Faculdade de Tecnologia de Sorocaba

Tecnologia em Análise e Desenvolvimento de Sistemas



**Programação para Web: Design Responsivo**

ATIVIDADE 02

Prof.º Denilce De Almeida

**Disciplina: Programação para Web**

**Antônio Miguel Borges Correa** 0030482111013

Sorocaba

Agosto/2022

# Introdução

Nos primórdios do web design, páginas eram criadas para serem visualizadas em um tamanho de tela específico. Se o usuário tivesse uma tela maior ou menor do que o esperado, os resultados iam de barras de rolagem indesejadas, tamanhos de linha excessivamente longos e uso inadequado do espaço. À medida que diferentes tamanhos de tela foram aparecendo, surgiu o conceito de web design responsivo (RWD), um conjunto de práticas que permite que páginas da Web alterem seu layout e aparência para se adequarem a diferentes larguras, resoluções, etc. É uma ideia que mudou a forma de como projetamos para a Web com múltiplos dispositivos

Com uma internet cada vez mais acessada a partir de dispositivos móveis, não é mais suficiente ter um design de website estático que só fica bem na tela de um computador.

Sem mencionar, você também tem que considerar comprimidos, laptops 2 em 1, e diferentes modelos de smartphone com diferentes dimensões de tela quando se trata de um projeto.

Portanto, colocar seu conteúdo em uma única coluna e chamá-lo de “desistir” não vai cortá-lo.

Com um web design ágil, você pode garantir que seu website tenha o melhor visual em celulares, tablets, laptops e telas de desktop.

E essa melhoria na experiência do usuário significa maiores conversões e crescimento dos negócios.

# História dos layouts de sites

Em algum ponto da história, você tinha duas opções ao criar um site:

Você pode criar um site líquido, que se estenderia para preencher a janela do navegador ou um site de largura fixa, que seria um tamanho fixo em pixels.

Essas duas abordagens, geralmente, resultavam em um site com a melhor aparência na tela da pessoa que o projetava! O site líquido resultou em um design esmagado para telas menores (como visto abaixo) e comprimentos de linha muito longos em telas maiores.

À medida que a Web para dispositivos móveis começava a se tornar realidade com os primeiros telefones com essas características, empresas que desejavam adotar os dispositivos móveis geralmente criavam uma versão mobile do seu site, com uma URL diferente (geralmente algo como m.exemplo.com ou exemplo.mobi). Isso significava que duas versões separadas do site tinham que ser desenvolvidas e mantidas atualizadas.

Além disso, esses sites para celular geralmente ofereciam uma experiência muito restrita. À medida que os dispositivos móveis se tornaram mais poderosos e capazes de exibir sites completos, os usuários de celular ficaram frustrados, pois, se viram presos na versão mobile do site e incapazes de acessar todas as informações que faziam parte da versão para desktop.

# 2.1. Layouts flexíveis antes do design responsivo

Várias abordagens foram desenvolvidas para tentar resolver as desvantagens dos métodos de largura líquida ou largura fixa da construção de sites. Em 2004, Cameron Adams escreveu um artigo intitulado Resolution dependent layout, descrevendo um método para criar um design que pudesse se adaptar a diferentes resoluções de tela. Essa abordagem necessitava do JavaScript para detectar a resolução da tela e carregar o CSS correto.

Zoe Mickley Gillenwater foi fundamental no seu trabalho de descrever e formalizar as diferentes maneiras pelas quais sites flexíveis poderiam ser criados, tentando encontrar um meio termo entre preencher a tela ou ter tamanho completamente fixo.

# 2.2. Design Responsivo

O termo design responsivo foi cunhado por Ethan Marcotte em 2010, e descreveu o uso de três técnicas combinadas.

A primeira foi a ideia de grids fluidas, que já estava sendo explorada por Gillenwater, e pode ser encontrada no artigo de Marcotte, Fluid Grids (publicado em 2009 em A List Apart).

A segunda técnica foi a ideia de imagens fluidas. Usando uma técnica muito simples que setava a propriedade max-width com 100%, as imagens seriam reduzidas se a coluna que as continha se tornasse mais estreita que o tamanho intrínseco da imagem, mas nunca aumentariam. Isso permitiu que uma imagem fosse reduzida em tamanho para caber em uma coluna de tamanho flexível, em vez de transbordar, mas não aumentava e nem tornava-se pixelizada se a coluna fosse mais larga que a imagem.

O terceiro componente-chave foi a media query. Media Queries habilitavam o tipo de opção de layout usando o JavaScript, que Cameron Adams havia explorado anteriormente, usando apenas CSS. Em vez de ter um layout para todos os tamanhos de tela, o layout podia ser alterado. As barras laterais podiam ser reposicionadas para a tela menor ou uma navegação alternativa podia ser exibida.

