenvolvedores têm o poder de controlar o layout , as cores , as fontes e outros aspectos visuais de uma página <i>web</i> .
 IDEs (Integrated Development Environments - Ambientes de Desenvolvimento Integrado)	IDEs, ou Ambientes de Desenvolvimento Integrado (em inglês, <i>Integrated Development Environments</i>), são ferramentas de <i>software</i> que oferecem um conjunto abrangente de recursos para desenvolvedores durante o processo de criação, edição e depuração de softwares. Essas ferramentas buscam proporcionar um ambiente integrado que simplifica e otimiza o desenvolvimento de softwares.

Para entender como ocorre a interação entre o HTML e o CSS, veja o exemplo.



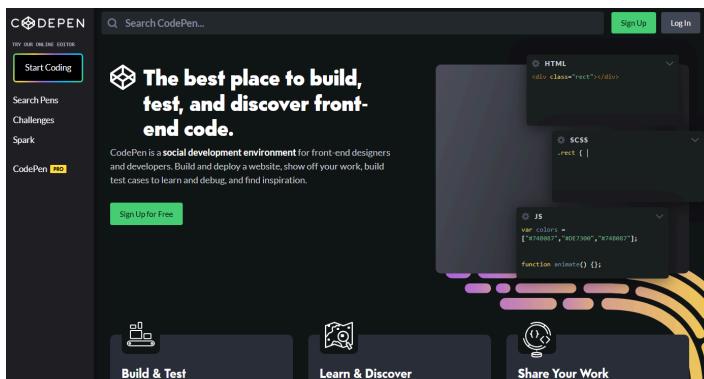
Nesse sentido, enquanto a estrutura HTML é utilizada para definir a **base** de uma página *web*, estabelecendo **parágrafos**, **links** e **listas**, o CSS é usado para determinar a **estética visual do site**, ou seja, incluir fontes, cores e a logotipo da plataforma.

2.2.1 Ferramentas de desenvolvimento

Agora que você já aprendeu sobre design responsivo, chegou a hora de praticar! Para tanto, considere as opções a seguir.

- **CodePen**

O website **codepen.io** é um editor *on-line* que possibilita codificar diretamente nas linguagens HTML, CSS e JavaScript. Utilizá-lo é simples: basta criar uma conta, já que o acesso é totalmente gratuito. A seguir, você encontra um QR code para facilitar o acesso à página.



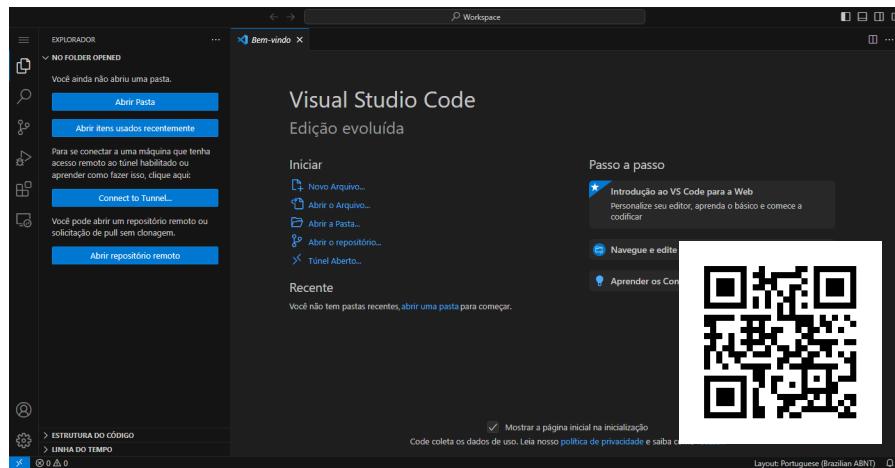
CODEPEN



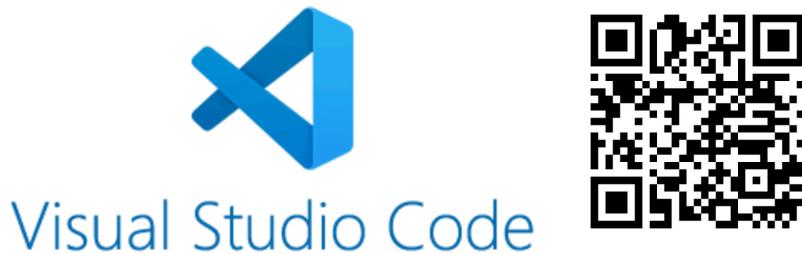
CODEPEN. Home. Disponível em: <https://codepen.io>. Acesso em: 5 fev. 2024.

- **Visual Studio Code**

O Visual Studio Code é um editor de código-fonte desenvolvido pela Microsoft que oferece uma variedade de recursos para programadores. Ele possui duas versões: a *web* e a *desktop*. Para **acessar a versão web**, acesse o QR code disponível na imagem.



Para **obter a versão desktop**, acesse o QR code a seguir.



VISUAL STUDIO CODE. *Download*. Disponível em: <https://code.visualstudio.com/download>.

Acesso em: 5 fev. 2024.

Para facilitar o *download*, a instalação e a configuração em seu computador, veja o **tutorial em vídeo**.



Vídeo no YouTube *Como baixar e instalar o Visual Studio Code (VS Code) - 2023*, do canal Logic with Ed (12 jul. 2022).

