

FACULDADE DE TECNOLOGIA DE SÃO PAULO

NICHOLAS GERADE

Responsividade em Aplicações Web e Mobile

São Paulo – SP

2024

FACULDADE DE TECNOLOGIA DE SÃO PAULO

NICHOLAS GERADE

Responsividade em Aplicações Web e Mobile

Monografia realizada como requisito para a obtenção da graduação de tecnólogo no curso de Análise e Desenvolvimento de Sistemas.

Orientador: Professor(a). Me. David Tsai

São Paulo – SP

2024

FACULDADE DE TECNOLOGIA DE SÃO PAULO

NICHOLAS GERADE

Responsividade em Aplicações Web e Mobile

Trabalho submetido como exigência parcial para a obtenção do Grau de Tecnólogo em
Análise e Desenvolvimento de Sistemas.

Parecer do Professor Orientador

Conceito/Nota Final: **9,0 (Nove)**

**Atesto o conteúdo contido na mídia entregue pelo aluno e assinado por mim para
avaliação do TCC.**

**Estou ciente de que se o aluno não tiver entregado a mídia conforme as regras do Roteiro
ele estará reprovado na disciplina mesmo que esteja aprovado por mim.**

Orientador: Professor(a) Me. David Tsai

SÃO PAULO, 17 de junho de 2024.

DEDICATÓRIA

Dedico este trabalho a todos que me apoiaram e incentivaram ao longo da minha jornada. Em especial, dedico aos meus pais, pelo amor e apoio, aos meus professores e orientador, pela orientação sábia e pelo estímulo à busca pelo conhecimento, à minha namorada pelo suporte e amor, e aos meus amigos pela motivação mútua e amizade.

A vocês, minha gratidão eterna.

AGRADECIMENTOS

Expresso minha gratidão a todos que contribuíram para a realização deste projeto. Em primeiro lugar, agradeço a Deus, por me conceder a possibilidade da vida.

À minha família, pelo amor, apoio e incentivo, especialmente aos meus pais. À minha namorada, pelo suporte e carinho. Aos meus amigos, por proporcionarem momentos de alegria e descontração.

Ao meu orientador, por compartilhar seu conhecimento e experiência comigo. Sem as suas orientações, este trabalho não seria possível. Agradeço ao Professor e Mestre David Tsai, por sua paciência, dedicação e sabedoria.

Também expresso minha gratidão à Fatec São Paulo, por incentivar e viabilizar a realização deste projeto, oferecendo o suporte necessário para que este trabalho se tornasse realidade.

A vocês, minha gratidão eterna.

EPÍGRAFE

“Continue faminto, continue tolo”.

Steve Jobs, Discurso de formatura na
Universidade de Stanford, 2005.

LISTA DE FIGURAS

Figura 1 - Porcentagem de brasileiros com mais de dez anos com acesso à internet por dispositivo.....	15
Figura 2 - Navegador Mosaic	19
Figura 3 - Logo das ferramentas HTML, CSS e Javascript	20
Figura 4 - Evolução da porcentagem do tráfego na internet de dispositivos móveis de 2012 a 2022	27
Figura 5 - Exemplo de uma página web sem a tag <meta viewport> em oposição à mesma página com a tag presente.....	30
Figura 6 - Principais tags semânticas e como podem ser implementadas em uma página como exemplo	31
Figura 7 - Exemplo de uma media query sendo aplicada na prática com código CSS	32
Figura 8 - Resumo dos principais códigos flexbox utilizados.....	33
Figura 9 - Exemplo de como a Grid pode funcionar de acordo com os elementos de uma página web	34
Figura 10 - Principais medidas relativas utilizadas	35
Figura 11 - Ilustração de como os breakpoints são determinados pelo tipo de dispositivo.....	37
Figura 12 - Representação de como cada ferramenta tem a sua própria função	38
Figura 13 - Selo W3C certificando que o site foi verificado quanto à sintaxe CSS.....	40
Figura 14 - Logo das Ferramentas: Adobe XD e Figma	41
Figura 15 - Página inicial do Google Search Console baseado na análise de uma página na web	44
Figura 16 - Métricas para Computador do site de Portfólio analisado pela ferramenta Google Lighthouse antes da aplicação das práticas estudadas.....	53
Figura 17 - Métricas para Celular do site de Portfólio analisado pela ferramenta Google Lighthouse antes da aplicação das práticas estudadas.....	54
Figura 18 - Métricas para Computador do site de Portfólio analisado pela ferramenta Google Lighthouse depois da aplicação das práticas estudadas.	56
Figura 19 - Métricas para Celular do site de Portfólio analisado pela ferramenta Google Lighthouse depois da aplicação das práticas estudadas.	57

LISTA DE ABREVIATURAS E SIGLAS

HTML	Hyper Text Markup Language
CSS	Cascading Style Sheets
HTTPS	HyperText Transfer Protocol Secure
JSON	JavaScript Object Notation
JPEG	Joint Photographic Experts Group
PNG	Portable Network Graphics
GIF	Graphics Interchange Format
WebP	Web Picture format
AVIF	AV1 Image File Format
SVG	Scalable Vector Graphics
PWA	Progressive Web App
API	Application Programming Interface
W3C	World Wide Web Consortium
WCAG	Web Content Accessibility Guidelines
SEO	Search Engine Optimization
FCP	First Contentful Paint
LCP	Largest Contentful Paint
FID	First Input Delay
CLS	Cumulative Layout Shift
TBT	Total Blocking Time
URL	Uniform Resource Locator

RESUMO

O presente trabalho tem como objetivo analisar e estudar a importância da responsividade para sites na web e mobile, isto é, a capacidade de uma aplicação web se adaptar a diversos tipos de dispositivos e telas, desta forma, analisando técnicas e ferramentas como HTML, CSS, JavaScript, Figma e Frameworks, além do papel de diretrizes e padrões criados por organizações como a W3C (World Wide Web Consortium) e WCAG (Web Content Accessibility Guidelines). Além de comunidades online como: Stack Overflow, GitHub e fóruns especializados, cujo servem como base no apoio a desenvolvedores na aplicação de boas práticas. Estas plataformas facilitam a troca de conhecimento, resolução de problemas e colaboração entre os desenvolvedores.

O estudo de caso foi realizado para aplicar e validar as técnicas analisadas, a partir da ferramenta Google Lighthouse, usada para identificar pontos de foco para melhoria baseado nas seguintes métricas: desempenho, acessibilidade, práticas recomendadas e SEO (Search Engine Optimization), tópicos que também são explorados em detalhes nesse trabalho.

A pesquisa destaca a importância da melhoria contínua dentro do contexto de responsividade para o desenvolvimento web e mobile, além da necessidade de uma abordagem integrada que considere diretrizes estabelecidas e novas práticas emergentes. Seu objetivo é garantir um desempenho excepcional e experiência de usuário de alta qualidade em todas as plataformas e dispositivos, promovendo um ambiente digital mais inclusivo e adaptável.

Palavras-chave: Responsividade, Adaptação, Web, Mobile, Inclusão, Aplicações Digitais, Diretrizes, Padrões, Tecnologias Assistivas.

ABSTRACT

The present work aims to analyze and study the importance of responsiveness for web and mobile sites, that is, the ability of a web application to adapt to various types of devices and screens. This involves analyzing techniques and tools such as HTML, CSS, JavaScript, Figma, and frameworks, as well as the role of guidelines and standards created by organizations such as the W3C (World Wide Web Consortium) and WCAG (Web Content Accessibility Guidelines). Additionally, online communities like Stack Overflow, GitHub, and specialized forums serve as a foundation in supporting developers in the application of best practices. These platforms facilitate knowledge exchange, problem-solving, and collaboration among developers.

A case study was conducted to apply and validate the analyzed techniques using the Google Lighthouse tool, which is used to identify areas for improvement based on the following metrics: performance, accessibility, best practices, and SEO (Search Engine Optimization). These topics are also explored in detail in this work.

The research highlights the importance of continuous improvement within the context of responsiveness for web and mobile development, as well as the need for an integrated approach that considers established guidelines and emerging practices. Its goal is to ensure exceptional performance and high-quality user experience across all platforms and devices, promoting a more inclusive and adaptable digital environment.

Keywords: Responsiveness, Adaptation, Web, Mobile, Inclusion, Digital Applications, Guidelines, Standards, Assistive Technologies.

SUMÁRIO

1. INTRODUÇÃO	13
2. PROPOSTA DE TRABALHO E METODOLOGIA	13
2.1. OBJETIVO.....	13
2.1.1. OBJETIVOS ESPECÍFICOS	14
2.2. JUSTIFICATIVA.....	14
2.3. METODOLOGIA	16
2.4. HIPÓTESES.....	16
3. DESAFIOS COMUNS.....	17
4. FUNDAMENTAÇÃO TEÓRICA	18
4.1. CRIAÇÃO DA WEB	18
4.2. HISTÓRIA DA RESPONSABILIDADE.....	20
4.2.1. W3C	22
4.2.2. WCAG.....	23
4.3. PILARES DO DESENVOLVIMENTO RESPONSIVO	24
4.3.1. ADAPTAÇÃO.....	25
4.3.2. LAYOUT FLUIDO	25
4.3.3. EXPERIÊNCIA DO USUÁRIO	25
4.3.4. MOBILE FIRST.....	26
4.3.5. ACESSIBILIDADE	28
4.4. FERRAMENTAS E FRAMEWORKS	29
4.4.1. HTML	29
4.4.2. CSS.....	31
4.4.3. BOOTSTRAP.....	35
4.4.4. JAVACRIPT	37
4.4.5. W3C MARKUP VALIDATION SERVICE	38
4.4.6. FIGMA E ADOBE XD	40
5. SEO E RESPONSABILIDADE.....	42
5.1. DEFINIÇÃO DE SEO	42
5.2. BOAS PRÁTICAS PARA SEO (SEARCH ENGINE OPTIMIZATION).....	45
6. EVOLUÇÃO DAS DIRETRIZES DE RESPONSABILIDADE.....	47
6.1. W3C (WORLD WIDE WEB CONSORTIUM).....	47
6.2. WCAG.....	48
6.3. OUTRAS DIRETRIZES E PADRÕES INTERNACIONAIS	51
7. CASO DE ESTUDO	52
7.1. TÉCNICAS APLICADAS	54
7.2. RESULTADOS OBTIDOS PARA DESKTOP	55
7.3. RESULTADOS OBTIDOS PARA DISPOSITIVOS MÓVEIS	57

7.4. RESULTADOS GERAIS	58
8. CONCLUSÃO	59
REFERÊNCIAS BIBLIOGRÁFICAS.....	61

1. INTRODUÇÃO

O tema de responsividade é de grande relevância visto a necessidade de fazer com que aplicações web e mobile se adaptem ao que o usuário necessita em seu cotidiano, além de garantir com que todas as pessoas, incluindo portadores de deficiência física ou cognitiva tenha acesso igualitário a todo tipo de informação disponível na internet através de uma aplicação web ou mobile.