É importante entender que o design responsivo não é uma tecnologia separada — é um termo usado para descrever uma abordagem ao web design, ou um conjunto de melhores práticas, usado para criar um layout que possa responder ao dispositivo que está sendo usado para visualizar o conteúdo. Na exploração original de Marcotte, isso significava grades flexíveis (usando floats) e media queries, no entanto, nos últimos 10 anos, desde que o artigo foi escrito, trabalhar de forma responsiva se tornou um padrão. Os métodos de layout CSS modernos são inerentemente responsivos, e temos coisas novas incorporadas à plataforma web para facilitar o design de sites responsivos.

O restante deste artigo indicará os vários recursos da plataforma web que você pode usar ao criar um site responsivo.

# 2.3. Media Queries

O Design Responsivo apenas foi capaz de emergir devido o recurse de media query. A especificação Media Queries Level 3 se tornou uma Recomendação de Candidato em 2009, significando que estava pronto para ser implementado nos browsers. Media Queries nos permitem executar uma série de testes (e.g. se a tela do usuário é maior que uma certa largura, ou uma certa resolução) e aplicar um CSS seletivamente para estilizar a página de acordo com as necessidades do usuário.

Por exemplo, a seguinte media querie testa se a página atual está sendo exibida como mídia de tela (portanto, não é um documento impresso) e o viewport tem pelo menos 800 pixels de largura. O CSS para o seletor .container será aplicado apenas se essas duas condições forem verdade.

Texto

Descrição gerada automaticamente

Você pode adicionar múltiplos media queries dentro de uma folha de estilo, ajustando inteiramente seu layout ou partes dele que melhor se adequem a vários tamanhos de tela. Os pontos em quem uma Media Query é introduzida e o layout alterado são conhecidos como breakpoints.

Uma abordagem comum ao utilizar Media Queries é criar um layout de única coluna para dispositivos de telas pequenas (e.g smartphones), então fazer a checagem para telas maiores e implementar um layout de múltiplas colunas quando houver largura suficiente. Esse design é frequentemente descrito como mobile first.

# 2.4. Grids Flexíveis

Sites responsivos não apenas mudam seu layout entre breakpoints, eles são construídos em grids flexíveis. Um grid flexível significa que não há necessidade de marcar todos os tamanhos possíveis existentes, e sim, construir um layout perfeito baseado em pixels que se adequa automaticamente à tela. Essa abordagem seria impossível dado o vasto número de dispositivos com tamanhos diferentes que existem e o fato de que, mesmo nos desktops, as pessoas nem sempre utilizam a janela do navegador maximizada.

Com o uso de um grid flexível, não há necessidade de adicionar um breakpoint e alterar o desing no ponto onde o conteúdo começa a parecer ruim em determinada tela. Por exemplo, se o comprimento da linha se torna ilegível à medida que o tamanho da tela aumenta, ou uma caixa se fica espremida com duas palavras em cada linha, conforme o tamanho diminui.

Nos primórdios do design responsivo a única opção disponível para realizar layouts era utilizando floats. Layouts flexíveis flutuantes eram alcançados dando a cada elemento uma largura percentual, garantindo que em todo layout os totais não fossem maior que 100%.

Gráfico

Descrição gerada automaticamente com confiança média

Por exemplo, se o tamanho da coluna de destino for 60 pixels e o contexto (ou contêiner) em que ela estiver for 960 pixels, dividiremos 60 por 960 para obter um valor que possamos usar em nosso CSS, depois de mover o ponto decimal duas casas para o certo.

Texto

Descrição gerada automaticamente com confiança baixa

Esta abordagem será encontrada em muitos lugares da web hoje, e está documentada aqui na seção de layout do nosso artigo de métodos de layout Legacy. É provável que você se deparará com sites usando essa abordagem em seu trabalho, por isso vale a pena entendê-la, mesmo que você não construa um site moderno usando uma grade flexível baseada em flutuação.

# 2.5. Tecnologias modernas de layout

Métodos modernos de layout, como layout de várias colunas, Flexbox e Grid são responsivos por padrão. Todos eles assumem que você está tentando criar uma grade flexível e dar-lhe maneiras mais fáceis de fazê-lo.

# 2.6 Multicol

O mais antigo desses métodos de layout é o multicol — quando você especifica um, isso indica quantas colunas você deseja que seu conteúdo seja dividido. O navegador então trabalha o tamanho destes, um tamanho que mudará de acordo com o tamanho da tela.