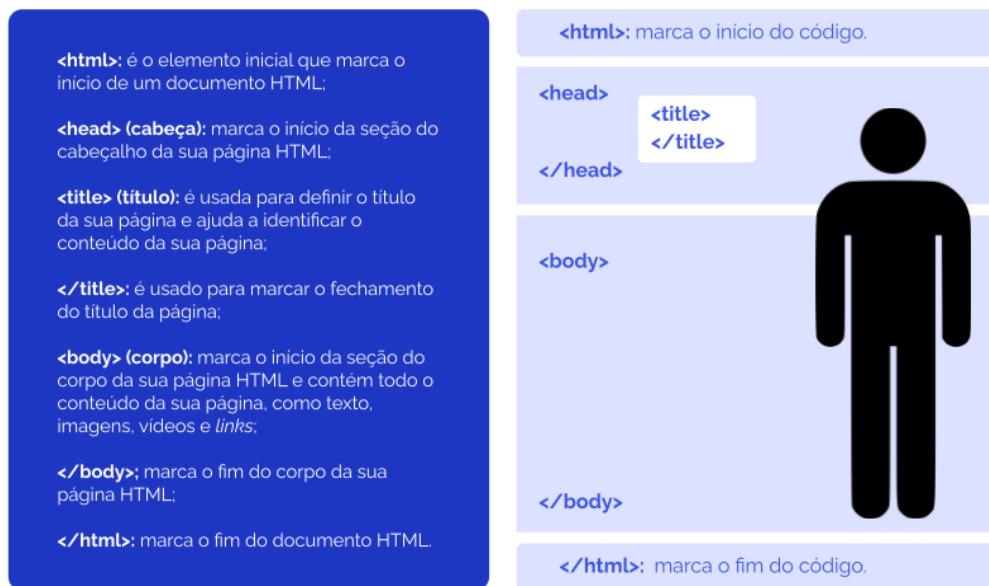
Disponível em: <https://abre.ai/iWS2>. Acesso em: 19 fev. 2024.

2.2.2 Entendendo a estrutura de uma página e sua responsividade

Conhecendo a estrutura básica do código

Agora que você já conhece algumas plataformas para desenvolver códigos, chegou a hora de pôr a mão na massa! Vamos lá?

Antes de partir para a prática, é essencial que você conheça a **estrutura básica** do seu código. Observe a imagem a seguir.



Para observar como isso ocorre na prática, observe uma página HTML de filmes de ação e analise como as diferentes *tags* são utilizadas.

Na prática

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Lista de Filmes de Ação</title>
</head>
<body>
    <h1>Lista de Filmes</h1>

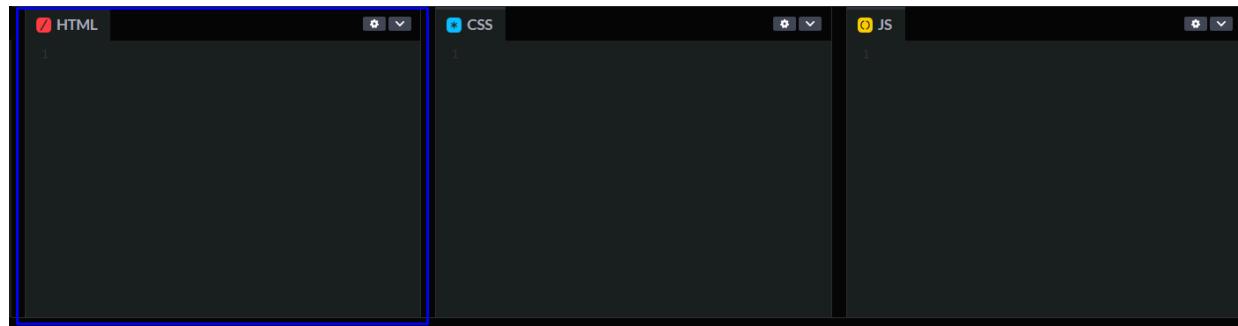
    <div>
        <h2>Filmes de Ação</h2>
        <ul>
            <li>Matrix</li>
            <li>Mad Max: Estrada da Fúria</li>
        </ul>
    </div>