A partir disso é possível relacionar o significado de responsividade com adaptação, ou seja, a capacidade de qualquer tipo de dispositivo móvel ou aplicação web se adaptar a diferentes espécies de dispositivos e telas, como celulares, tablets e computadores. Esta demanda é promovida pela abundante gama de dispositivos web ou mobile utilizados atualmente, com isso, há diversos obstáculos a serem ultrapassados, por exemplo, um público com perda da visão total ou parcial, surdos, mudos, e daltônicos, assim traz ao desenvolvedor a responsabilidade de integrar essa adaptação à estrutura de sua aplicação e ao desenvolver seu código-fonte.

O acesso à web se trata de um meio de inclusão do usuário na sociedade, o que permite acessar conteúdos e conhecimentos que sem a internet não seriam possíveis serem acessado, além de ser uma ferramenta de comunicação e socialização com todo o mundo. Entretanto, muitas aplicações têm sua capacidade de adaptação limitada, ou seja, não atendem àqueles que precisam dessa inclusão. Contudo, existem tecnologias visando resolver este problema, cujo principal objetivo é exatamente auxiliar o usuário que tem uma dificuldade com acesso ou uso em utilizar a aplicação de forma que satisfaça a sua necessidade.

Portanto, é importante educar os desenvolvedores e criadores de aplicações e sistemas a respeitarem e seguirem os princípios e diretrizes de responsividade, além de boas práticas para desenvolvimento web e mobile.

2. PROPOSTA DE TRABALHO E METODOLOGIA

2.1. OBJETIVO

Analisar de forma aprofundada e específica a importância da responsividade para aplicações Web e Mobile, com o foco em entender como interfaces digitais acessíveis e responsivas afetam a experiência do usuário com os diversos tipos existentes de telas.

2.1.1. OBJETIVOS ESPECÍFICOS

1. Analisar diretrizes e padrões internacionais como WCAG (Web Content Accessibility Guidelines);
2. Estudar práticas comuns de desenvolvimento responsivo web e mobile;
3. Avaliar o desempenho de aplicações responsivas e acessíveis em comparação a aplicações ausentes dessas práticas;
4. Identificar problemas e desafios na implementação de acessibilidade e responsividade dentro do desenvolvimento web ou mobile;
5. Propor alternativas visando aperfeiçoar a integração entre acessibilidade e responsividade para oferecer aplicações mais inclusivas.

2.2. JUSTIFICATIVA

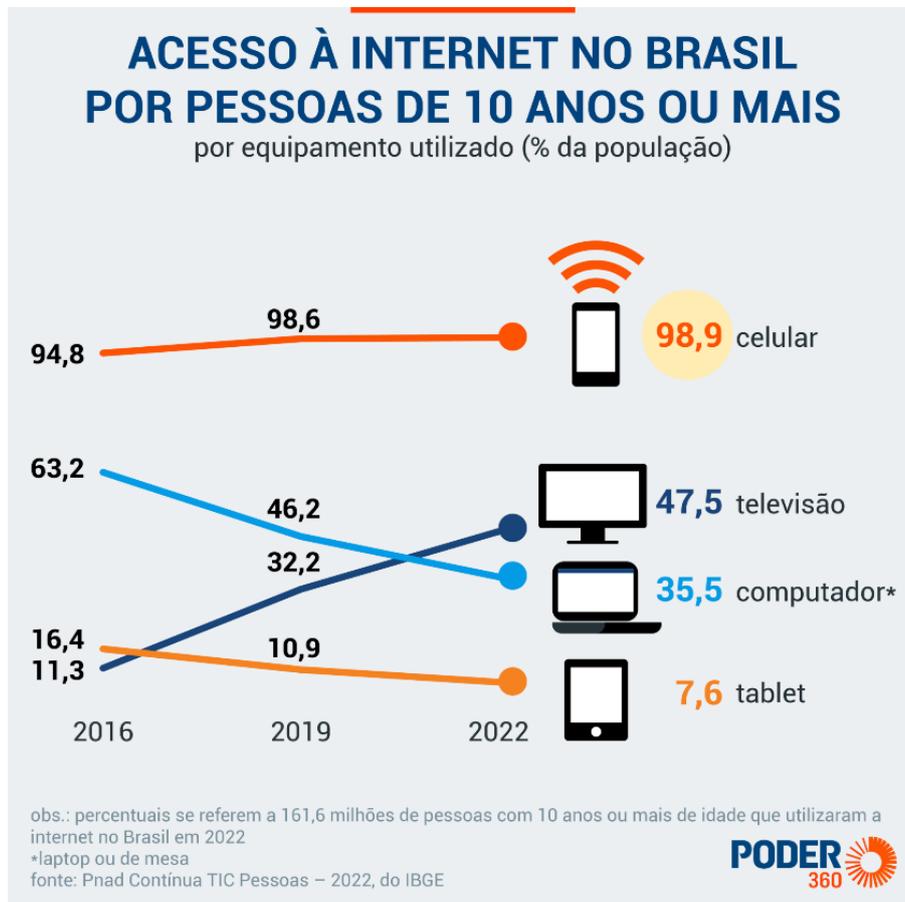
A responsividade para aplicações web e mobile são de suma importância, visto seu impacto na era digital, onde todos tem acesso à informação, dados e serviços de forma totalmente digital, visto isso, é de extrema importância que todos tenham acesso a este direito e sejam incluídos. Desta forma pessoas portadoras de qualquer tipo de deficiência física ou atraso cognitivo tem uma dificuldade a mais para acessar e interagir com aplicações web ou mobile, por isso é importante assegurar este direito aos mesmos.

Também é importante analisar a parte operacional das empresas, onde uma aplicação digital que não seja responsiva e acessível resulta na perda de clientes, e conseqüentemente, oportunidades de negócio.

Ao desenvolver e manter aplicações web e mobile, as empresas não apenas estão atendendo às necessidades de usuários com deficiência, mas também estão cumprindo com suas responsabilidades legais e éticas, investir em responsividade amplia o alcance do público-alvo, e promove a inclusão e a diversidade, contribuindo para uma sociedade mais justa e equitativa. Em uma pesquisa do IBGE (Instituto Brasileiro de Geografia e Estatística) mostrou que que 98,9% dos brasileiros com pelo menos 10 anos de idade têm acesso à internet pelo celular. Isto mostra que o mercado para desenvolvimento de aplicações voltadas aos dispositivos móveis é enorme.

Além disso, é nítido uma grande dependência do acesso à internet exclusivamente a partir de dispositivos móveis, não apenas como um celular, mas também tablets, o que contrasta a grande diferença do acesso por computadores pessoais, mostrando uma oportunidade para desenvolvedores desenvolverem aplicações responsivas para dispositivos móveis, já que maior parte do uso é através dos mesmos.

Figura 1 - Porcentagem de brasileiros com mais de dez anos com acesso à internet por dispositivo



Fonte: <https://www.poder360.com.br/tecnologia/989-dos-brasileiros-acessam-internet-pelo-celular/#:~:text=Embora%20o%20pa%C3%ADs%20tenha%20apresentado,dos%20dispositivos%20m%C3%B3veis%20no%20Brasil.> (Poder 360)

2.3. METODOLOGIA

O presente trabalho terá como metodologia de pesquisa a revisão bibliográfica de artigos acadêmicos, livros, estudos e documentos relacionados ao tema de responsividade, visando fixar uma base sólida sobre as diversas teorias e práticas aplicadas na área.

Desta forma é importante analisar a literatura existente relacionada ao tema, e verificar por possíveis erros e falhas, investigando as melhores práticas e teorias relacionadas ao tema, tendo um foco prioritário para diretrizes e padrões, como o WCAG e as diretrizes da W3C (World Wide Web Consortium).

Tendo um estudo aprofundado das teorias e práticas, se torna possível realizar uma pesquisa de campo, envolvendo entrevistas com usuários locais, a fim de levantar dados e principalmente, amostras sobre como a responsividade está sendo aplicadas em aplicações web e mobile no cotidiano.

2.4. HIPÓTESES

Este trabalho tem como objetivo central analisar a importância de princípios para melhorar a experiência do usuário em um ambiente digital, seja web ou mobile. Portanto, as seguintes hipóteses servirão de base para estudar possíveis respostas aos desafios relacionados a responsividade e acessibilidade que podem surgir:

1. Implementação de diretrizes de responsividade. Por exemplo, a WCAG, se baseia em uma melhoria considerável na experiência do usuário, e em maior porcentagem, para aqueles cujo são portadores de deficiência física ou atraso cognitivo;
2. Desenvolvimento responsivo. Ou seja, uma aplicação que consiga se adaptar perfeitamente a qualquer tipo de dispositivo ou tela, seja para a web quanto para mobile;
3. Performance e satisfação. Aplicações digitais tendem a ter uma performance inferior quanto a usabilidade e satisfação de seu usuário em relação a aplicação responsivas acessíveis. Além de ter uma avaliação negativa em ferramentas de busca, como o Google devido a sua relação com SEO (Search Engine Optimization), ou em português: Motor de Otimização de Busca;

4. Aumentar aceitação e inclusão de público. Ao implementar a responsividade em uma aplicação, se torna possível abranger o atendimento de diversos tipos de públicos e dispositivos;
5. Impacto nas Operações de Empresas. Aplicações responsivas e acessíveis resultam em uma maior captação de clientes, incentivando o investimento em desenvolvimento responsivo e acessível, isto leva a gerar um aumento significativo em oportunidades para empresas e negócios.

3. DESAFIOS COMUNS

A implementação de responsividade em aplicações web e mobile enfrenta uma série de desafios, desde questões técnicas até considerações sobre usabilidade e acessibilidade. Neste tópico, serão discutidos alguns dos desafios mais comuns enfrentados pelos desenvolvedores ao buscar tornar suas aplicações responsivas e adaptáveis a diferentes dispositivos e telas, e em tópicos posteriores, apresentado possíveis soluções através de diversos tipos de ferramentas e trechos de código.

