



DESIGN DE INTERAÇÃO E UI/UX

AULA 4



Prof.^a Margarete Klamas

CONVERSA INICIAL

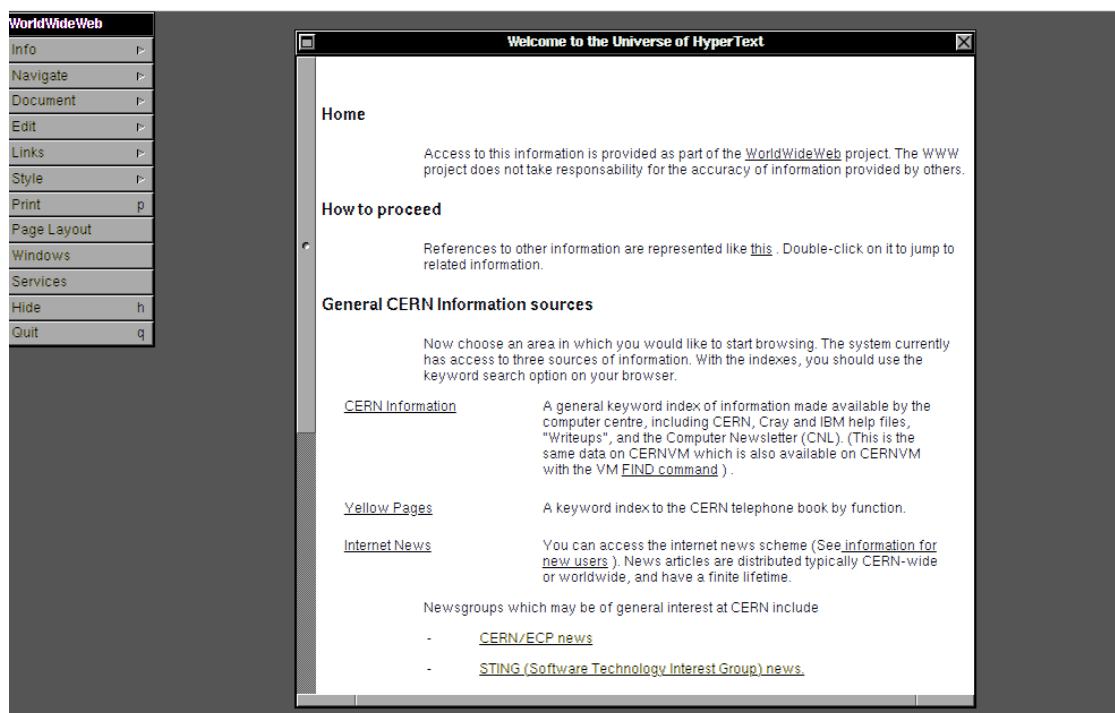
Nesta aula, inicialmente, vamos abordar o desenvolvimento do layout, desde o primeiro a ter sido desenvolvido. Falaremos também sobre a questão do Mobile First, vamos ver as recomendações do Google a respeito de responsividade, as diferenças entre mobile e desktop, e as fases de desenvolvimento da ideia, com base no que vimos ao estudarmos o design sprint, abordando esboço e wireframe.

Vamos dar início aos padrões de projeto para Android, com as recomendações do Google, e finalizaremos com um exemplo dos elementos essenciais de um aplicativo e algumas dicas de ferramentas.

TEMA 1 – COMO CHEGAMOS ATÉ AQUI: SOBRE COMO FAZER LAYOUTS

Tudo começou com Tim Berners-Lee, o inventor da web e atualmente diretor do W3C (World Wide Web Consortium). Na época, o navegador foi chamado de *World Wide Web*, posteriormente sendo nomeado *Nexus*. Na época, só suportava HTML, o que não era um problema, porque só eram compartilhados documentos científicos entre cientistas.

Figura 1 – Navegador Word Wide Web



Fonte: World Wide Web, [S.d.].



Posteriormente, a NCS, Centro Nacional de Aplicações para Supercomputadores da Universidade de Illinois, em 1992, desenvolveu o primeiro navegador gráfico para web, o Mosaic. Foi quando tudo começou. Esse navegador introduziu o primeiro elemento para apresentação de imagens.

Figura 2 – Navegador Mosaic (1992)



Fonte: NCSA, [S.d.].

Um acontecimento importante relacionado ao layout se deu em 1997. Trata-se da criação do elemento <table>, que permitia criar layouts complexos, diferente dos layouts em uma única coluna até então utilizados. As tabelas são maravilhosas para diagramar layouts, porém, infelizmente, não podemos utilizá-las para essa finalidade, em virtude de inúmeros inconvenientes:



- Dificuldade para fazer a manutenção dos códigos;
- Layouts inacessíveis a vários agentes de usuários, por exemplo, robôs de busca;
- Prejuízos ao SEO (Search Engine Optimization), o qual é o algoritmo para colocar as páginas no topo das buscas.

Uma fase importante ocorreu em 1996, com a criação do CSS1, lançado no Brasil em 2003. Pode-se estabelecer um marco aqui, com o HTML como camada de estruturação e o CSS como camada de estilos (apresentação).

O Responsive Web Design (RWD) entra em cena em 2010, com uma matéria escrita por Ethan Marcotte, que começa com a seguinte citação de John Allsopp:

O controle que os designers têm sobre conteúdos publicados na mídia impressa e tão desejado para aplicação na web é, simplesmente, consequência das restrições que a página impressa impõe. Nós devemos nos conscientizar de que a web não impõe as restrições da mídia impressa, ela necessita de design flexível. Devemos, acima de tudo, "aceitar o fluxo e o refluxo das coisas. (Allsopp, 2000 citado por Marcotte, 2010)

Resumindo, havia o layout fluído com tabelas, que migrou para layouts diagramados com DIVs, sendo que os primeiros layouts eram fixos: 800px x 600px. Alguns anos depois, a resolução dos monitores aumentou, passando a ser 1024 x 768. A solução na época foi trabalhar com porcentagens para tornar o layout fluído. A verdade é que dava muito trabalho e isso se agravou com as novas resoluções dos novos dispositivos. Era preciso projetar para este contexto. Nesse sentido, Bernard de Luna, em 2014, aconselha: priorize a leveza e flexibilidade na interface.

São elementos do RWD: grid fluído, imagens e mídias flexíveis e media queries.