Interface gráfica do usuário

Descrição gerada automaticamente

Se você especificar um column-width, estará especificando uma largura mínima. O navegador criará tantas colunas dessa largura quantas couberem confortavelmente no contêiner e, em seguida, dividirá o espaço restante entre todas as colunas. Portanto, o número de colunas mudará de acordo com a quantidade de espaço existente.

Logotipo, nome da empresa

Descrição gerada automaticamente

# 2.7. Flexbox

No Flexbox, os itens flex encolherão e distribuirão espaço entre os itens de acordo com o espaço em seu recipiente, como seu comportamento inicial. Alterando os valores e você pode indicar como você quer que os itens se comportem quando eles encontram mais ou menos espaço ao seu redor.

No exemplo abaixo os itens flex levarão cada um um espaço igual no recipiente flex, usando a taquigrafia de como descrito no tópico de layout Flexbox: Dimensionamento flexível de itens flex.

# Interface gráfica do usuário, Texto, Aplicativo Descrição gerada automaticamente com confiança média

# 2.8. Grid

No Layout da grade CSS, a unidade permite a distribuição do espaço disponível entre as faixas de grade. O exemplo seguinte cria um contêiner de grade com três faixas dimensionadas em . Isso criará três faixas de coluna, cada uma ocupando uma parte do espaço disponível no contêiner. Você pode saber mais sobre essa abordagem para criar uma grade no tópico Learn Layout Grids, em grades flexíveis com a unidade fr.

Interface gráfica do usuário

Descrição gerada automaticamente com confiança média

# 2.9. Responsive Image

A abordagem mais simples das imagens responsivas foi descrita nos primeiros artigos de Marcotte sobre design responsivo. Basicamente, você pegaria uma imagem que estava no maior tamanho que poderia ser necessária, e a reduziria. Esta ainda é uma abordagem usada hoje em dia, e na maioria das folhas de estilo você encontrará o seguinte CSS em algum lugar:

Logotipo, nome da empresa

Descrição gerada automaticamente

Há desvantagens óbvias nessa abordagem. A imagem pode ser exibida muito menor do que seu tamanho intrínseco, que é um desperdício de largura de banda — um usuário de celular pode estar baixando uma imagem várias vezes do tamanho do que eles realmente vêem na janela do navegador. Além disso, você pode não querer a mesma proporção de imagem no celular como no desktop. Por exemplo, pode ser bom ter uma imagem quadrada para celular, mas mostrar a mesma cena que uma imagem de paisagem no desktop. Ou, reconhecendo o tamanho menor de uma imagem no celular, você pode querer mostrar uma imagem completamente diferente, que é mais facilmente compreendida em um tamanho de tela pequeno. Essas coisas não podem ser alcançadas simplesmente escalando uma imagem.

Imagens responsivas, usando o elemento e os atributos resolvem ambos os problemas. Você pode fornecer vários tamanhos juntamente com "dicas" (meta dados que descrevem o tamanho da tela e resolução para o qual a imagem é mais adequada), e o navegador escolherá a imagem mais apropriada para cada dispositivo, garantindo que um usuário baixe um tamanho de imagem apropriado para o dispositivo que está usando.

Você também pode arter imagens diretas usadas em diferentes tamanhos, fornecendo assim uma cultura diferente ou imagem completamente diferente para diferentes tamanhos de tela.

# 2.10. Tipografia Responsiva

Um elemento de design responsivo não abordado em trabalhos anteriores foi a ideia de tipografia responsiva. Essencialmente, isso descreve a alteração dos tamanhos das fontes dentro das consultas de mídia para refletir quantidades menores ou maiores de imóveis de tela.

Neste exemplo, queremos definir nossa posição de nível 1, o que significa que será quatro vezes o tamanho da nossa fonte base. É uma posição muito grande! Só queremos esse título jumbo em tamanhos de tela maiores, por isso primeiro criamos um título menor e depois usamos consultas de mídia para substituí-lo com o tamanho maior se soubermos que o usuário tem um tamanho de tela de pelo menos 1200px.

Texto

Descrição gerada automaticamente

Editamos nosso exemplo de grade responsiva acima para incluir também o tipo responsivo usando o método descrito. Você pode ver como o título muda de tamanho à medida que o layout vai para a versão de duas colunas.

# 2.11. Utilização de unidades de viewport para tipografia responsiva

Uma abordagem interessante é usar a unidade de viewport “vw” para habilitar a tipografia responsiva. 1vwé igual a um por cento da largura da janela de visualização, o que significa que se você definir o tamanho da fonte usando vw, ele sempre estará relacionado ao tamanho da janela de visualização.