1. **Diversidade de Dispositivos e Tamanhos de Tela:** Um dos principais desafios é a variedade de dispositivos e tamanhos de tela disponíveis no mercado que os desenvolvedores devem se preocupar em incluir em suas aplicações. Desde smartphones e tablets até monitores de desktop de diferentes resoluções, é preciso garantir que suas aplicações funcionem de maneira adequada e visualmente atraente em todos esses tipos de telas.
2. **Performance e Velocidade de Carregamento:** A responsividade não se trata apenas de ajustar o layout para diferentes telas, mas também de garantir que a aplicação seja rápida e eficiente, especialmente em dispositivos móveis com conexões de internet mais lentas. O desafio reside em otimizar o desempenho da aplicação, mantendo-a responsiva e com tempos de carregamento mínimos.
3. **Navegação Intuitiva e Usabilidade:** Adaptar a interface de usuário para diferentes dispositivos sem comprometer a usabilidade é um desafio significativo. Os desenvolvedores precisam garantir que a navegação seja intuitiva e que os elementos da interface sejam facilmente acessíveis e entendíveis, independentemente do tamanho da tela ou do dispositivo utilizado.

4. **Acessibilidade para Usuários com Deficiências:** Garantir que a aplicação seja acessível para todos os usuários, incluindo aqueles com deficiências físicas ou cognitivas, é outro desafio crucial, o que envolve a implementação de recursos como texto alternativo para imagens, compatibilidade com leitores de tela e a consideração de requisitos de acessibilidade conforme definido pelas diretrizes como WCAG.
5. **Testes e Manutenção Contínua:** Testar a aplicação em uma ampla variedade de dispositivos e garantir que a mesma permaneça responsiva após atualizações e alterações no código são desafios contínuos. É necessário dedicar tempo e recursos para testar e manter a aplicação em diferentes ambientes, visando garantir uma experiência consistente para todo tipo de tela e usuário.
6. **Adoção de Padrões e Diretrizes:** Embora existam padrões e diretrizes estabelecidos, como o WCAG, nem sempre é fácil implementá-los de forma eficaz. Os desenvolvedores enfrentam desafios ao interpretar e aplicar esses padrões em suas próprias aplicações, especialmente em projetos complexos ou com prazos de entrega mais curtos.

Enfrentar esses desafios requer não apenas habilidades técnicas, mas também uma compreensão e interpretação profunda das necessidades e expectativas dos usuários. Ao superar estes desafios, os desenvolvedores conseguem criar aplicações web e mobile que ofereçam uma experiência mais inclusiva e satisfatória para todo tipo de usuário, independentemente do dispositivo que o mesmo está acessando sua aplicação.

No tópico seguinte será apresentada a história e fundamentação de conceitos base, e cada ferramenta e código que será aprofundado, para logo após, fundamentar como estes desafios podem ser contornados através dessas técnicas e ferramentas especificamente voltadas ao ramo de responsividade para a web e mobile.

4. FUNDAMENTAÇÃO TEÓRICA

4.1. CRIAÇÃO DA WEB

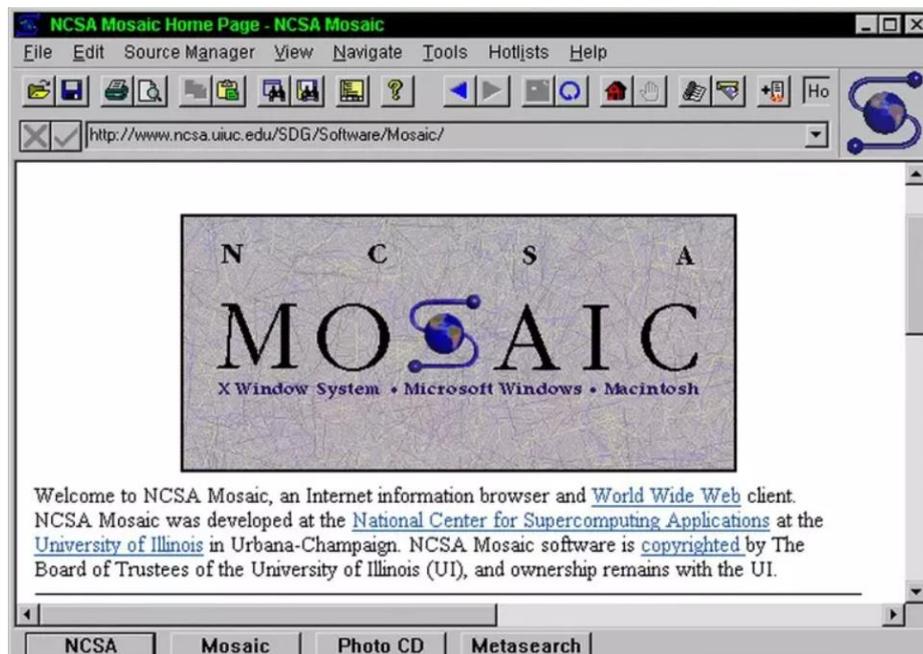
A web como é conhecida atualmente teve seu primeiro site oficial lançado em 1991 por Tim Berners-Lee, físico e cientista da computação britânico, porém o projeto da World Wide Web começou em meados de 1989, quando Tim Berners-Lee trabalhava para o CERN (Organização Europeia para Pesquisa Nuclear), e então propôs um sistema de gerenciamento

para todo tipo de informação baseado em hipertexto, o que mais tarde deu origem a três tipos de ferramentas em 1990, o HTML (Hyper Text Markup Language), o HTTP (Hypertext Transfer Protocol), e a URL (Uniform Resource Locator).

Dentro deste website continha informações iniciais sobre o projeto que estava sendo desenvolvido por Tim e sua equipe, além de fornecer uma documentação detalhada sobre como criar páginas na web, e algumas diretrizes de boas práticas.

O primeiro navegador gráfico web criado, foi o Mosaic, o que permitiu integrar uma interface gráfica de fácil uso com informações em hipertexto, o que popularizou o acesso à web ao público. Mais tarde foram lançados diversos navegadores ao público, onde cada um conta com uma característica única e exclusiva, entre alguns deles se encontram: Netscape Navigator, Internet Explorer, Mozilla Firefox, Google Chrome e outros.

Figura 2 - Navegador Mosaic



Fonte: <https://www.terra.com.br/byte/mosaic-o-navegador-que-revolucionou-a-web-completa-25-anos,85e54a2e36ea308d8609aade30c36ab3b6cizv4y.html>. (Terra)

Novas tecnologias também foram surgindo para implementar cada vez mais a web, como por exemplo o CSS (Cascading Style Sheets) e o Javascript, cujo permitiram o controle de estilo de hipertexto e a interatividade entre o cliente e a base de dados da aplicação.

Atualmente existem diversos frameworks e bibliotecas para a web que visam facilitar o desenvolvimento dessas aplicações e aprimorar a interatividade com o usuário, como o React e Angular, as quais são frameworks de desenvolvimento do Javascript.

Atualmente o próprio HTML se encontra em sua 5ª versão, onde são suportados elementos semânticos, suporte para áudio e vídeo, e até mesmo geolocalização, diferentemente de versões mais antigas, o que mostra a evolução de uma das principais ferramentas para o desenvolvimento web, lembrando que o mesmo segue as diretrizes da ISO/IEC 15445:2000, no qual implementa padronizações a serem seguidas pelos desenvolvedores que utilizam a ferramenta.

Figura 3 - Logo das ferramentas HTML, CSS e Javascript



Fonte: <https://stock.adobe.com/br/images/html5-css3-js-icon-set-web-development-logo-icon-set-of-html-css-and-javascript-programming-symbol/321243084>. (Adobe Stock)

4.2. HISTÓRIA DA RESPONSIVIDADE

Com a popularização da internet para dispositivos móveis, como o celular nos anos 2000, surge a necessidade de adaptar o conteúdo na web para esses tipos de dispositivos, e quando há uma falta de padronização, surge diversos desafios que tornam o acesso à web um empecilho para os usuários.

À medida que os celulares e tablets foram se desenvolvendo conforme a tecnologia progredia, a demanda por aplicações responsivas se tornou cada vez maior, isto torna a necessidade de tornar conteúdos disponíveis para todo tipo de dispositivo, onde inclusive a maior porcentagem do acesso à web é realizada a partir de celulares.

Uma das primeiras vezes que o termo “responsividade” foi mencionado, foi em um artigo de 2010 publicado por Ethan Marcotte, designer gráfico estadunidense, onde foi abordado a ideia de que o conceito de design responsivo deveria ser flexível, de forma que todo site consiga se adaptar de forma automática a diferentes tipos e tamanhos de telas e dispositivos. Essa ideia, se torna possível por causa de técnicas como layout fluido, imagens flexíveis e media queries. De forma que revolucionou a abordagem de desenvolvedores e designer, tornando a responsividade praticamente uma lei a ser seguida por qualquer um que deseja estudar ou desenvolver sobre web design responsivo, o que segundo o próprio autor afirma:

Recentemente, uma disciplina emergente chamada "arquitetura responsiva" começou a questionar como espaços físicos podem responder à presença de pessoas que passando entre eles. Através de uma combinação de robótica embutida e materiais tensionados, arquitetos estão experimentando com instalações de arte e estruturas de parede que se curvam, flexionam e se expandem conforme multidões se aproximam delas. Sensores de movimento podem ser combinados com sistemas de controle climático para ajustar a temperatura de uma sala e a iluminação ambiente conforme ela se enche de pessoas.

(MARCOTTE, Ethan, 2010).

Em relação à arquitetura responsiva citada pelo autor, é possível relacioná-la como uma metáfora, onde a capacidade de websites se adaptarem ao usuário, depende de como essa prática é realizada a partir da tecnologia utilizada pelos desenvolvedores da aplicação, e conforme novas tecnologias e práticas surgem, a abordagem utilizada pelos criadores também muda de acordo.

No artigo, o autor também cita a partir de exemplos de websites existentes, práticas dentro das tecnologias HTML, e CSS, de como conteúdo de media, como imagens podem ser flexibilizadas através dessas ferramentas.