TEMA 2 – MOBILE FIRST

Em 2009, surgiu o conceito de Mobile First. Luke Wroblewski defendeu a ideia de se iniciar um projeto com base no layout para apresentação em dispositivos móveis. As razões apontadas por Wroblewski para se pensar mobile first foram:



- Uma explosão da tecnologia mobile está em curso;
- O layout para dispositivos móveis implica em pensar no que é importante comunicar, com descarte de elementos desnecessários;
- Dispositivos móveis estão incorporando a cada dia mais e mais funcionalidades nativas e capacidade de criação de conteúdos ricos não existentes na maioria dos navegadores para desktop.

Aproveitamos para fazer um parêntese, retornando a um termo citado em aulas anteriores, o *web design adaptativo* (AWD – Adaptive Web Design), que foi criado por Aaron Gustafson. A ideia de AWD é de criar layouts que se adaptem às características e às capacidades do dispositivo do usuário. O termo não é sinônimo de RWD. A diferença está nas premissas do desenvolvimento. Segundo Aaron, AWD diz respeito à criação de interfaces que se adaptem às capacidades do usuário, seja na sua forma, seja nas suas funções.

É preciso lembrar que, apesar de, entre as recomendações das boas práticas do Google, estar a preferência ao RWD, isso obviamente não invalida o AWD.

Saiba mais



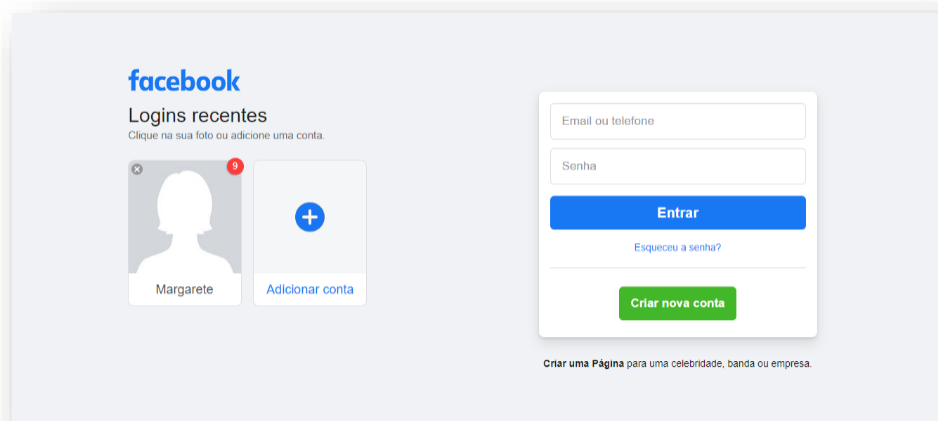
Web Design Responsivo - Google

Disponível em: <<https://developers.google.com/search/mobile-sites/mobile-seo/responsive-design>>. Acesso em: 8 nov. 2021

O Facebook, por exemplo, utiliza AWD:

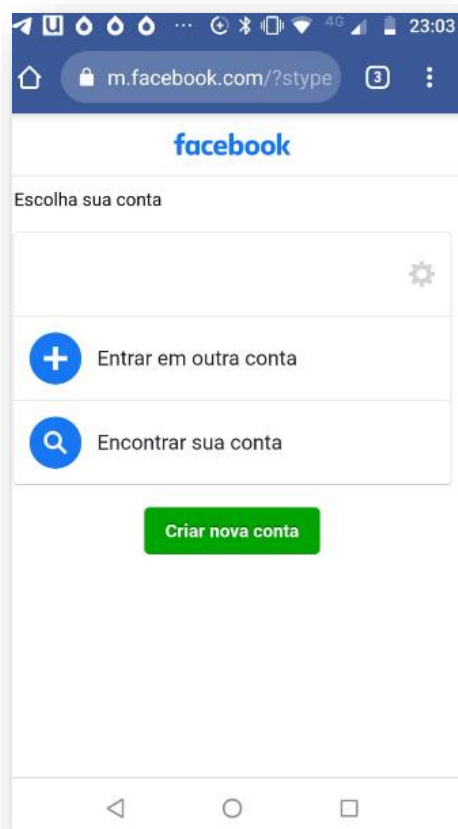


Figura 3 – Facebook para desktop



Fonte: Facebook, [S.d.].

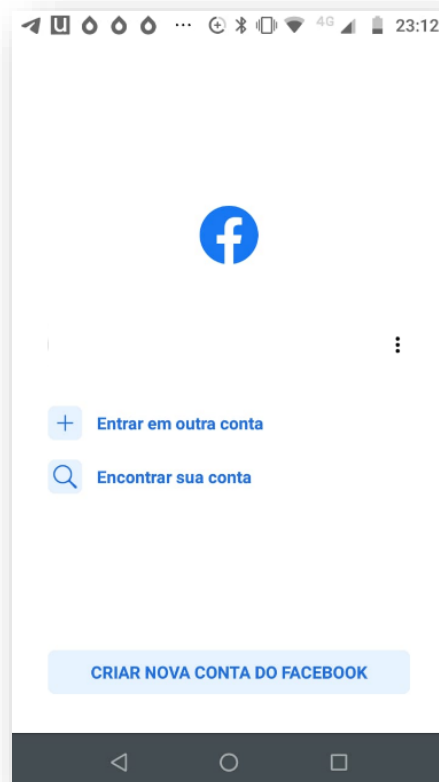
Figura 4 – Facebook utilizando o navegador do smartphone



Fonte: Facebook Mobile, [S.d.].



Figura 5 – Facebook – Aplicativo disponível na Play Store



Fonte: Aplicativo Facebook; Play Store, [S.d.].

2.1 Desktop versus mobile

Embora os objetivos das pessoas que acessam sites em desktops ou mobiles sejam os mesmos ou semelhantes, é necessário considerar que as experiências poderão ser diferentes.

Tabela 1 – Experiências em desktop e mobile

Desktop	Mobile
Tela grande	Tela pequena, em relação ao desktop
Teclado-padrão	Teclado pequeno
Mouse	Dedo, sem cursor
Banda Larga	3G, 4G, limite de banda, custo alto, muitos usuários com acesso pré-pago
Energia abundante	Energia limitada, principalmente se GPS estiver ativo



Rede consistente (cabo)	Rede inconsistente, em locais com sinal de internet 3G e 4G mais fraco
Maior espaço de armazenamento (HD e SSD)	Limite de espaço de armazenamento (memória interna e cartão de memória)
Maior poder de processamento	Menor poder de processamento
Uso em casa ou trabalho	Uso em qualquer hora, em qualquer lugar
Ambiente calmo, confortável, seguro e controlado	Em diferentes contextos, possivelmente realizando mais de uma tarefa.