</body>
</html>
```

E aí, conseguiu compreender um pouco mais sobre a estrutura básica do HTML? Que tal praticar um pouco?

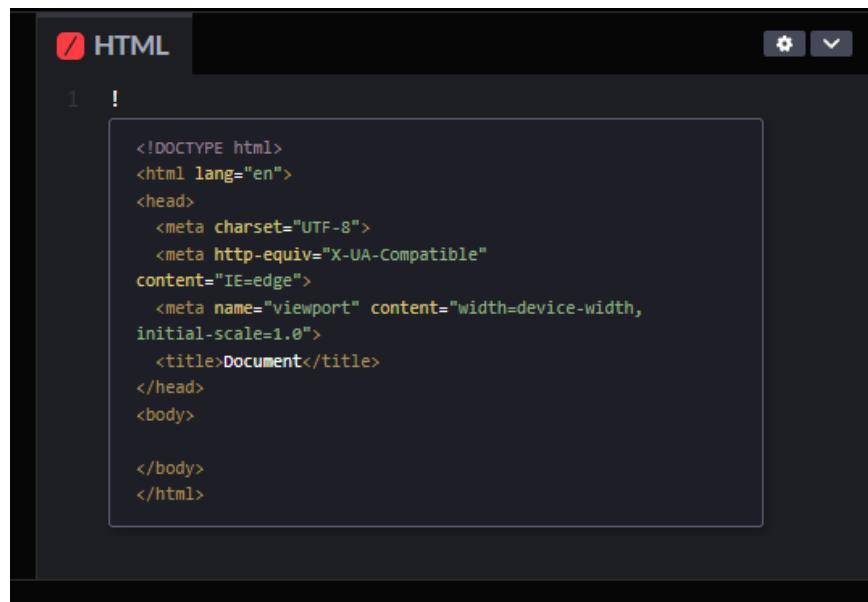
Para isso, vamos nos concentrar nas informações presentes na tag <body>.

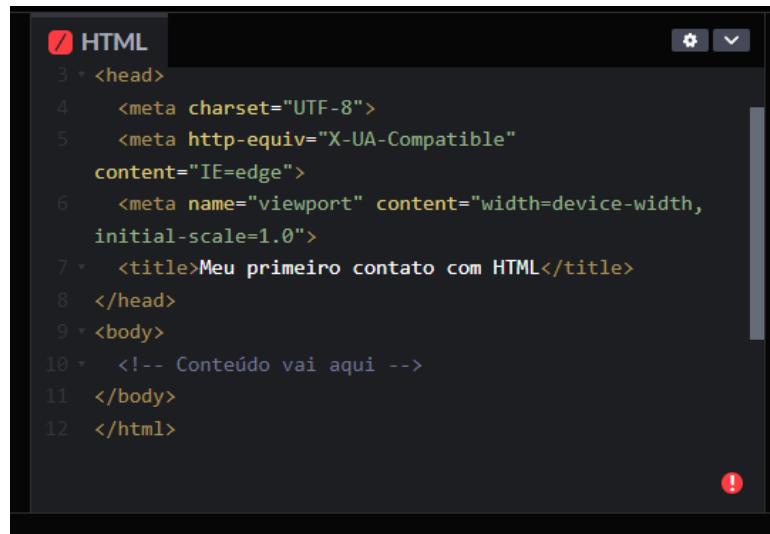
Após se cadastrar ou efetuar o *login* no **CodePen**, abra a página inicial da plataforma e utilize a aba intitulada "HTML".



Em seguida, pressione a tecla "!" e, depois, a tecla "**tab**", preenchendo o esqueleto básico do HTML. Esse é um atalho muito utilizado para autocompletar o documento HTML, estando presente, também, em diversas IDEs.

Observe a imagem a seguir. Ela corresponde ao que você verá na sua tela.





```
HTML
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible"
6     content="IE=edge">
7   <meta name="viewport" content="width=device-width,
8     initial-scale=1.0">
9   <title>Meu primeiro contato com HTML</title>
10  </head>
11  <body>
12    <!-- Conteúdo vai aqui -->
13  </body>
14 </html>
```

Resetando no CSS

O CSS é responsável por definir como os elementos HTML são exibidos na tela, o que inclui *layout*, cores, fontes, espaçamento e outros aspectos visuais. Com o CSS, é possível controlar a aparência e o *design* de uma página da web de forma separada do conteúdo. Isso possibilita maior flexibilidade e organização.

Na atividade prática, vamos utilizar o conceito de *reset* (resetar) no CSS, o que corresponde ao ato de **retirar os espaçamentos e configurações padrões** do *layout*, inserindo as configurações da imagem a seguir. Com ela, é possível remover as **margens, espaçamentos, tamanhos de fonte, alinhamentos e bordas indesejadas**. Assim, torna-se possível trabalhar em um **template do zero**.

Tente reproduzir em seu programa!