4.2.1. W3C

A W3C (World Wide Web Consortium) é uma organização internacional sem interesse monetário criada em 1994 por Tim Berners-Lee no MIT (Massachusetts Institute of Technology) em parceria com a CERN, com o objetivo de garantir a evolução e estabilidade na web, nela foi definida as primeiras diretrizes e padrões a serem seguidos na internet, principalmente para as ferramentas HTML, CSS e HTTP. Atualmente, existem diversas tecnologias que auxiliam a evolução e desenvolvimento de novos websites, onde todos devem atender às expectativas definidas pelas W3C, isso facilita a padronização para desenvolvimento. Algumas ferramentas emergentes são: XML, SVG, XPath, e entre outras.

O principal pilar proposto pela W3C é a evolução contínua, ou seja, todos os padrões e diretrizes elaborados, visam refinar a criação de novas aplicações com diferentes tecnologias, independentemente de qual o desenvolvedor usar, além de adotar uma abordagem que visa consentir entre especialistas, acadêmicos, e outras partes interessadas, como investidores, assim apoiando esta padronização em massa internacionalmente.

Hoje estou orgulhoso do profundo impacto que o W3C teve, suas muitas conquistas realizadas com nossos Membros e o público, e aguardo com expectativa as contínuas melhorias capacitadoras que o W3C permite ao lançar sua própria organização sem fins lucrativos de interesse público, construindo sobre 28 anos de experiência.

(BENERS-LEE, Tim, 2023).

A W3C se torna tão importante para toda a comunidade de desenvolvedores de aplicações web e mobile, quanto aos usuários, por sua relevância em oferecer a base para todo o desenvolvimento desde sua criação, ou seja, o desenvolvimento responsivo acaba por ser facilitado pela existência de suas regras e diretrizes.

Além de ser responsável por toda a padronização, a W3C possui a W3Schools, onde também é oferecido conteúdo gratuito sobre diversas linguagens de programação, inclusive para desenvolvimento responsivo para a web e mobile, estando disponível a qualquer um com acesso à internet, e um dispositivo para aprender o conteúdo, junto disso há exercícios, tutoriais, certificações de linguagens de programação e tecnologias, e fóruns online onde qualquer pessoa que deseja aprender ou tirar dúvidas, pode entrar em contato com desenvolvedores mais experientes e entusiastas.

4.2.2. WCAG

A WCAG (Web Content Accessibility Guidelines) é um conjunto de boas práticas de acessibilidade criada pela W3C, por meio do WAI (Web Accessibility Initiative). Sua primeira versão foi lançada em 1999, tendo sua versão mais recente sendo lançada no ano de 2018. Essas diretrizes baseiam-se em garantir que a acessibilidade exista para qualquer dispositivo móvel, o que influencia de forma direta na responsividade de uma aplicação, assim, para que uma aplicação seja responsiva, ela também precisa ser acessível, e vice-versa.

De acordo com as regras da WCAG, uma aplicação deve ser dividida em 4 eixos: Perceptibilidade, Operabilidade, Compreensibilidade e Robustez, além de haver outras subseções, todos estes eixos dividem e organizam a aplicação visando a melhor experiência possível do usuário dentro do programa.

O 1º eixo referente à perceptibilidade, estrutura que qualquer sistema deve apresentar suas informações e componentes de forma que todo usuário consiga o entender de forma clara, sem impedimentos ou segundas intenções. Um exemplo deste eixo citado são aplicações web que possuem um contraste ruim, isto é, a forma que o texto do site se diferencia da cor de fundo não tornando difícil a visualização do texto, como um texto de cor amarela, sobre um fundo com cor branca. A melhor solução neste caso é aumentar o contraste entre o texto e o fundo, como um texto preto sobre um fundo branco, algumas ferramentas verificativas ajudam a analisar se o contraste atinge um nível satisfatório de acordo às diretrizes da WCAG.

Operabilidade se refere à capacidade dos usuários interagirem com a aplicação de forma eficiente, fornecendo diversas opções de navegação, garantindo que todos os elementos sejam interativos por meio de diferentes dispositivos de entrada, como mouse, teclado ou até mesmo touchscreen, permitindo que o usuário consiga controlar e personalizar sua interface de acordo com suas necessidades individuais, um site que não permite a navegação por teclado, dificulta a navegação para usuários com deficiências motoras acessarem o conteúdo, nessa situação é imprescindível que todos os elementos interativos do site possam ser acessados e operados por qualquer usuário, não apenas com um teclado, mas também qualquer ferramenta que possa estar o assistindo, como por exemplo, as teclas de tabulação, atalhos de teclado, e até mesmo ferramentas de sopro ou voz para tetraplégicos.

A compreensibilidade visa tornar a aplicação fácil de compreender e navegar para todos, independentemente de seu nível de habilidade ou experiência, o que envolve fornecer instruções claras e consistentes, utilizando uma estrutura de navegação lógica e intuitiva, além evitar

terminologia técnica que possa eventualmente confundir o usuário, também é muito importante fornecer um feedback claro e imediato para suas ações, para que saibam o que está acontecendo ao realizar uma função. Um exemplo deste eixo é um formulário de dados que não fornece instruções claras sobre o formato esperado para os dados a serem inseridos, o botão de enviar do formulário também é incluído nesse caso, visto que uma vez que o usuário envia seus dados, é necessária uma confirmação do envio, o que pode trazer confusão e erros por parte dos usuários. Para resolver isso, o formulário deve incluir instruções claras e concisas ao lado de cada campo, explicando o tipo de informação solicitada e o formato esperado para garantir que os usuários compreendam como preencher corretamente este formulário.

O princípio de robustez diz respeito à capacidade do sistema lidar com uma variedade de tecnologias, navegadores e dispositivos, garantindo uma experiência consistente e funcional, envolvendo o uso de padrões na web e diferentes tecnologias, além de realizar testes regulares na aplicação em uma variedade de ambientes para identificar e corrigir problemas de compatibilidade. Garantir a robustez da aplicação é fundamental para garantir que sempre se permaneça acessível e utilizável em todo tipo de situação, por exemplo, um site que depende fortemente de recursos Javascript avançados que podem não ser suportados por todos os navegadores ou dispositivos, resulta em uma experiência quebrada para alguns usuários, para que a robustez do site seja garantida, os desenvolvedores devem usar tecnologias para a web compatíveis com diferentes navegadores e testar regularmente a aplicação em uma variedade de navegadores e dispositivos para identificar e corrigir problemas de compatibilidade, garantindo uma experiência consistente e funcional para todos os usuários.

4.3. PILARES DO DESENVOLVIMENTO RESPONSIVO

O desenvolvimento responsivo, tanto para web, como para mobile, seguem primeiramente padrões e diretrizes de organizações como a W3C e WCAG, como já foi citado anteriormente, mas existem pilares que regem o seu desenvolvimento, cujo não necessariamente estão documentados de forma que qualquer usuário possa estudar estes princípios sem um estudo mais específico, mas ainda sim são essenciais para trazer uma experiência consistente e eficiente ao usuário.

Desta forma, os pilares que serão citados a seguir serão desenvolvidos a partir de uma análise mais aprofundada das diretrizes mais conhecidas atualmente para a internet, como a

W3C, além de exemplos práticos de como aplicações responsivas realmente funcionam e são criadas.

4.3.1. ADAPTAÇÃO

A palavra adaptar vem do latim, “adaptare” o que significa ajustar, ajuntar ou até mesmo, apropriar. Assim, sua essência está na ideia de ajustar, modificar algo para que se adeque ao ambiente em que se está presente, como em uma situação ou condição.

Ao ser aplicado para a web dentro do contexto de responsividade, a palavra “adaptar” ganha ainda mais relevância, portanto, o desenvolvimento de sites e aplicativos responsivos tem como sua base, adaptar o conteúdo e o layout de forma que se adequem às diferentes condições de visualização, o que é essencial, e garante uma experiência de usuário consistente e eficiente em qualquer dispositivo.

4.3.2. LAYOUT FLUIDO

Em contraste com layouts fixos, elementos fluidos têm dimensões dinâmicos e se ajustam às diferentes resoluções de tela, de forma que permite a expansão e contração de elementos proporcionalmente conforme o tamanho da tela do dispositivo.

Para que um layout fluido possa ser implementado por um desenvolvedor, unidades de medidas flexíveis podem ser utilizadas, como porcentagens ao invés de valores fixos, como pixels, o que colabora para que os elementos se ajustem mais facilmente ao se adequarem a diferentes telas. É importante considerar a usabilidade de cada tela e diferentes dispositivos, o que pode levar ao rearranjo de elementos, mídia e conteúdo dentro da aplicação. Portanto, é sempre essencial que o desenvolvedor leve em conta a experiência do usuário de acordo com o dispositivo utilizado, o que será discutido com mais detalhes no próximo tópico.

4.3.3. EXPERIÊNCIA DO USUÁRIO

A experiência do usuário tem como foco exclusivo, a satisfação do usuário ao interagir com um site ou aplicação, visando a facilidade no uso, eficiência e bem-estar. Aplicando ao contexto da responsividade, o foco da experiência do usuário se amplia para abranger também

o design do sistema, garantindo uma experiência boa. Para otimizar essa experiência, desenvolvedores devem considerar alguns aspectos como:

1. **Apresentação Visual:** Estando em 1º como um dos principais diferenciais de um site para o usuário, o que para atrair sua atenção, é necessária uma atraente apresentação visual, o que faz com que o usuário escolha entrar, navegar e até mesmo comprar em uma aplicação web ao invés de outra que não tem uma forte identidade visual estabelecida;
2. **Performance:** Quando um usuário entra em um website, sua primeira expectativa é de que seja carregado rápido, até no máximo 3 segundos. Em uma pesquisa realizada no Canadá pela Microsoft, determinou que o tempo de atenção dos seres humanos já se encontra menor do que o de um peixe dourado, ou seja, no máximo de 9 segundos. Isso mostra que a sociedade está cada vez mais em busca de novos estímulos, novos produtos, novos sites, e se a aplicação web não possui performance para atender a este requisito, estes possíveis clientes irão optar por outras escolhas;
3. **Navegabilidade:** Ou seja, a forma com que os usuários navegam pelo site deve ser intuitiva, o que envolve organizar o conteúdo de forma lógica e entendível. Quando uma aplicação tem sua navegabilidade aplicada de forma pobre, sua utilidade e credibilidade são afetadas, de modo que afeta sua taxa de conversão tanto para a área de vendas, quanto para a questão de acessos por período;
4. **Interatividade:** O usuário deve receber uma resposta em tempo real para toda ação que o mesmo realizar durante sua navegação no sistema, alguns tipos de resposta incluem animações, microinterações, mensagens informativas, e entre outras. Essas respostas dentro da aplicação demonstram interatividade e principalmente responsabilidade com a experiência que o usuário terá com o sistema.