Fonte: Samy, 2014.

Mobile First é a recomendação para se iniciar o projeto. Além de melhorar a experiência dos usuários nos dispositivos móveis, pode melhorar as versões para desktop.

TEMA 3 – FASES DO PROJETO

Já estudamos, em aulas anteriores, sobre a fase do design sprint. Sabemos que uma das primeiras fases é o **esboço**. Para fazer o esboço, você pode desenhar à mão, numa folha de papel. Pode também utilizar, para testes, a prototipagem em papel.

Figura 6 – Protótipo: papel



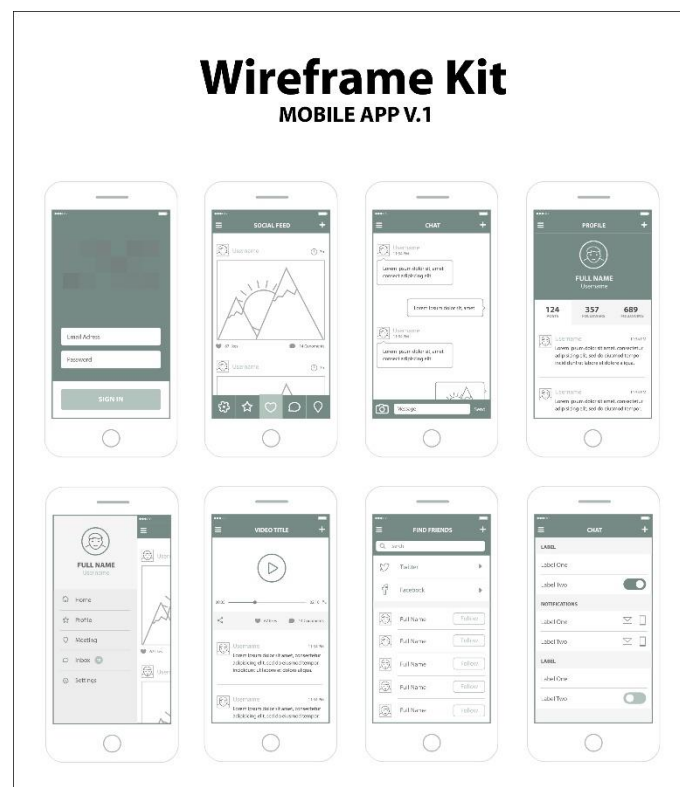
Crédito: REDPIXEL.PL/Shutterstock



Não pense que esboçar é perder tempo, pois, na verdade, é o contrário, já que você compreenderá com maior facilidade a interação. Quando o design estiver decidido, o desenvolvedor irá programar sabendo exatamente o que deve ser feito, sem perder tempo projetando um layout não funcional, que não agradará esteticamente, ou que não comunica o que é necessário comunicar.

Outra fase a ser desenvolvida é a fase do **wireframe**, que mostra a diagramação de cada tela para cada UI device sem detalhar, como na imagem a seguir, para visualizar o layout como um todo.

Figura 7 – Exemplo de Wireframe

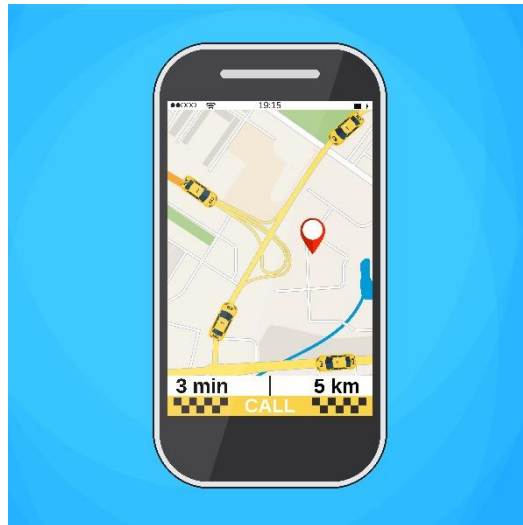


Crédito: AngelSID/Shutterstock

Na fase de projeto de layout mais elaborado, com detalhes, você criará os objetos que serão utilizados no desenvolvimento. Lembre-se de que cada uma das etapas requer aprovação dos responsáveis, ou stakeholders.



Figura 8 – Interface de aplicativo



Crédito: Modvector/Shutterstock

TEMA 4 – O AMBIENTE MÓVEL

Caro programador, já conversamos muito sobre UX e sobre compreender o usuário móvel. É necessário aplicar as heurísticas vistas. As recomendações do Google são: utilizar elementos e espaçamentos uniformes para manter a consistência entre dispositivos, ambientes e tamanhos de tela.

Os princípios do design móvel, segundo o Google, são:

- Previsível: use à vontade layouts intuitivos e previsíveis;
- Consistente: utilize grids;
- Responsivo: construa layouts adaptáveis.

4.1 Anatomia do design – Android

Para projetarmos para Android, precisamos conhecer as recomendações ergonômicas de quem criou a possibilidade de desenvolvermos para Android, que, no caso, é o próprio **Google**, com suas recomendações.

4.1.1 Regiões de layout

As regiões do layout são a base das experiências interativas. São containers que compõem o layout. Uma tela para desktop possui três containers principais:



1. Barra de aplicativos
2. Navegação
3. Área de conteúdo.

Figura 9 – Regiões de Layout



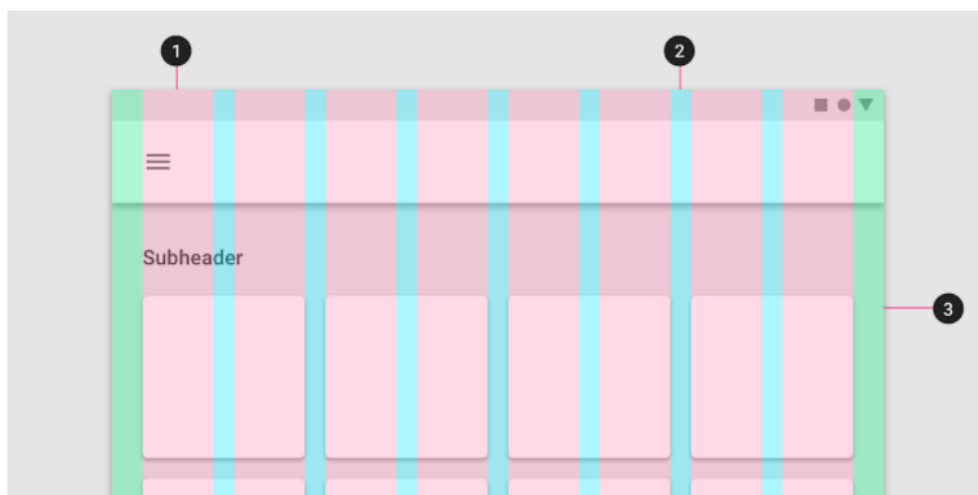
1. Barras de aplicativos 2. Navegação 3. Corpo

Fonte: Google, material.io, [S.d.].