Uma imagem contendo Logotipo

Descrição gerada automaticamente

O problema de fazer o acima é que o usuário perde a capacidade de ampliar qualquer texto definido usando a unidade vw, pois esse texto está sempre relacionado ao tamanho da janela de visualização. Portanto, você nunca deve definir o texto usando apenas unidades de viewport

Existe uma solução e envolve o uso do calc(). Se você adicionar a vwunidade a um valor definido usando um tamanho fixo, como ems ou rems, o texto ainda poderá ser ampliado. Essencialmente, a vwunidade adiciona em cima desse valor ampliado:

Logotipo

Descrição gerada automaticamente com confiança baixa

Isso significa que só precisamos especificar o tamanho da fonte do título uma vez, em vez de configurá-lo para dispositivos móveis e redefini-lo nas consultas de mídia. A fonte aumenta gradualmente à medida que você aumenta o tamanho da janela de visualização.

# 2.12. A metatag da janela de visualização

Se você observar a fonte HTML de uma página responsiva, normalmente verá a seguinte <meta> tag no <head>documento.



Essa metatag informa aos navegadores móveis que eles devem definir a largura da janela de visualização para a largura do dispositivo e dimensionar o documento para 100% do tamanho pretendido, o que mostra o documento no tamanho otimizado para dispositivos móveis desejado.

Por que isso é necessário? Porque os navegadores móveis tendem a mentir sobre a largura da janela de visualização.

Essa metatag existe porque quando o iPhone original foi lançado e as pessoas começaram a visualizar sites em uma pequena tela de telefone, a maioria dos sites não era otimizada para dispositivos móveis. O navegador móvel, portanto, definiria a largura da janela de visualização para 960 pixels, renderizaria a página nessa largura e mostraria o resultado como uma versão reduzida do layout da área de trabalho. Outros navegadores móveis (por exemplo, no Google Android) fizeram a mesma coisa. Os usuários podiam dar zoom e deslocar o site para ver os bits em que estavam interessados, mas parecia ruim. Você ainda verá isso hoje se tiver a infelicidade de encontrar um site que não tenha um design responsivo.

O problema é que seu design responsivo com pontos de interrupção e consultas de mídia não funcionará como pretendido em navegadores móveis. Se você tiver um layout de tela estreito com largura de janela de visualização de 480px ou menos, e a janela de visualização estiver definida em 960px, você nunca verá seu layout de tela estreita em dispositivos móveis. Ao configurar, width=device-widthvocê está substituindo o padrão da Apple pela width=960pxlargura real do dispositivo, para que suas consultas de mídia funcionem conforme o esperado.

Portanto, você deve sempre incluir a linha de HTML acima no cabeçalho de seus documentos.

Existem outras configurações que você pode usar com a meta tag viewport, no entanto, em geral, a linha acima é o que você deseja usar.

* initial-scale: define o zoom inicial da página, que definimos como 1.
* height: define uma altura específica para a viewport.
* minimum-scale: define o nível mínimo de zoom.
* maximum-scale: Define o nível máximo de zoom.
* user-scalable: Impede o zoom se definido como no.

Você deve evitar o uso minimum-scalede maximum-scale, e em particular a configuração user-scalablede no. Os usuários devem ter permissão para aumentar ou diminuir o zoom conforme necessário; evitar isso causa problemas de acessibilidade.

# 3. Conclusão

O Design responsivo refere-se a um design de site ou aplicativo que responde ao ambiente em que é visualizado. Ele engloba vários recursos e técnicas de CSS e HTML, e agora é basicamente como construímos sites por padrão. Considere os sites que você visita em seu telefone - provavelmente é bastante incomum encontrar um site que é a versão para desktop reduzida ou onde você precisa rolar para os lados para encontrar coisas. Isso ocorre porque a web mudou para essa abordagem de design responsivo.

Também ficou muito mais fácil obter designs responsivos com a ajuda dos métodos de layout que você aprendeu nessas lições. Se você é novo no desenvolvimento web hoje, você tem muito mais ferramentas à sua disposição do que nos primórdios do design responsivo. Portanto, vale a pena verificar a idade de todos os materiais que você está referenciando. Embora os artigos históricos ainda sejam úteis, o uso moderno de CSS e HTML facilita muito a criação de designs elegantes e úteis, independentemente do dispositivo com o qual seu visitante visualiza o site.

# 4. Referências:

Design Responsivo - Aprendendo desenvolvimento web. Disponível em < https://developer.mozilla.org/pt-BR/docs/Learn/CSS/CSS\_layout/Responsive\_Design> Acesso em 25/08/2022.

O Guia para Iniciantes em Design Responsivo da Web 2022. Disponível em < https://kinsta.com/pt/blog/design-responsivo-web/> Acesso em 25/08/2022.