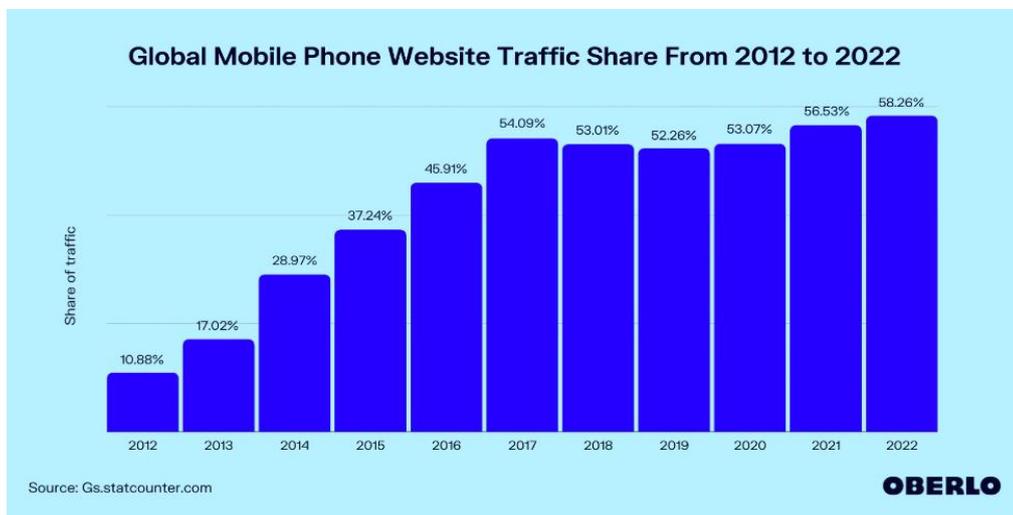
4.3.4. MOBILE FIRST

O conceito de Mobile First consiste em uma abordagem do desenvolvimento que prioriza a criação da aplicação desde o seu protótipo, baseando-se na usabilidade para telas de dispositivos móveis, ou seja, celulares e tablet, dessa forma incentivando uma maior priorização da experiência para os usuários de dispositivos móveis, visto sua importância e relevância no mercado atual.

Uma vez que o escopo da aplicação web para dispositivos móveis esteja estabelecido, os desenvolvedores expandem e aprimoram a experiência para outros tipos de dispositivos, como desktops e até mesmo para monitores maiores e televisões HD, adaptando o layout e adicionando recursos adicionais, ou o contrário caso necessário. Esse enfoque garante que os usuários móveis não sejam negligenciados e recebam uma experiência otimizada, mas também sem negligenciar que a experiência seja estendida para outros dispositivos de tela maior.

Além de melhorar a experiência do usuário, a abordagem de Mobile First também pode ter benefícios adicionais, como melhorar o desempenho do site ou aplicativo em dispositivos móveis, reduzindo o tempo de carregamento e o consumo de dados. Existem pontos que o conceito de mobile first também traz como acréscimo à aplicação web, como um melhor ranqueamento no Google através do SEO, o que mostra que a aplicação web é responsável com o indexamento de seu conteúdo, dessa maneira, tornando-o mais bem ranqueado na web e conseqüentemente, mais acessado.

Figura 4 - Evolução da porcentagem do tráfego na internet de dispositivos móveis de 2012 a 2022



Fonte: <https://www.browserstack.com/guide/how-to-implement-mobile-first-design>.

(Browser Stack)

Um aspecto que também deve ser considerado é o comportamento do usuário em relação ao site, já que o comportamento varia de acordo com cada tipo de tela, e como o uso de dispositivos móveis cresceu exponencialmente, o que ressalta a importância do conceito de mobile first.

Ao tratar de dispositivos móveis, como smartphones, usuários tendem a interagir de forma mais rápida e direta com o conteúdo, esperando uma experiência ágil, com informações facilmente acessíveis e interfaces mais simplificadas. Além disso, a navegação em dispositivos móveis geralmente é feita com toques e gestos, o que influencia a forma como os elementos interativos devem ser desenvolvimentos pelos desenvolvedores da aplicação.

Já em tablets, o comportamento pode variar mais, ainda que seja compartilhado muitas semelhanças com smartphones em termos de interação por toque. Tablets oferecem uma tela maior, o que permite uma experiência de navegação mais semelhante à de desktops. Por outro lado, em desktops e laptops, os usuários tendem a esperar uma experiência mais completa e robusta, tendo acesso a uma tela maior e um teclado físico, o que possibilita a realização de múltiplas tarefas e uma navegação mais detalhada. No entanto, também é importante considerar que, mesmo em dispositivos de tela maior, os usuários valorizam a rapidez e a eficiência na interação com o site ou aplicativo.

De acordo com Steve Krug, profissional de UX e autor do livro best-seller, *Don't Make Me Think*, o usuário têm uma grande expectativa de rapidez e eficiência ao interagir com um site ou aplicativo, independentemente do dispositivo que está sendo usado, o que mostra que mesmo que o seu comportamento se altere de acordo com o tipo de tela, a necessidade de que haja rapidez na aplicação é existente, e determinante para o usuário decidir se continuará consumindo ou não o conteúdo do site.

4.3.5. ACESSIBILIDADE

Finalizando com um dos principais pilares para o desenvolvimento responsivo, a acessibilidade garante que todo usuário, independentemente de suas habilidades ou limitações, consigam utilizar as funcionalidades de um aplicativo web como um usuário sem necessidades especiais usaria. Ao desenvolver uma aplicação responsiva existem alguns focos referente à óptica da acessibilidade que também são importantes levar em consideração, tais como:

1. Compatibilidade com tecnologias assistivas: Garantir que o site desenvolvido seja compatível com tecnologias assistivas, como leitores de tela, ampliadores, e dispositivos de entrada alternativos, garantindo que usuários com deficiência visual, auditiva ou motora consigam interpretar o conteúdo e utilizar suas funcionalidades de maneira eficaz;

2. Design inclusivo: Ou seja, implementar práticas inclusivas, como um contraste adequado entre cores, tamanhos de fonte ajustáveis, além de configurar o layout de forma que ele se torne flexível a diferentes configurações de tela;
3. Mídia alternativa: Isto é, fornecer alternativas para conteúdo multimídia, como legendas para vídeos e descrições de imagens para usuários com deficiência auditiva ou visual;
4. Navegação simplificada: Projetar uma navegação intuitiva, junto a utilização de marcadores semânticos e estrutura lógica, facilitando a orientação e compreensão do conteúdo por parte do usuário, incluindo aqueles que utilizam tecnologias assistivas;
5. Testes de usabilidade: Realizar testes de usabilidade na aplicação com diversos tipos de usuários, incluindo usuários com diferentes níveis de habilidades e necessidades de acessibilidade, visando identificar e corrigir possíveis empecilhos antes de seu “go live”.

Priorizar a acessibilidade como um dos pilares do desenvolvimento responsivo, permitindo com que os desenvolvedores possam criar experiências digitais mais inclusivas e acessíveis, atendendo às necessidades de qualquer usuário. Isso não apenas aumenta a captação de novos usuários em potencial, mas também demonstra um compromisso com a inclusão digital.

4.4. FERRAMENTAS E FRAMEWORKS

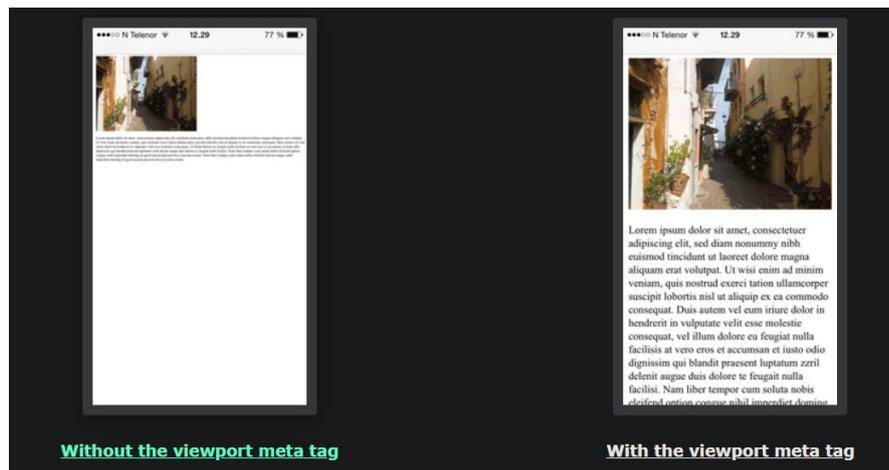
4.4.1. HTML

O uso adequado de tags HTML desempenha um papel crucial na criação de uma experiência consistente para dispositivos móveis. Estas tags oferecem uma estrutura básica para a organização e apresentação de conteúdo no website, e ao utilizá-las de forma estratégica, os desenvolvedores podem garantir que seus projetos se tornem adaptáveis a uma variedade de tamanhos de tela e dispositivos móveis. A seguir será analisado as funções que algumas das principais tags semânticas desempenham dentro do tema de responsividade.

1. Meta Viewport: Uma das tags mais importantes para o desenvolvimento responsivo em dispositivos móveis é a `<meta viewport>`. Essa tag permite com que os desenvolvedores consigam controlar como o navegador renderiza o conteúdo em diferentes tamanhos de tela. Ao atribuir a tag: `<meta name="viewport" content="width=device-width, initial-scale=1.0">` dentro da tag: `<head>`, os desenvolvedores garantem que o navegador se ajuste

automaticamente a largura do layout de acordo com a largura do dispositivo, proporcionando uma experiência de visualização otimizada em dispositivos móveis.