4.1.2 Grade de layout responsiva

A grade do layout responsivo possui três elementos: colunas, calhas e margens.

Figura 10 – Grade de Layout



1. Colunas
2. Calhas
3. Margens

Fonte: Google, material.io, [S.d.].



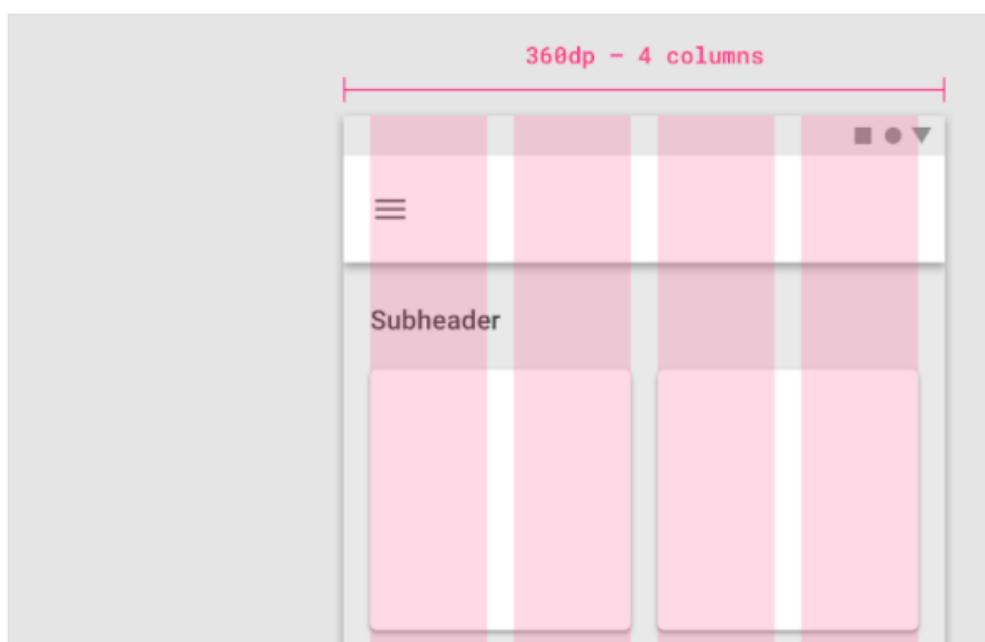
Vimos anteriormente que as medidas das colunas são estipuladas em porcentagens, o que facilita a responsividade. O número de colunas exibidas na grade é determinado pelo intervalo do ponto de interrupção, um intervalo de tamanhos de tela predeterminados. Um ponto de interrupção pode corresponder a um celular, tablet ou outro tipo de tela. Observe na imagem anterior que temos oito colunas, quantidade normalmente utilizada para se determinar o ponto de interrupção em tablets. Para smartphones, normalmente se projeta com grid de quatro colunas.

Tabela 2 – Breakpoint por dispositivo

Tamanho da tela	Margem	Corpo	GRID
Smartphone			
0-599 dp	16 dp		4
Tablet			
600-904 dp	32 dp		8
905-1239 dp	Dimensionamento	840 dp	12
Laptop			
1240-1439 dp	200 dp		12
Desktop			
1440+ dp	Dimensionamento	1040 dp	12

Fonte: Google, material.io, [S.d.].

Figura 11 – Ponto de interrupção para smartphones



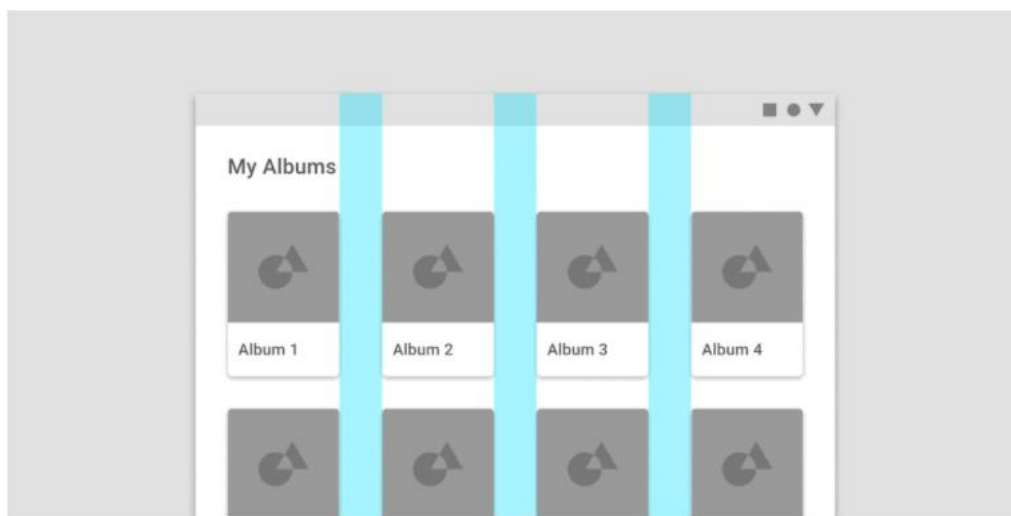
No celular, em um ponto de interrupção de 360 dp, esta grade de layout usa 4 colunas.

Fonte: Google, material.io, [S.d.].

As **calhas** são utilizadas para separar os conteúdos. A recomendação para celulares, em um ponto de interrupção (breakpoint) de 360 dp, é utilizar calhas de 16 dp. Para tablets com breakpoint de 600 dp, são utilizadas calhas de 24 dp. As margens para smartphone são de 16 dp.