```
*{
  margin: 0;
  padding: 0;
  border: 0;
  font-size: 100%;
  font: inherit;
  vertical-align: baseline;
}
```

Observando o modelo, podemos notar que é possível alterar *margin* (margem), *border* (borda) e outros aspectos estéticos da página.

Unidades e medidas utilizadas no CSS

Como a linguagem CSS é responsável pela **estética** da página, há uma variedade de unidades e medidas que podem ser usadas para definir dimensões, tamanhos, espaçamentos e outras propriedades. Com a tabela a seguir, você conhecerá as **principais** dimensões para ajudar o seu desenvolvimento web. Confira!

Px (Pixels)

Unidade que representa o tamanho de um elemento ou medida em *pixels* na tela. É uma das mais comuns, juntamente com **porcentagens** e **medida rem**.

Rem (Root Em)

É uma medida amplamente adotada, porque ajusta os tamanhos na tela de acordo com a medida padrão do navegador do usuário configurada no navegador. É uma prática acessível, especialmente para pessoas com dificuldades visuais, pois adapta os elementos para facilitar a visualização.

VH (View Height)

Representa a altura da página do dispositivo em questão. Imagine, por exemplo, que um dispositivo possui a resolução 1280 x 800 (altura/largura). Com a informação de que a altura da tela é de 1280 pixels, a interface do usuário pode se redimensionar de acordo com regras preestabelecidas.

% (Porcentagem)

É uma das principais funcionalidades do CSS. Permite medir tamanhos e utilizar unidades de medida como porcentagem para, conforme necessário, adaptar a tela.

Para saber um pouco mais sobre como utilizar cada uma delas, confira a página a seguir.



Camilo Micheletto

Posted on 17 de ago. de 2021 • Updated on 19 de ago. de 2021



31



4



3

Entendendo unidades CSS e quando utilizá-las

#css #iniciante



MICHELLETO, Camilo. *Entendendo unidades CSS e quando utilizá-las*. DEV Community, 17 ago. 2021.

Disponível em: <https://encurtador.com.br/ehDKP>. Acesso em: 19 fev. 2024.

2.3 A utilização de media queries

Media queries são partes fundamentais do *design responsivo* em desenvolvimento web. São utilizadas em CSS para aplicar estilos diferentes, tomando como base as características do dispositivo ou da mídia em que a página está sendo exibida.

Com *media queries*, os desenvolvedores podem criar *layouts* e estilos adaptativos para diferentes tamanhos de tela, orientação (horizontal ou vertical) e tipos de dispositivos, como *smartphones*, *tablets*, *laptops* e *desktops*. Isso permite que o conteúdo da página se ajuste automaticamente, proporcionando uma experiência de usuário otimizada em vários dispositivos.

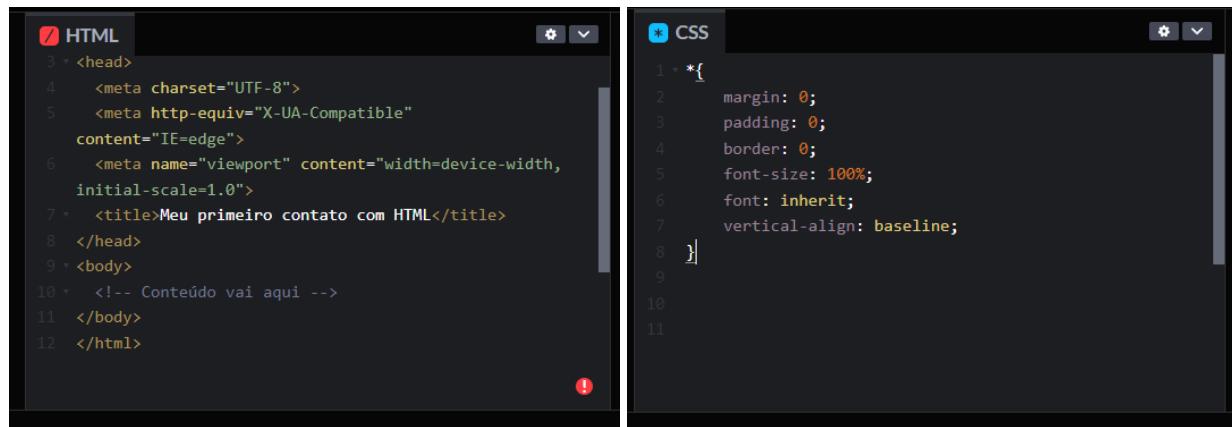
As *media queries* funcionam verificando determinadas condições, como a **largura da tela**, a **resolução**, a **orientação** ou a **capacidade do dispositivo**, aplicando estilos CSS específicos quando essas condições são atendidas.



Disponível em <https://encurtador.com.br/qGSU8>. Acesso em: 19 fev. 2024.

Vamos voltar ao nosso projeto anterior e ver um exemplo na prática para, assim, entender como as *media queries* funcionam. Nosso objetivo é construir uma Navbar (*navigation bar*, ou, em tradução livre, barra de navegação).

Veja, adiante, o nosso progresso até agora.



The screenshot shows a code editor interface with two tabs: 'HTML' and 'CSS'. The 'HTML' tab contains the following code:

```
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Meu primeiro contato com HTML</title>
</head>
<body>
<!-- Conteúdo vai aqui -->
</body>
</html>
```

The 'CSS' tab contains the following code:

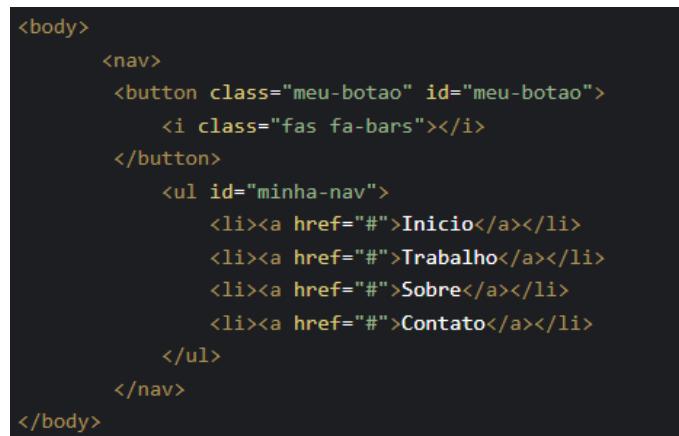
```
* {
    margin: 0;
    padding: 0;
    border: 0;
    font-size: 100%;
    font: inherit;
    vertical-align: baseline;
}
```

Na tag **<head>**, vamos adicionar um *link* externo. Ele permitirá a configuração com o CSS para que, em seguida, possamos utilizar os ícones no nosso menu de navegação.