Figura 5 - Exemplo de uma página web sem a tag <meta viewport> em oposição à mesma página com a tag presente



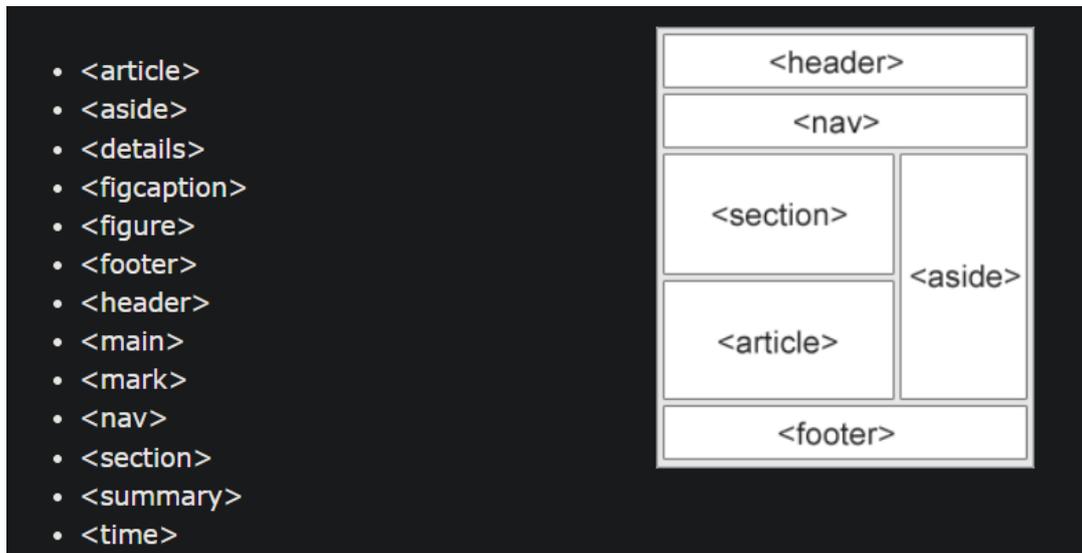
Fonte: https://www.w3schools.com/css/css_rwd_viewport.asp (W3C Schools)

O exemplo anterior demonstrado define que a cor de fundo do documento HTML seja alterada para verde claro apenas em dispositivos de tela com uma largura mínima de 480 pixels. Diversos exemplos podem ser desenvolvidos para que satisfaça qualquer necessidade do usuário.

2. Tags de Divisão de Conteúdo: Algumas tags semânticas HTML, como <header>, <nav>, <main>, <section>, <article> e <footer>, ajudam a estruturar o conteúdo de forma mais organizada, tornando-o compreensível para os usuários, e para os mecanismos de busca também, além de facilitar a navegação em dispositivos móveis. Essas tags também podem ser estilizadas de forma responsiva usando Media Queries.

No geral, o uso adequado dessas tags semânticas HTML não apenas ajuda no desenvolvimento responsivo, garantindo uma experiência consistente em diferentes tamanhos de tela, mas também pode melhorar o SEO, ajudando os mecanismos de busca a entenderem a estrutura e a relevância do conteúdo de uma página. Isso pode levar a um melhor posicionamento nos resultados de pesquisa e a uma experiência de usuário mais positiva.

Figura 6 - Principais tags semânticas e como podem ser implementadas em uma página como exemplo



Fonte: https://www.w3schools.com/html/html5_semantic_elements.asp (W3C Schools)

4.4.2. CSS

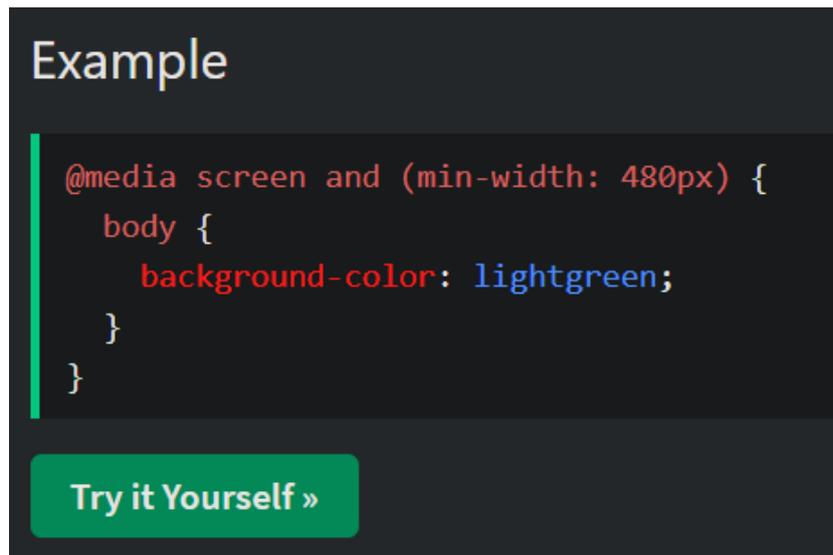
A criação de sites responsivos é um aspecto crucial do desenvolvimento web moderno, e o CSS desempenha um papel fundamental neste processo. Ao aplicar estilos específicos com base nas características do dispositivo, os desenvolvedores podem garantir uma experiência consistente e agradável para uma variedade de dispositivos, desde smartphones até desktops e televisões. Existem diversos recursos dentro do próprio CSS que podem ser aplicados nos websites de forma que atenda aquilo que o desenvolvedor deseja implementar, além de frameworks que ajudam a agilizar a produção de projetos novos de forma eficiente e confiável.

Esses frameworks são bibliotecas de código prontos que facilita a implementação de novos recursos com menos linhas de código e conseqüentemente menos trabalho, ao realizar essa integração fica mais fácil para o próprio desenvolvedor organizar o conteúdo, layout e criar componentes dentro de sua aplicação. Existem diversos frameworks atualmente, e cabe ao desenvolvedor escolher qual será utilizado, baseado em diversas métricas, como a linguagem de programação escolhida, diferenciais, vantagens e entre outros.

A seguir será explorado três dos conceitos mais utilizados para o desenvolvimento responsivo com CSS, entre eles estão: Media Queries, Flexbox e Grid Layout, e Unidades de Medida Relativas.

1. **Medias Queries:** As medias queries são recursos cruciais para o desenvolvimento responsivo, permitindo a aplicação de estilos CSS específicos com base nas características do dispositivo, como largura da tela, orientação e densidade de pixels, o que permite criar layouts que se ajustem de forma dinâmica a diferentes tamanhos de tela.

Figura 7 - Exemplo de uma media query sendo aplicada na prática com código CSS



Fonte: https://www.w3schools.com/css/css3_mediaqueries.asp (W3C Schools)

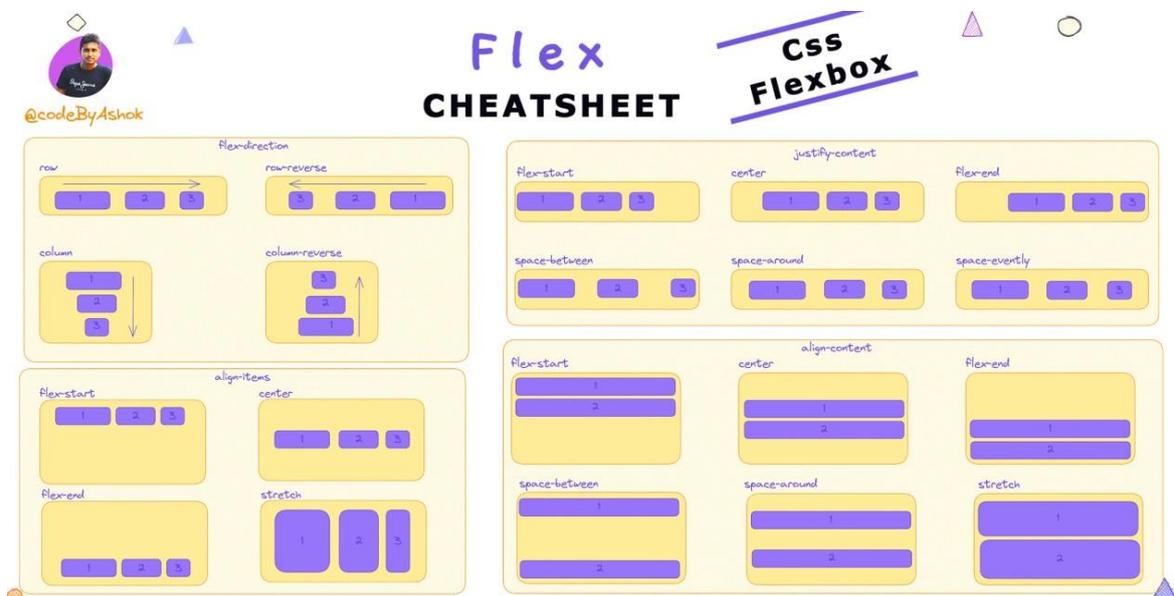
2. **Flexbox:** Além das media queries, técnicas de layout mais avançadas como o Flexbox é extremamente útil para criar designs responsivos e mais flexíveis. Esses recursos permitem o posicionamento e o dimensionamento dos elementos de forma dinâmica, facilitando a criação de layouts complexos que se ajustam automaticamente às diferentes telas dos dispositivos.

O Flexbox é um modelo projetado para organizar elementos em uma linha ou coluna. Ele permite que os itens dentro de um contêiner se ajustem de forma dinâmica ao espaço disponível e às diferentes proporções de tamanho dos elementos. O Flexbox foi criado com o intuito de aperfeiçoar a criação de layouts CSS, já que existiam poucos recursos de posicionamento de container e elementos. Alguns dos principais exemplos de código que o Flexbox oferece são:

1. **display: flex:** Aplica um modelo de layout flexível a um contêiner, o que permite que outros códigos flexbox sejam utilizados;

2. flex-direction: Define a direção principal do leiaute, por exemplo, row (linha), column (coluna), row-reverse (linha reversa), column-reverse (coluna reversa);
3. justify-content: Define o alinhamento dos itens ao longo do eixo horizontal;
4. align-items: Define o alinhamento dos itens ao longo do eixo vertical;
5. flex-grow, flex-shrink, flex-basis: Propriedades para controlar como os itens flexíveis crescem, encolhem e se baseiam em relação aos outros itens.