A grade pode ser personalizada, com a utilização de calhas de 8 dp. No caso, sugere-se uma proximidade dos elementos (um dos princípios da Gestalt). Também pode-se utilizar calhas com espaçamento de 32dp.

Figura 12 – Grade de Layout



Esta grade de layout usa calhas maiores de 32 dp para criar mais separação entre as colunas. O espaço extra ajuda cada álbum a ser percebido como uma entidade individual dentro de uma coleção.

Fonte: Google, material.io, [S.d.].

Procure manter a harmonia das calhas, cujas medidas necessariamente devem ser menores que as das colunas. Você pode variar os valores das margens também, podendo utilizar, por exemplo, uma margem de 32 dp e calhas de 8 dp.

Observe que as medidas que apresentamos são informadas em dp, ou seja, densidade de pixels. As telas que possuem alta densidade possuem mais pixels por polegada em relação às telas de baixa densidade. Os elementos de UI de mesmas dimensões em pixels parecerão maiores em telas de baixa densidade e menores em telas de alta densidade (Google, 2021). Ao desenvolver para Android, use dp.

Para calcular densidade de pixels em relação à densidade de tela, utilize a equação a seguir:



$$dp = (\text{largura em pixels} \times 160) / \text{densidade de tela}$$

O google nos fornece uma tabela com as medidas:

Tabela 3 – Medidas de Layout

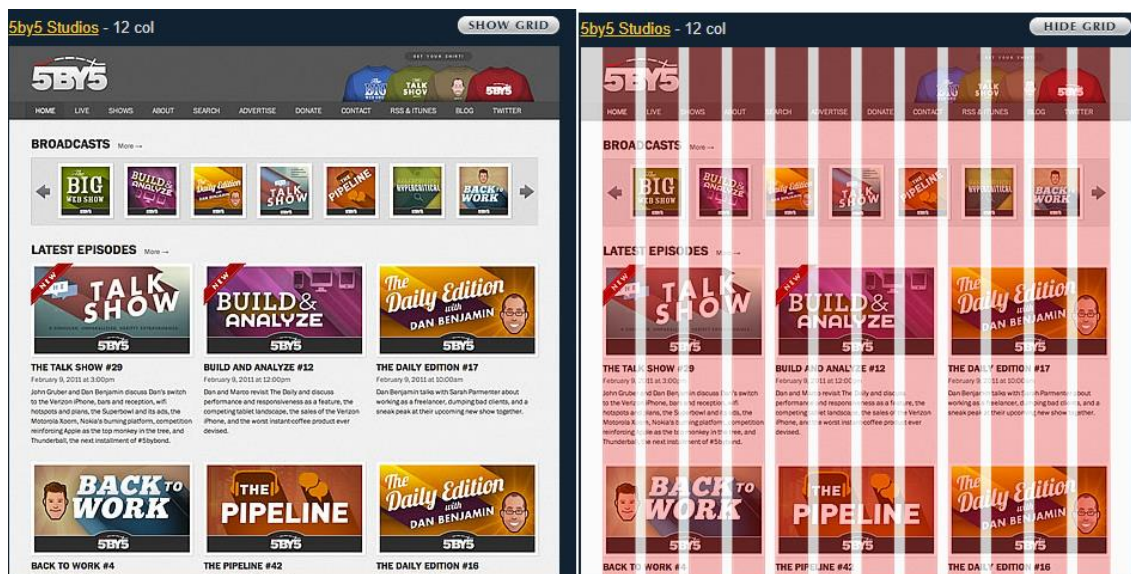
Largura física da tela	Densidade da tela	Largura da tela em pixels	Largura da tela em dps
1,5 pol	120	180 px	240 dp
1,5 pol	160	240 px	
1,5 pol	240	360 px	

Fonte: Google, material.io, [S.d.].

Como tamanho de fonte, são utilizados os pixels escaláveis (sp), que preservam as configurações de fontes do usuário. Para iOS, são usadas unidades em pontos (pts), e, para a web, o px (pixel).

4.2 Grid para WEB

Figura 13 – Exemplo de Grid - Desktop



Fonte: 960 Grid System, [S.d.].

A seguir, há um link para um Framework CSS desenvolvido por Nathan Smith. Você pode visualizar vários projetos em Grids de 12 ou 16 colunas e baixar os códigos também.



Por que estou mostrando Grid de até 16 colunas? Porque você pode precisar desenhar responsivamente, não exclusivamente mobile.

Saiba mais



Framework CSS

Disponível em: <<https://960.gs/>>. Acesso em: 10 out. 2021.

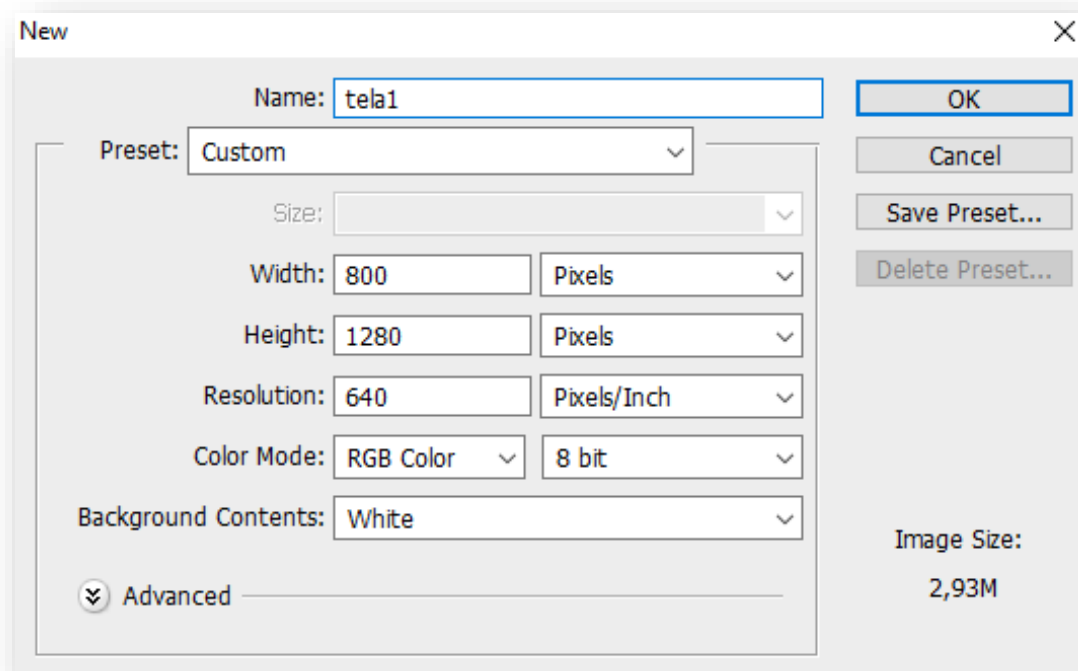
Ao preparar um layout responsivo, você utiliza uma coluna ao projetar para os dispositivos móveis. Lembre-se: Mobile First.