```
<link rel="stylesheet" href="https://pro.fontawesome.com/releases/v5.10.0/css/all.css" integrity="sha384-AYmEC3Yw5cVb3ZcuHtOA93w35dYTsvhLPVnYs9eStHfGJv0vKxVfELGroGkvsg+p" crossorigin="anonymous"/>
```

Dessa maneira, a estrutura dentro do **<body>** será a seguinte:



```
<body>
  <nav>
    <button class="meu-botao" id="meu-botao">
      <i class="fas fa-bars"></i>
    </button>
    <ul id="minha-nav">
      <li><a href="#">Inicio</a></li>
      <li><a href="#">Trabalho</a></li>
      <li><a href="#">Sobre</a></li>
      <li><a href="#">Contato</a></li>
    </ul>
  </nav>
</body>
```

Agora, precisamos **estilizar** o nosso botão e a lista da nossa Navbar no CSS, como se vê no exemplo:

```
/*Estilos Botão*/
.meu-botao {
    background-color: transparent;
    border: 10px;
    color: black;
    cursor: pointer;
    font-size: 30px;
    margin-left: 32px;
    float: left;
    line-height: 70px;
    display: none;
}

.meu-botao:focus {
    outline: none;
}
```

```
/*Estilo Navbar*/
nav {
    background-color: grey;
    flex-wrap: wrap;
}

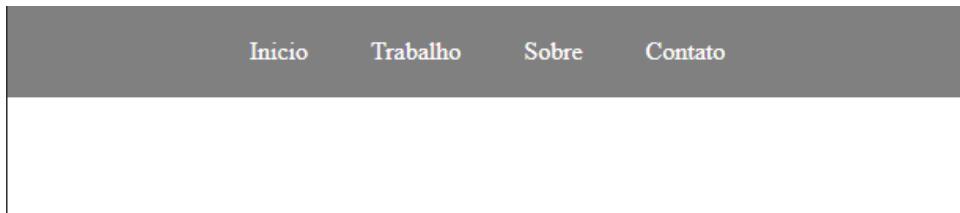
nav ul {
    display: flex;
    justify-content: center;
    list-style: none
}

nav ul li {
    padding: 20px 20px;
}

nav ul li a {
    color: white;
    text-decoration: none;
}

nav ul li:hover {
    text-decoration: underline;
    text-decoration-color: red;
}
```

Teremos, então, o resultado a seguir.



Que tal adaptarmos o *layout*? Vamos considerar que o usuário esteja usando um dispositivo móvel. Desejamos oferecer uma experiência mais amigável com a página e, para fazê-lo, vamos utilizar *media queries*.

Observe a imagem:

```
@media (max-width: 800px) {  
  nav ul {  
    display: none;  
    flex-direction: column;  
    width: 100%;  
  }  
  
  nav ul.show {  
    display: flex;  
  }  
  
  .meu-botao {  
    display: block;  
  }  
}  
  
@media (max-width: 600px) {  
  nav ul {  
    display: none;  
    flex-direction: column;  
    width: 100%;  
  }  
  
  nav ul.show {  
    display: flex;  
  }  
  
  .meu-botao {  
    display: block;  
  }  
  nav {  
    background-color: rgb(62, 109, 149);  
  }  
}
```

Do lado esquerdo, utilizamos a funcionalidade “**max-width**” (largura máxima) para configurar uma ação que ocorre quando o dispositivo possui até *800 pixels* de largura. Nesse caso, o elemento da lista desaparece e é substituído pelo botão da Navbar.

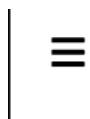
Já do lado esquerdo, usamos, novamente, “**max-width**” para configurar uma nova ação: quando a largura máxima for de até *600 pixels*, a cor de fundo será modificada.

Assim, teremos os resultados mostrados adiante.

Até *600 pixels*:



Abaixo de *600 pixels*:



Para adicionar funcionalidade à Navbar quando o dispositivo for menor que 600 pixels, podemos adicionar JavaScript ao nosso sistema.

```
const button = document.getElementById("meu-botao");
const nav = document.getElementById("minha-nav");