Figura 8 - Resumo dos principais códigos flexbox utilizados



Fonte: <https://www.linkedin.com/pulse/mastering-css-flexbox-layout-ashok-kumar-x065c/>

(Ashok Kumar – LinkedIn)

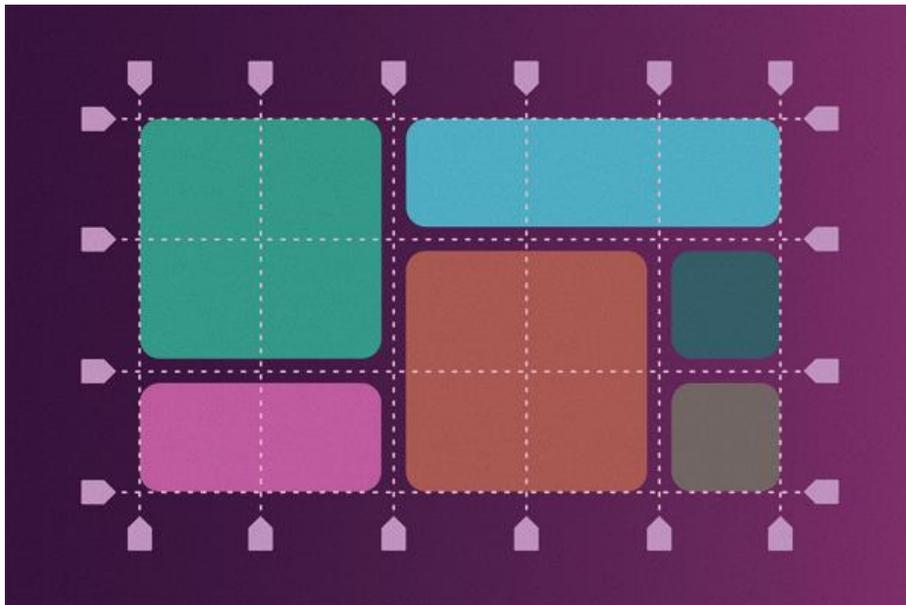
3. Grid Layout: O grid layout é um modelo bidimensional que permite organizar elementos e conteúdo de uma página na web em linhas e colunas, criando um leiaute de grade, o que oferecendo um controle mais minucioso sobre a posição e o espaçamento dos itens, facilitando a criação de leiautes mais complexos.

Enquanto Flexbox é ideal para layouts unidimensionais, como organizar itens em uma linha ou coluna, o Grid Layout é perfeito para estruturas bidimensionais, permitindo a criação de grades com colunas, linhas e espaçamentos, o que proporciona maior consistência para aplicação web, de forma que à medida que o usuário avança de uma página para a outra, os

elementos e conteúdo não mudam de largura nem saem para fora de sua visualização. Seus principais elementos incluem:

1. `display: grid`: Aplica o modelo de leiaute de grade a um contêiner;
2. `grid-template-columns`, `grid-template-rows`: Definem o número e o tamanho das colunas ou linhas na grade;
3. `grid-gap`: Especifica o espaçamento entre as células da grade;
4. `grid-column`, `grid-row`: Controla a posição de um item na grade.

Figura 9 - Exemplo de como a Grid pode funcionar de acordo com os elementos de uma página web



Fonte: <https://www.tabnews.com.br/pedroespindula/criando-malhas-com-css-grid> (Pedro Espindula – TabNews)

4. Unidades de Medida Relativas: Para garantir que os elementos da página se ajustem proporcionalmente ao tamanho da tela, é recomendável o uso de unidades de medida relativas, como porcentagens, em vez de unidades absolutas, como pixels. Isso permite que o layout se adapte de maneira mais fluida a diferentes dispositivos e tamanhos de tela.

Figura 10 - Principais medidas relativas utilizadas

Unit	Description
em	Relative to the font-size of the element (2em means 2 times the size of the current font)
ex	Relative to the x-height of the current font (rarely used)
ch	Relative to the width of the "0" (zero)
rem	Relative to font-size of the root element
vw	Relative to 1% of the width of the viewport*
vh	Relative to 1% of the height of the viewport*
vmin	Relative to 1% of viewport's* smaller dimension
vmax	Relative to 1% of viewport's* larger dimension
%	Relative to the parent element

Fonte: https://www.w3schools.com/cssref/css_units.php (W3C Schools)

As unidades relativas são influenciadas por fatores externos, o que significa que seus valores dependem de um elemento pai específico. Por exemplo, a porcentagem (%) varia em relação ao tamanho de seu elemento pai definido pelo desenvolvedor, por exemplo, se é definido como 50% a largura de um elemento filho, essa largura será metade da largura do elemento pai, que pode ser um contêiner, enquanto o view height (vh), altura da visão do usuário, varia de acordo com a altura da viewport, a atual área exibida na tela do usuário. Já as unidades absolutas são fixas e não dependem de contextos externos, já que são ancoradas em medidas físicas reais, como pixels (px) e centímetros (cm).

4.4.3. BOOTSTRAP

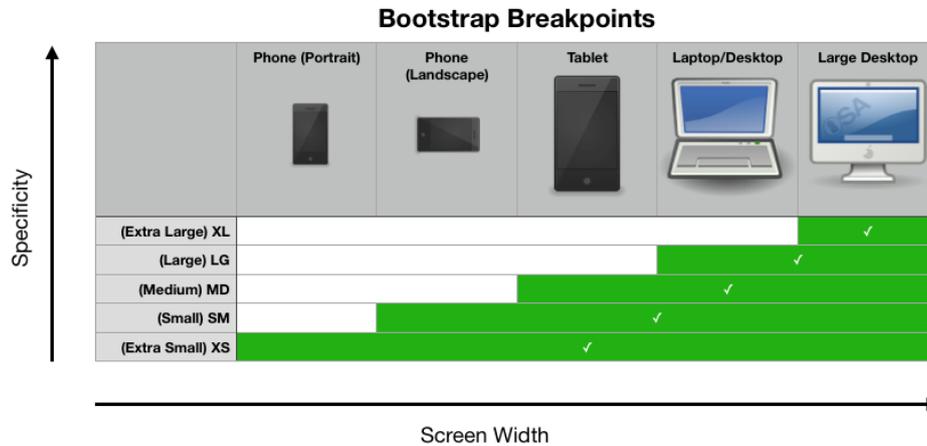
O Bootstrap é um framework CSS criado em 2010 por integrantes do Twitter, atualmente nomeado como: X, é uma das bibliotecas de código aberto mais populares e utilizadas atualmente para o desenvolvimento de novas aplicações responsivas. O Bootstrap oferece um conjunto abrangente de componentes, estilos e utilitários CSS pré-estilizados, juntamente com plugins Javascript, o que facilita a criação de layouts e telas compatíveis com diversos dispositivos.

Além de permitir a criação de sites totalmente responsivos, ele conta com o auxílio de microprocessadores de stylesheets (folhas de estilo) para um desempenho eficiente sem ignorar a implementação. Também é possível adicionar plugins Javascript para ter acesso a uma vasta biblioteca de componentes, como carrossel de imagens, botões e até mesmo dashboards. A comunidade desenvolvedora do Bootstrap é enorme e muito ativa, o que garante ao framework uma documentação completa e sempre atualizada de acordo com novas atualizações.

O Bootstrap usa um sistema de grid baseado em doze colunas, o que significa que qualquer layout pode ser dividido em até doze partes iguais, proporcionando flexibilidade para o design. Os desenvolvedores podem distribuir esse conteúdo em uma ou várias colunas para criar diferentes tipos layouts, sejam eles complexos ou simples, dependendo das necessidades básicas do projeto inicial. Algo que também facilita e agiliza o desenvolvimento são os breakpoints (pontos de quebra), que são pontos que se adaptam de acordo com o tamanho da página web, o que varia de acordo com cada dispositivo. Existem cinco tipos de breakpoints, que correspondem aos seguintes tamanhos de tela:

1. Extra small (xs): Utilizado para telas menores que 576px;
2. Small (sm): Utilizado para telas maiores ou iguais a 576px;
3. Medium (md): Utilizado para telas maiores ou iguais a 768px;
4. Large (lg): Utilizado para telas maiores ou iguais a 992px;
5. Extra large (xl): Utilizado para telas maiores ou iguais a 1200px.

Figura 11 - Ilustração de como os breakpoints são determinados pelo tipo de dispositivo



Fonte: <https://pbs.bartificer.net/pbs58> (Programming by Stealth)

Cada breakpoint define o comportamento da grid em diferentes larguras de tela, por exemplo, pode se especificar quantas colunas devem ser ocupadas em uma determinada largura de tela para garantir que o layout permaneça legível e agradável visualmente em dispositivos até o tamanho determinado pelo desenvolvedor.

Um ponto a ser observado pelos desenvolvedores é o aumento causado nas requisições de carregamento da página, o que depende da quantidade de recursos adicionados na página, ora que sua modularização pode exigir alguns conhecimentos mais avançados.

4.4.4. JAVACRIPT

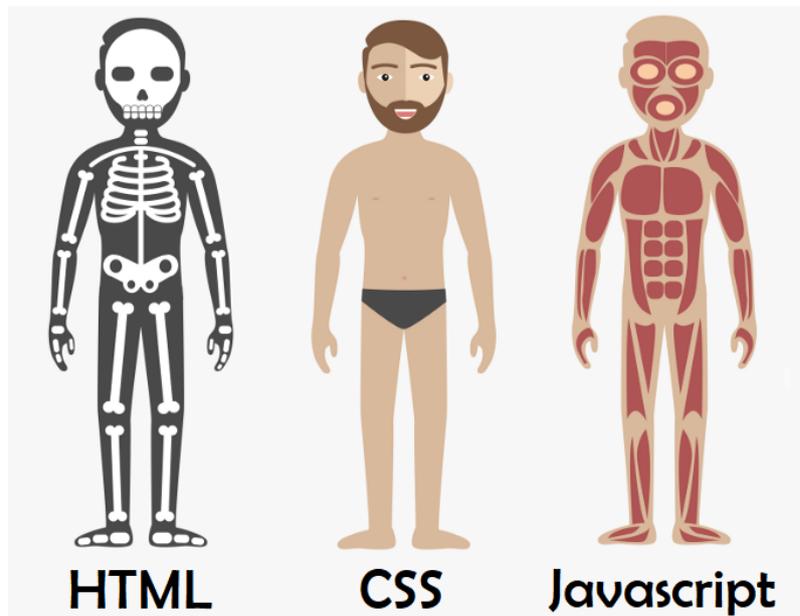
O JavaScript desempenha ajuda a proporcionar interatividade e dinamismo às aplicações web. Além disso, oferece diversas ferramentas e recursos que podem contribuir para a criação de experiências de usuário mais fluidas e adaptáveis a diferentes dispositivos.

Existem inúmeros frameworks e bibliotecas Javascript disponíveis atualmente que facilitam o desenvolvimento de funcionalidades interativas e responsivas, como por exemplo, o React e Angular, cujo são bibliotecas populares para a construção de interfaces de usuário mais flexíveis, essas ferramentas permitem a criação de componentes reutilizáveis e o gerenciamento eficiente do estado da aplicação.