TEMA 5 – NA PRÁTICA

Digamos que você irá utilizar o Photoshop para desenvolver um wireframe e precisa trabalhar com as medidas recomendadas pelo Google para Android. Faremos um exemplo utilizando margem. Usaremos margem de 16 dp.

Vamos iniciar com a configuração da tela do Photoshop: **File >> New**.

Figura 14 – Photoshop > File > New



Fonte: Adobe Photoshop, [S.d.].



- Width (Largura): 800;
- Height (Altura): 1280;
- Unidade: pixel;
- Resolução: 640 pixels/inch (pixels/polegadas);
- Utilizaremos sempre a melhor resolução.

Nosso objetivo é criar uma margem de 16 dp no Photoshop. Para atingir esse objetivo, é necessário converter a unidade dp para px. Iremos utilizar uma calculadora on-line disponível em: <<https://pixplicity.com/dp-px-converter>>.

Figura 15 – Tela do pixplicity.com para converter dp para px

2. Mudar para dp 1. Mudar para xxxhdpi

16 dp no xxxhdpi compensando o tamanho da fonte do usuário em 100%

3. Digitar o valor 16

	dp	px	sp	milímetros	dentro	pt
ldpi	@ 16,00dp	= 12,00px	= 16,00 sp	= 2,54 mm	= 0,10 pol.	= 7,20pt
mdpi	@ 16,00dp	= 16,00px	= 16,00 sp	= 2,54 mm	= 0,10 pol.	= 7,20pt
tvdpi	@ 16,00dp	= 21,30 px	= 16,00 sp	= 2,54 mm	= 0,10 pol.	= 7,20pt
hdpi	@ 16,00dp	= 24,00px	= 16,00 sp	= 2,54 mm	= 0,10 pol.	= 7,20pt
xhdpi	@ 16,00dp	= 32,00px	= 16,00 sp	= 2,54 mm	= 0,10 pol.	= 7,20pt
xxhdpi	@ 16,00dp	= 48,00px	= 16,00 sp	= 2,54 mm	= 0,10 pol.	= 7,20pt
xxxhdpi	@ 16,00dp	= 64,00px	= 16,00 sp	= 2,54 mm	= 0,10 pol.	= 7,20pt

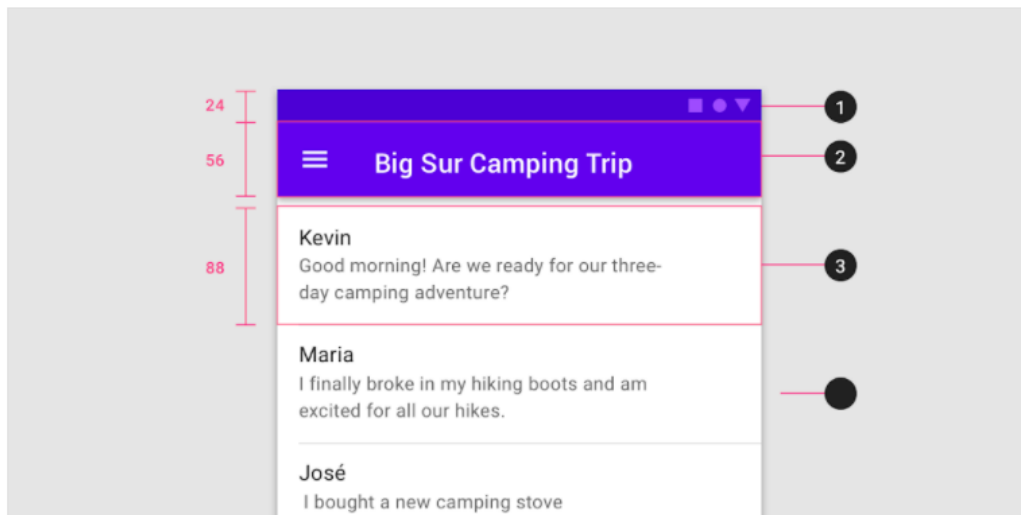
Fonte: Pixplicity, [S.d.].

Você pode utilizar esta calculadora on-line para fazer a conversão das medidas em dp para medidas em px.

Outro elemento importante é a altura da barra de status e a altura da barra do aplicativo.



Figura 16 – Altura das barras



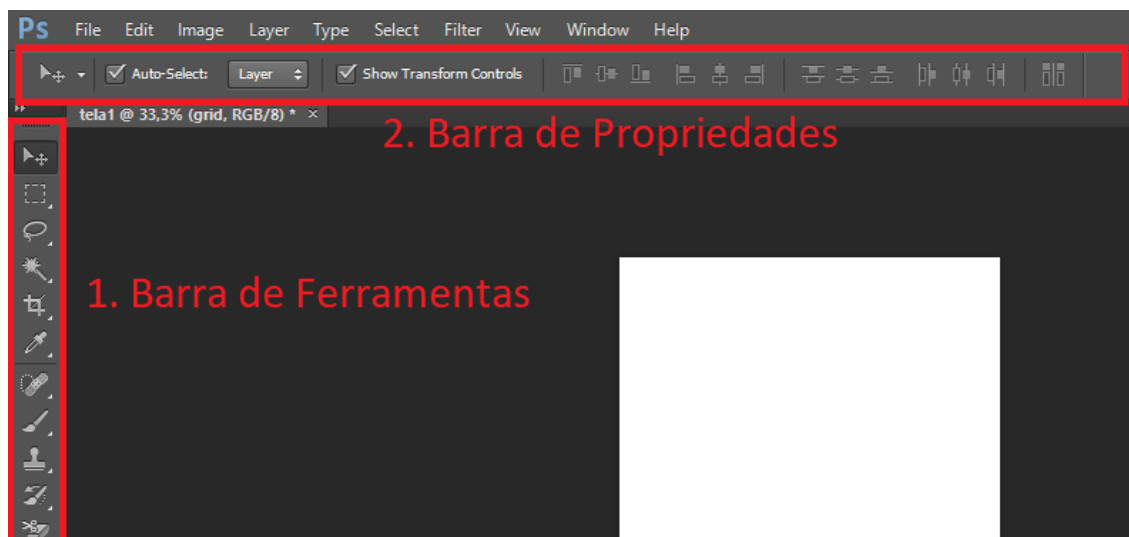
1. Altura da barra de status: 24 dp
2. Altura da barra de aplicativos: 56 dp
3. Altura do item da lista: 88 dp

Fonte: Google, material.io, [S.d.].