button.addEventListener('click', () => {
  nav.classList.toggle('show');
});
```

O código acima seleciona os elementos pelo atributo **id** presente no HTML, alterando a classe do botão e da Navbar quando ocorre um clique. Observe a seguir:



E aí, conseguiu compreender tudo até aqui? Com o conhecimento obtido nesses tópicos, você poderá desenvolver a página *web* que sempre quis!

É o momento de aplicar tudo o que aprendemos e testar os nossos conhecimentos na prática. Para isso, tente solucionar o desafio prático a seguir. Ele trata do uso de *media queries* em CSS, visando garantir um *layout* simples e responsivo.



DESAFIO PRÁTICO

Uso de *media queries* em CSS: aplicação prática.



Descrição

Utilize o projeto existente para implementar um conteúdo responsivo, mas use o mesmo *layout* que acabamos de criar. Pesquise como criar uma caixa de conteúdo usando as tags HTML `<div>`, `<main>` e `<h1>`. Pode ser um conteúdo com nome de usuário, espaçamento interno e cor de fundo.



Objetivos

- Explorar os usos da ferramenta e capacidade de customização;
 - Desenvolver habilidades práticas na codificação de CSS e criação de estruturas HTML.
-

Orientações

- Websites que podem ser úteis como referência:
www.w3schools.com | developer.mozilla.org;
 - Realize uma pesquisa prática e desenvolva uma pequena caixa no meio da tela. Ela deve se adaptar conforme redimensionada.
-

2.4 Conhecendo e utilizando ferramentas de depuração para testes em diferentes tamanhos de tela

Você acessa um *site* com o objetivo de ver um filme pelo qual você está ansioso há tempos. Então, você prepara a pipoca e senta no sofá, mas, ao abrir o *site*, percebe que o *player* de vídeo não está carregando ou reproduzindo o filme corretamente.



Disponível em: <https://encurtador.com.br/abzJZ>. Acesso em: 19 fev. 2024.

Embora frustrante, esse tipo de situação pode ocorrer com todas as pessoas. Durante o desenvolvimento de uma página *web*, é possível que as cores e fontes estejam incorretas. É possível, ainda, que o usuário enfrente problemas ao interagir com elementos da página, como botões que não funcionam conforme o esperado.

É para isso que existe a **depuração para testes em diferentes tamanhos de tela!** Com eles, é possível identificar e corrigir problemas ou erros que afetam a aparência ou funcionalidade de um *site* ou aplicativo. Isso considera diferentes dispositivos com tamanhos de tela variados, oferecendo uma experiência de usuário de alta qualidade em dispositivos móveis, *tablets*, *laptops* e *desktops*. Para realizar essa tarefa, é fundamental utilizar ferramentas de depuração que permitam identificar e corrigir problemas específicos relacionados ao *layout* e ao *design* responsivo. A seguir, vamos conhecer essas ferramentas e utilizar uma delas em um pequeno exemplo prático. Preparado?

DevTools

É a ferramenta padrão de testes em diferentes dispositivos. Integrada aos navegadores, possui uma ótima assertividade quando se trata de testes desse tipo.

Vamos utilizá-la neste capítulo.

Testes manuais	Os testes são conduzidos manualmente por um ser humano em vez de serem automatizados por meio de <i>scripts</i> ou ferramentas de teste automatizadas. Nesse processo, os testadores realizam interações com o <i>software</i> , explorando diferentes funcionalidades, inserindo dados, verificando resultados e identificando problemas ou <i>bugs</i> .
Emuladores	São dispositivos que possuem configurações padrão de tela e sistema semelhantes aos dispositivos reais, "emulando" (copiando) a sua interface. Há diversas opções para <i>download</i> , como BlueStacks , MEmu Play e Android Studio Emulator .
W3C Validator	Usar validadores de HTML e CSS, como o W3C Validator, pode ajudar a identificar problemas de código que podem afetar a renderização em diferentes tamanhos de tela. Garantir que o código esteja em conformidade com os padrões pode evitar comportamentos inesperados.

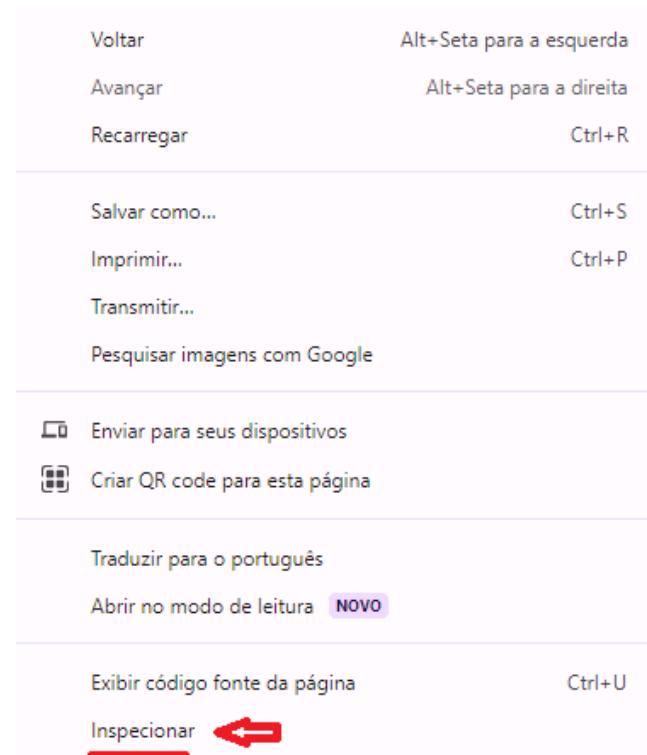
2.4.1 Operando e entendendo o DevTools

Os navegadores oferecem ferramentas de depuração e comparações de tamanhos dos elementos da tela que permitem examinar como o código-fonte se comportaria em vários dispositivos. A melhor parte é que isso ocorre sem que seja preciso utilizar diferentes dispositivos físicos.

Vamos explorar uma das principais ferramentas para depuração e testes em diferentes dispositivos para nossos sistemas, o **DevTools**. Como *designer* ou desenvolvedor *front-end*, o uso dessa ferramenta será parte essencial de sua rotina. Ela te ajudará a entender e analisar o comportamento de sua aplicação em tempo de execução. Agora, vamos partir para a prática!

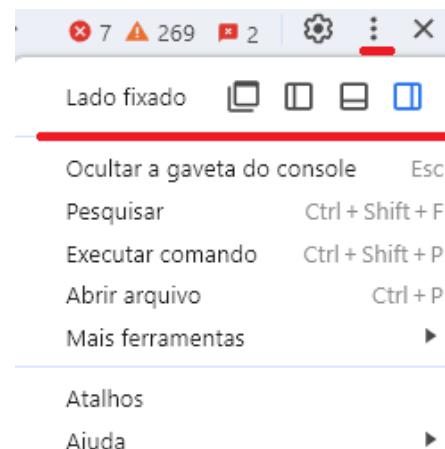
- Para essa atividade, utilizaremos o navegador **Chrome**, mas outros navegadores, como **Opera GX**, **Opera**, **Firefox**, **Edge**, **Safari** e **Brave**, também possuem essa funcionalidade.

Para fazer o uso do DevTools (ferramenta de desenvolvimento), você pode apertar a **tecla F12** ou, com o botão direito, clicar na tela do navegador e ir até a opção “**inspecionar**”.

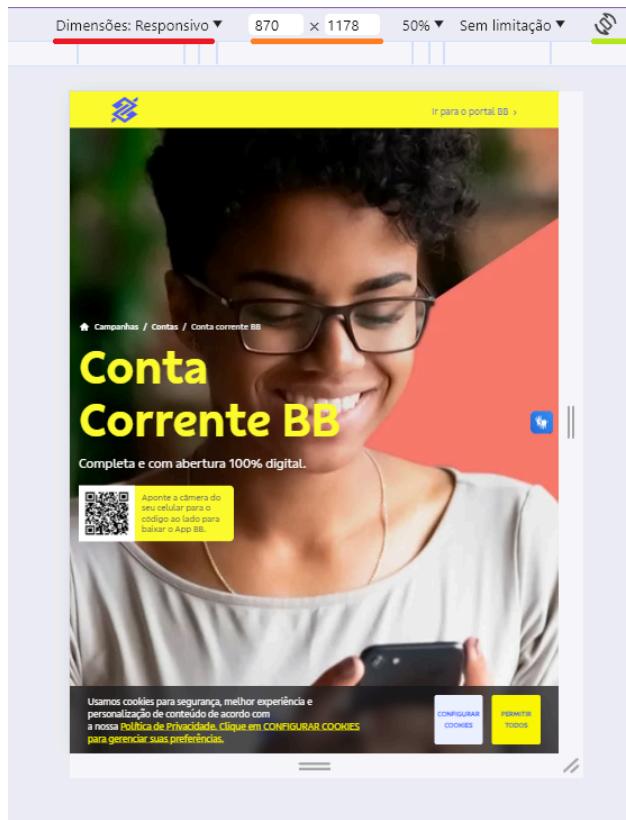


- Como exemplo, utilizaremos o *website* do **Banco do Brasil**. Aqui, buscaremos verificar como a interface do usuário se adapta a diferentes dispositivos.

Observe que uma janela com várias opções se abriu. Você pode personalizar o lado da tela em que ela é redimensionada clicando nos três pontos no canto superior direito e escolhendo entre as opções disponíveis.



3. Acima da tela, há as dimensões, resolução e opções para reorientar a tela, o que permite analisar como ela ficaria caso o dispositivo permita o modo paisagem.

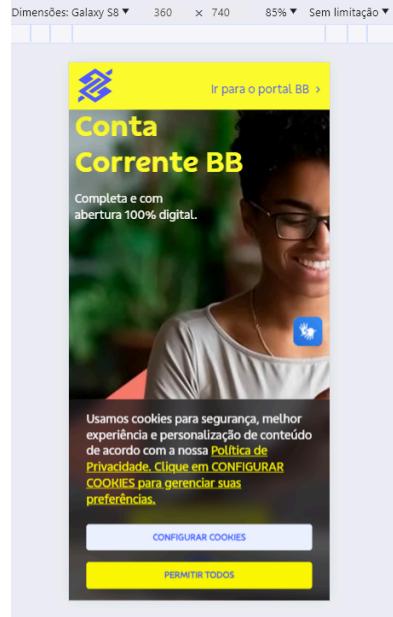


Disponível em: <https://www.bb.com.br/site/>

4. Para testar como a página ficaria em diferentes dispositivos, clique na opção de "dimensões" e, em seguida, selecione "diferentes dispositivos". Aqui, há acesso ao redimensionamento para diversos tipos de tela, como computadores pessoais, notebooks, celulares e tablets.



5. Escolhendo a opção “**Galaxy S8**”, é possível obter o resultado a seguir.



Disponível em: <https://www.bb.com.br/site/>

6. Também é possível inserir resoluções totalmente personalizadas, escolhendo a opção “responsivo” e, nos campos disponíveis, digitando a altura e largura desejadas. Observe:

Pronto! Agora que você aprendeu uma maneira fácil e eficaz de verificar a responsividade em diferentes dispositivos, que tal explorar um pouco mais a funcionalidade?

No desafio a seguir, você terá a oportunidade de realizar testes de responsividade para aprimorar as suas habilidades. Vamos lá?



DESAFIO PRÁTICO

Realizar testes de responsividade em sistemas amplamente utilizados



Descrição

Agora é a sua vez! Que tal praticar o tema que acabamos de discutir? Para isso, realize testes de usabilidade e responsividade nos sistemas que você mais usa no dia a dia.

Escolha um ou mais sistemas para realizar análises de responsividade e identificar pontos de melhoria. Depois, compartilhe as suas descobertas com outras pessoas, enriquecendo ainda mais o aprendizado em conjunto.



Objetivos

- Explorar os usos da ferramenta DevTools;
- Analisar os diferentes cenários e casos de uso;

- Entender, analisar e comunicar análises técnicas.
-

Orientações

- Realize uma análise prática e detalhada do produto escolhido;
 - Anote as considerações e compartilhe-as, posteriormente, com outras pessoas.
-

2.5 Otimizando experiências: testes de usabilidade, métricas UX e feedback do usuário na interação de design

Agora que exploramos um pouco da rotina e do dia a dia envolvidos no trabalho de um profissional de UI/UX, é fundamental entender como **coletar** e **medir informações** relevantes para impulsionar o ciclo de produção de um produto digital.

Esse especialistas são os responsáveis pela arquitetura por trás de produtos de sucesso. A comunicação eficaz, por sua vez, aliada à troca de opiniões, é essencial para um *design* bem-sucedido.

Assista a dois vídeos para aprofundar o seu entendimento sobre os temas abordados.



Vídeo no YouTube *Como é feito o teste de usabilidade?*, do canal Attri (3 ago. 2022).

Disponível em: <https://abre.ai/iOYx>. Acesso em: 6 fev. 2024.



Vídeo no YouTube *A importância de usar métricas em UX Design*, do canal Ux Unicórnio (22 jan. 2020).

Disponível em: <https://abre.ai/IOYM>. Acesso em: 6 fev. 2024.

A partir do que você observou nos vídeos anteriores, responda:

- 1.** O que um teste de usabilidade garante?
- 2.** O que é um teste assistido?
- 3.** Por que devemos sempre ser atenciosos em relação aos problemas dos usuários?
- 4.** É necessário ser um profissional com muita experiência para criar um *design* de qualidade?

Parabéns por chegar até aqui! É hora de refletir sobre o que você aprendeu e aplicar esse conhecimento em desafios futuros. Continue praticando e explorando novas áreas da programação para expandir ainda mais as suas habilidades.