Vamos desenhar os elementos básicos da tela do dispositivo móvel no Photoshop.

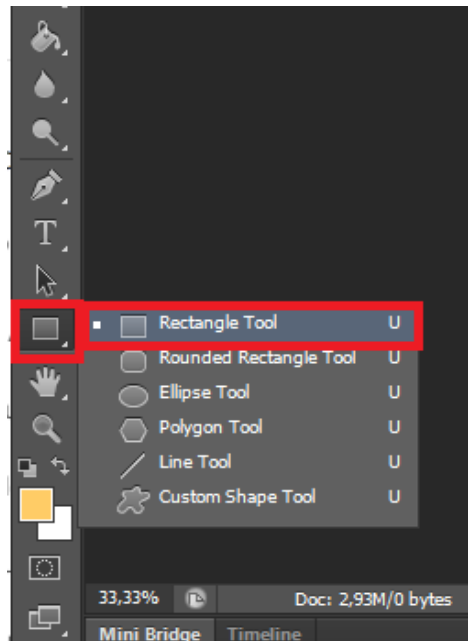
Na barra de ferramentas (está localizada na lateral), vamos selecionar a ferramenta retângulo:

Figura 17 – Barra de ferramentas



Fonte: Adobe Photoshop, [S.d.].

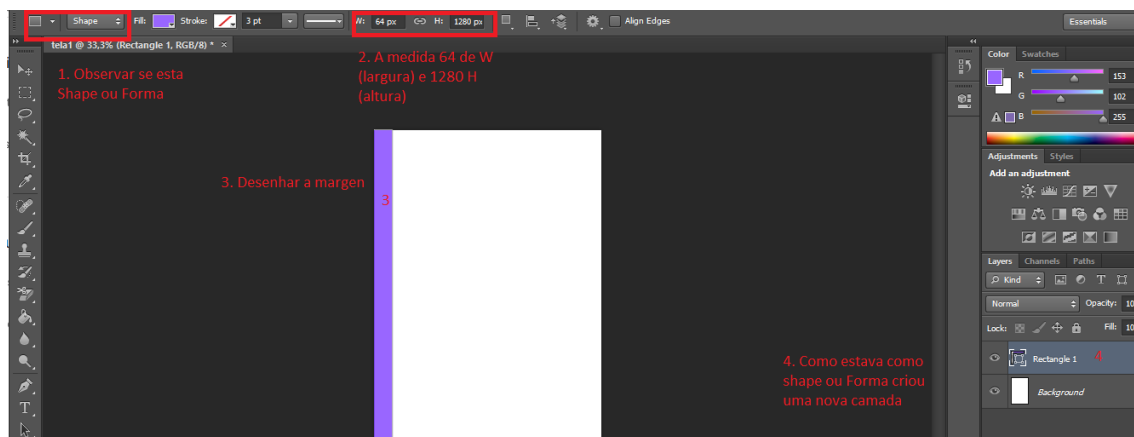
Figura 18 – Ferramenta retângulo



Fonte: Adobe Photoshop, [S.d.].

Vamos desenhar o retângulo que irá posicionar as margens à esquerda e à direita:

Figura 19 – Desenho do retângulo




Fonte: Adobe Photoshop.

É necessário:

1. Observar se está selecionado *shape* ou *forma*;
2. Desenhar o retângulo;
3. Converter a medida de 16 dp para 64 px (W: largura) e 1280 px (H: altura);

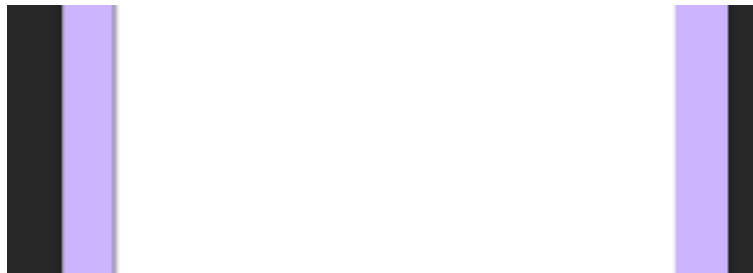


4. Como estava selecionado *Shape* (Forma), foi criada automaticamente uma segunda camada (layer).

Com a ferramenta de seleção , você pode pressionar a tecla CTRL+J e duplicar a camada (Layer) para marcar a margem direita.

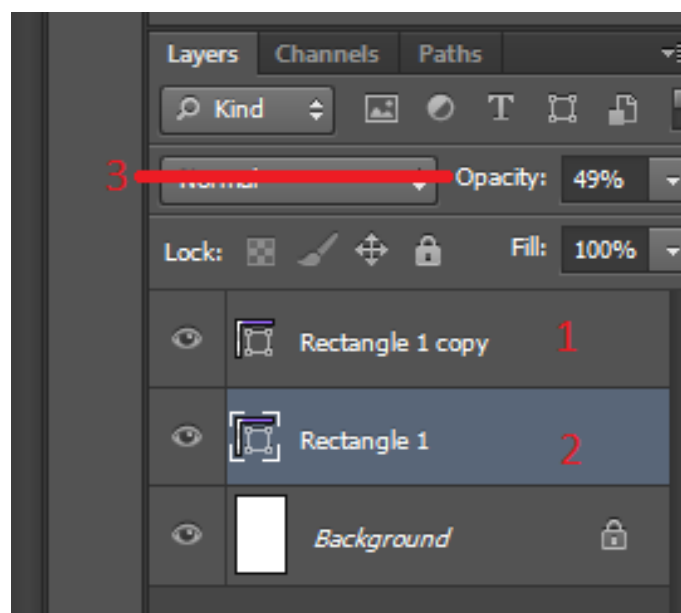
Com as duas margens desenhadas, podemos mudar a opacidade para em torno de 50%.