Aprimoramento de experiência do usuário por meio de testes de usabilidade e análise de métricas de UX.



Descrição

Enfrentando crescente concorrência no mercado, uma renomada plataforma de *streaming* de música percebeu a necessidade de aprimorar significativamente a sua interface de usuário. Usuários relataram dificuldades na navegação para encontrar *playlists* específicas e descobrir novas músicas. O desafio é melhorar a interface para garantir maior satisfação e retenção de quem usa a plataforma.



Objetivos

- Aumentar a usabilidade da plataforma;
- Retenção dos usuários: que tipo de abordagem utilizar?;
- Melhorar as descobertas musicais: como oferecer uma descoberta musical eficaz para os usuários? Quais estratégias utilizar?

Orientações

- Realizar uma análise prática dos problemas existentes e propor soluções;
- Registrar os resultados, bem como descobertas, e elaborar um plano de ação..



Neste capítulo, abordamos a **comunicação eficaz** do *design* entre desenvolvedores e partes interessadas, destacando a importância de transmitir, **de forma clara**, as intenções do que é feito. Exploramos, ainda, os princípios do *design* responsivo, incluindo a adaptação para dispositivos móveis com o uso de *media queries* em CSS. Discutimos, também, a realização de testes em diferentes dispositivos móveis, a análise de métricas de UX e a importância da coleta de *feedback* dos usuários. Além disso, destacamos a relevância da iteração de *design* com base no ***feedback*** recebido, garantindo melhorias contínuas ao longo do processo de desenvolvimento e promovendo uma abordagem **ágil** e **adaptativa** para o alcance de resultados otimizados.

A seguir, veja como todos esses conceitos se relacionam.





ATIVIDADE DE FIXAÇÃO

1. Em uma pequena organização, como você se comunicaria com um colega que não possui a mesma expertise técnica que a sua, sendo o único profissional de TI?
2. O que é HTML e como ele se relaciona com a construção de sistemas?
3. Como utilizar o Visual Studio Code? Por que ele é uma ferramenta importante?
4. Onde escrever o código HTML para que ele apareça corretamente na tela?
5. Como se referir a um elemento HTML no CSS?
6. Já ouviu falar no *reset* de CSS? Como ele pode ajudar no desenvolvimento?
7. Na sua opinião, é necessário comprar vários dispositivos para testar os protótipos ou há alternativas mais eficientes?
8. Em termos de qualidade de produto, o que o teste de usabilidade garante?
9. Como você mediria os resultados de pesquisas de satisfação? Quais indicadores são os mais relevantes, em sua opinião?
10. Imagine-se como o encarregado de investigar a razão de um produto da empresa estar recebendo tantas reclamações acerca da navegação em dispositivos móveis. Quais seriam as suas primeiras ações para tentar resolver o problema de forma rápida?

Referências

- ANDERSON**, Stephen. *Seductive interaction design: creating playful, fun, and effective user experiences*. New Riders, 2011.
- BRINGHURST**, Robert. *The Elements of typographic style*. Hartley & Marks Publishers, 2008.
- BUTTERICK**, Matthew. *Typography for lawyers*. Jones McClure Publishing, 2010.
- COOPER**, Allan et al. *About face: the essentials of interaction design*. Wiley, 2014.
- EYAL**, Nir. *Hooked: How to build habit-forming products*. Penguin, 2014.
- GARRETT**, Jesse J. *The elements of user experience: user-centered design for the web and beyond*. New Riders, 2002.
- GOTHELF**, Jeff; **SEIDEN**, Josh. *Lean UX: designing great products with agile teams*. O'Reilly Media, 2013.
- GRAHAM**, Lisa. *Basics of design: layout and typography for beginners*. Cengage Learning, 2001.
- KRUG**, Steve. *Don't make me think: a common sense approach to web usability*. 3. ed. New Riders Publishing, 2014.
- KRUG**, Steve. *Rocket surgery made easy: the do-it-yourself guide to finding and fixing usability problems*. New Riders, 2009.
- NORMAN**, Don. A. *The design of everyday things*. Basic Books, 2013.
- RIES**, Eric. *The lean startup: how today's entrepreneurs use continuous innovation to create radically successful businesses*. Crown Business, 2011.
- TIDWELL**, Jeniffer; BREWER, Charles; VALENCIA, Aynne. *Designing interfaces: patterns for effective interaction design*. O'Reilly Media, 2010.
- TONDREAU**, Beth. *Layout essentials: 100 design principles for using grids*. Rockport Publishers, 2016.
- WILLIAMS**, Robin. *The non-designer's design book*. Peachpit Press, 2005.