Figura 20 – Parte da imagem com as duas margens



Fonte: Adobe Photoshop.

Figura 21 – Painel Camadas (Layers)



1 e 2 – selecionando no nome das camadas, basta clicar sobre os nomes, mudar a Opacity para 49 ou 50%

Esse painel é o painel Layers (Camadas)

Fonte: Adobe Photoshop, [S.d.].

Na calculadora on-line de conversão de dp para pixel, convertermos todos os valores:

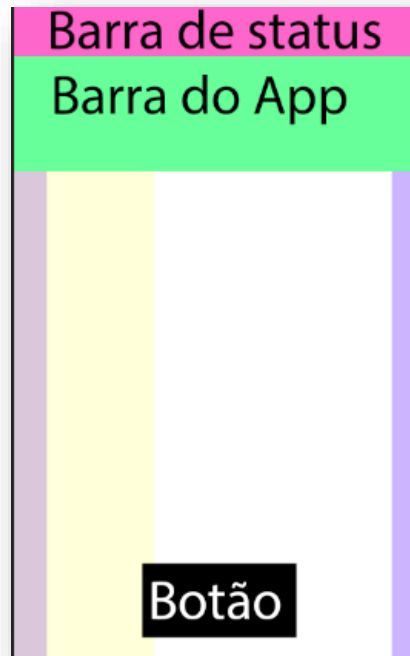


Tabela 4 – Valores de conversão

Valor em dp	Valor em pixel	Elemento
16dp	64px	Margens
24dp	96px	Barra de status
56dp	224px	Barra do aplicativo
88dp	352px	Altura da lista
36dp	144px	Altura de botão
64dp	256px	Largura mínima do botão

Você, então, deve recriar a base do aplicativo, conforme mostra a figura a seguir.

Figura 22 – Base do aplicativo

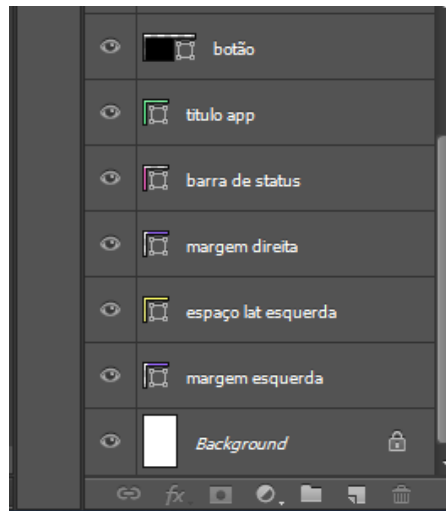


Fonte: Adobe Photoshop, [S.d.].



Você deve nomear as camadas (Layers) no Photoshop para selecionar e localizar o item com mais facilidade. Para isso, basta dar um clique duplo sobre o nome da camada e alterar para o nome desejado.

Figura 23 – Camadas (Layers)



Fonte: Adobe Photoshop, [S.d.].

5.1 Ferramentas úteis

A seguir, apresentaremos ferramentas que verificam medidas de telas de dispositivos.

1. My Device

Figura 24 – MyDevice



Fonte: MyDevice, [S.d.].



2. DPI Love

Figura 25 – DPI Love

PPI – RESOLUÇÃO DE UM DEVICE (NOTEBOOK)



PPI – RESOLUÇÃO DE UM SMARTPHONE



Fonte: DPILove, [S.d.].

FINALIZANDO

Nesta aula, você viu quantos detalhes há no design para dispositivos móveis. É necessário projetar cada tela do aplicativo, respeitando as recomendações dos padrões Android/iOS.

A fase do design é rica em informações e não deve ser ignorada. É muito comum pensar que basta sentar-se e programar. No entanto, ao publicar o projeto, os planos podem não sair conforme o imaginado e o retrabalho será mais desgastante.

Obviamente, em muitas empresas, há pessoas especializadas para cada função. Provavelmente, haverá responsáveis pela elaboração das telas da aplicação, para ser utilizada na prototipação interativa ou mesmo para realizar os protótipos diretamente nas ferramentas interativas.

É importante conhecer todas as fases do processo!



REFERÊNCIAS

960 GRIDE SYSTEM. [S.d.]. Disponível em: <<https://960.gs/>>. Acesso em: 8 nov. 2021.

ALLSOPP, J. A Dao of Web Design. **A List Apart**, 7 abr. 2000. Disponível em: <<https://alistapart.com/article/dao/>>. Acesso em: 11 out. 2021.

DPILOVE. [S.d.]. Disponível em: <<https://dpi.lv/>>. Acesso em: 8 nov. 2021

FACEBOOK. [S.d.]. Disponível em: <<https://www.facebook.com/>>. Acesso em: 8 nov. 2021.

FACEBOOK MOBILE. Disponível em: <<https://mbasic.facebook.com/>>. Acesso em: 8 nov. 2021.

GOOGLE. Material Design. Disponível em: <<https://material.io/>>. Acesso em: 11 out. 2021.

_____. **Web Design Responsível.** Disponível em: <<https://developers.google.com/search/mobile-sites/mobile-seo/responsive-design>>. Acesso em: 11 out. 2021.

MARCOTTE, E. Responsive Web Design. **A List Apart**, 25 mai. 2010. Disponível em: <<https://alistapart.com/article/responsive-web-design/>>. Acesso em: 11 out. 2021.

MEW, K. **Aprendendo Material Design:** Domine o Material Design e crie interfaces bonitas e animadas para aplicativos móveis e web. São Paulo: Novatec, 2016.

MYDEVICE. [S.d.]. Disponível em: <<https://www.mydevice.io/>>. Acesso em: 8 nov. 2021

NCSA. [S.d.]. Disponível em: <<http://www.ncsa.illinois.edu/enabling/mosaic>>. Acesso em: 8 nov. 2021.

PIXPLICITY, [S.d.]. Disponível em: <<https://pixplicity.com/dp-px-converter>>. Acesso em: 8 nov. 2021

SILVA, M. S. **Web Design Responsivo.** São Paulo: Novatec, 2014.

W3CSCHOOLS. **Grid.** Disponível em: <https://www.w3schools.com/css/css_grid.asp>. Acesso em: 11 out. 2021.



WORLD WID WEB. [S.d.] Disponível em:
<<https://worldwideweb.cern.ch/browser/>>. Acesso em: 8 nov. 2021.