O Javascript também pode ser utilizado para manipular o DOM (Document Object Model), ajudando os desenvolvedores a atualizarem dinamicamente o conteúdo e estilo das páginas com base nas interações do usuário e nas características do dispositivo que está sendo utilizado, por exemplo, eventos de redimensionamento da janela podem ser manipulados para ajustar dinamicamente o layout e o conteúdo da página conforme o tamanho da tela muda.

Além disso, uma característica muito presente e utilizada pelos desenvolvedores é a criação de animações, o que proporciona uma experiência de usuário mais agradável, também é possível adicionar comportamentos interativos, como menus de navegação responsivos, carrosséis de imagens e efeitos de transição, tornando o site mais dinâmico e atraente para os usuários.

Figura 12 - Representação de como cada ferramenta tem a sua própria função



Fonte: <https://www.linkedin.com/pulse/falando-de-css-html-e-javascript-responsividade-paulo-marinato/> (Paulo Marinato - LinkedIn)

4.4.5. W3C MARKUP VALIDATION SERVICE

O W3C Markup Validation Service (Serviço de Validação de Marcação) é uma ferramenta criada pela W3C de validação de código HTML e XHTML, que visa garantir que um site na web esteja em conformidade com os padrões e diretrizes estabelecidos pela própria

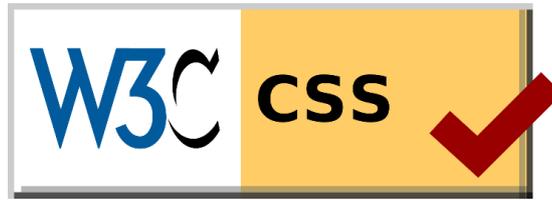
organização. Esta ferramenta analisa o código HTML, CSS e JavaScript de um site web fornecido pelo desenvolvedor em busca de erros de sintaxe, conformidade e de acessibilidade também, identificando se o código desenvolvido está de acordo com os padrões estabelecidos para a web.

Um dos primeiros conceitos criados sobre ferramentas de validação de código de marcação, foi um projeto desenvolvido em 1994 por Dan Connoly e Mark Gaither, desenvolvedores estadunidenses, onde a proposta era validar documentos de marcação, e fazer uma análise de possíveis erros. Este projeto foi aprimorado posteriormente no mesmo ano por Gerald Oskoboiny, desenvolvedor canadense, com uma ferramenta chamada: The Kinder, Gentler HTML Validator, propondo aprimorar o validador prévio criado pelos outros dois desenvolvedores.

No ano de 1997, Oskoboiny começou a trabalhar pela W3C, logo após foi anunciado a W3C Markup Validation Service baseado em seus trabalhos prévios, em 2008 após algumas atualizações de versão, foi possibilitado que documentos pudessem ser lidos pela ferramenta, e não apenas linhas de código, algumas das principais funções da ferramenta incluem:

1. **Análise de Erros:** Identificando e relatando erros de sintaxe no código HTML, CSS e Javascript de um site. Corrigir estes erros ajuda a garantir a compatibilidade com diferentes navegadores e dispositivos, além de melhorar a usabilidade da aplicação web;
2. **Otimização para SEO:** Além de verificar erros de sintaxe, a ferramenta de validação também ajuda na otimização para mecanismos de busca (SEO). Um código limpo e bem estruturado não apenas melhora a experiência do usuário com o site, mas também torna mais fácil para que os mecanismos de busca consigam indexar e classificar o site nos resultados de pesquisa. Esta abordagem terá uma análise mais aprofundada nos tópicos posteriores;
3. **Compatibilidade com Dispositivos:** Ao garantir que um site seja validado por este serviço de validação da W3C, os desenvolvedores podem ter mais confiança de que seu site será exibido corretamente em uma variedade de dispositivos e navegadores. Isso é essencial para a responsividade, pois um código válido e sem erros tem menos probabilidade de causar problemas de renderização e qualidade em diferentes plataformas e telas.

Figura 13 - Selo W3C certificando que o site foi verificado quanto à sintaxe CSS



Fonte: De autoria própria

O selo da W3C certificando que a aplicação web foi verificado quanto à sintaxe e uso semântico do código, garante mais credibilidade e segurança aos usuários, que por sua vez entendem que o site que está acessando é seguro e confiável.

4.4.6. FIGMA E ADOBE XD

Ferramentas como o Figma e o Adobe XD são amplamente utilizadas para o desenvolvimento de protótipos de tela, estas ferramentas oferecem recursos avançados para a prototipação de novos designs, além de proporcionar à prototipagem uma maior interatividade e colaboração em equipe, permitindo que os desenvolvedores criem layouts responsivos e visualizem como eles se adaptam a diferentes tamanhos de tela. Algumas maneiras pelas quais essas ferramentas facilitam a criação de designs responsivos estão:

1. Visualização em tempo real: Ambas as plataformas permitem que os desenvolvedores visualizem seus protótipos em tempo real em diferentes dispositivos e tamanhos de tela junto de sua equipe de desenvolvimento, isso permite que todos que estão colaborando no mesmo projeto vejam como o layout se ajustará em smartphones, tablets e desktops, facilitando a identificação de possíveis problemas de layout e ajustes que podem ser necessários posteriormente;
2. Colaboração em equipe: Tanto o Figma quanto o Adobe XD oferecem recursos de colaboração em equipe, permitindo que designers, desenvolvedores e outras partes interessadas no projeto trabalhem juntos no design. Isso é essencial para garantir que todos estejam trilhando o mesmo caminho proposto dentro do escopo inicial e que o design atenda às necessidades dos usuários finais;

3. Layouts flexíveis: Ambas ferramentas oferecem recursos de layout flexível, o que facilita a criação de designs responsivos. Breakpoints podem ser definidos visando adaptar o layout conforme a necessidade de cada tela.

Figura 14 - Logo das Ferramentas: Adobe XD e Figma



Fonte: <https://bootcamp.uxdesign.cc/have-i-finally-switched-to-figma-52c33e484eed>
(Sebiomo Aanuoluwapo - Medium)

Antes da criação de ferramentas específicas para prototipagem, ou seja ferramentas dedicadas a criação de designs e protótipos voltados ao UX e UI Design (User Experience e User Interface), ferramentas como Adobe Photoshop e Illustrator eram usadas para este fim, o que não é o propósito dessas ferramentas, até 2010 surgir o Sketch, primeira ferramenta totalmente dedicada a criação de protótipos, e após algum tempo vieram surgindo novas ferramentas, como o Adobe XD, criado em março de 2016 e o Figma, criado em setembro de 2016.

Ambas as ferramentas são similares e propósitos iguais, mas ainda sim possuem particularidades que a tornam diferentes e únicas uma da outra, como por exemplo: o Figma é muito conhecido por ser mais acessível e flexível ao mercado, até por permitir o seu uso por meio de navegadores web, o que já não acontece para a ferramenta da Adobe, que além de ter uma licença paga para liberar uma versão completa ao usuário, é focado mais na criação de designs a partir de componentes já existentes, o que contrasta ao Figma, quando o mesmo já é mais dependente da criatividade do desenvolvedor, mas ainda sim possui diversas bibliotecas e plugins com componentes já prontos, como botões, imagens e gráficos. Com isso o Figma tende a ser mais escolhido por desenvolvedores iniciantes na prototipagem, principalmente por sua acessibilidade referente ao uso e custo.

A principal forma que esse tipo de ferramenta pode auxiliar no desenvolvimento de uma aplicação web responsiva é através de testes, ou seja, antes do desenvolvimento começar, é realizado a prototipagem, que se dedica a criar uma réplica de como a aplicação web funcionará quando já estiver no ar, através disso tanto o desenvolvedor como o usuário conseguem ter uma noção de como o conteúdo deverá ser visualizado para diferentes tipos de telas, o que cabe a responsabilidade ao desenvolvedor cuidar da prototipagem dessas telas para todos os tipos de tela, o que inclui smartphones, tablets, notebooks, e até mesmo televisões. É crucial ter esse cuidado na prototipagem pois é o primeiro passo antes de todo o desenvolvimento, e caso esse pilar não estiver fortalecido, todo o projeto terá problemas em sua continuação, é exatamente o momento de testar novas possibilidades, além de medir os padrões de comportamento da aplicação, como definir breakpoints.

Dessa forma é facilitado o desenvolvimento da aplicação tanto para a web como para mobile, sempre atendendo às necessidades do usuário que terá o principal contato com o sistema a ser desenvolvido.

5. SEO E RESPONSIVIDADE

Nos tópicos a seguir será realizada uma análise aprofundada referente a relação entre o tema de Responsividade e SEO, como o conceito de SEO pode afetar tanto positivamente como negativamente o desenvolvimento de aplicações responsivas, como os desenvolvedores podem aplicar seu conceito para facilitar e apoiar o seu desenvolvimento, e práticas que podem ser adotadas e evitadas.

5.1. DEFINIÇÃO DE SEO

A responsividade de um site refere-se à sua capacidade de se adaptar a diferentes dispositivos e tamanhos de tela, garantindo uma experiência de usuário consistente e otimizada, independentemente do dispositivo que está sendo usado para acessá-lo como já foi discutido em tópicos anteriores. Por outro lado, o SEO (Search Engine Optimization) trata da otimização de um site visando garantir uma classificação mais alta nos resultados dos mecanismos de busca em navegadores digitais, como: Google Chrome e Mozilla Firefox.

Existem diversos impactos positivos e negativos que podem ser encontrados ao integrar as práticas de SEO com o desenvolvimento responsivo, alguns dos principais pontos que tendem a favorecer a aplicação responsiva são:

1. **Melhor Classificação em Mecanismos de Busca:** O Google e outros mecanismos de busca favorecem sites responsivos em suas classificações. Ao otimizar um site para diferentes dispositivos, os desenvolvedores aumentam suas chances de obter uma classificação mais alta nos resultados de pesquisa;
2. **Maior Taxa de Retenção de Usuários:** Sites responsivos tendem a ter uma taxa de rejeição menor e uma taxa de retenção maior dos usuários, o que é um fator importante para o SEO, uma vez que os usuários permanecem por mais tempo em um site e interagem mais com seu conteúdo em uma aplicação responsiva, isso sinaliza para os mecanismos de busca que o site oferece uma experiência mais valiosa aos usuários do que diferentes sites;
3. **Compatibilidade para Mobile:** Com o aumento do uso de dispositivos móveis para acessar a web, a compatibilidade móvel é crucial. Sites responsivos oferecem uma experiência de usuário consistente em dispositivos móveis, o que pode resultar em um aumento do tráfego proveniente de dispositivos móveis, sendo outro fator positivo para o SEO.

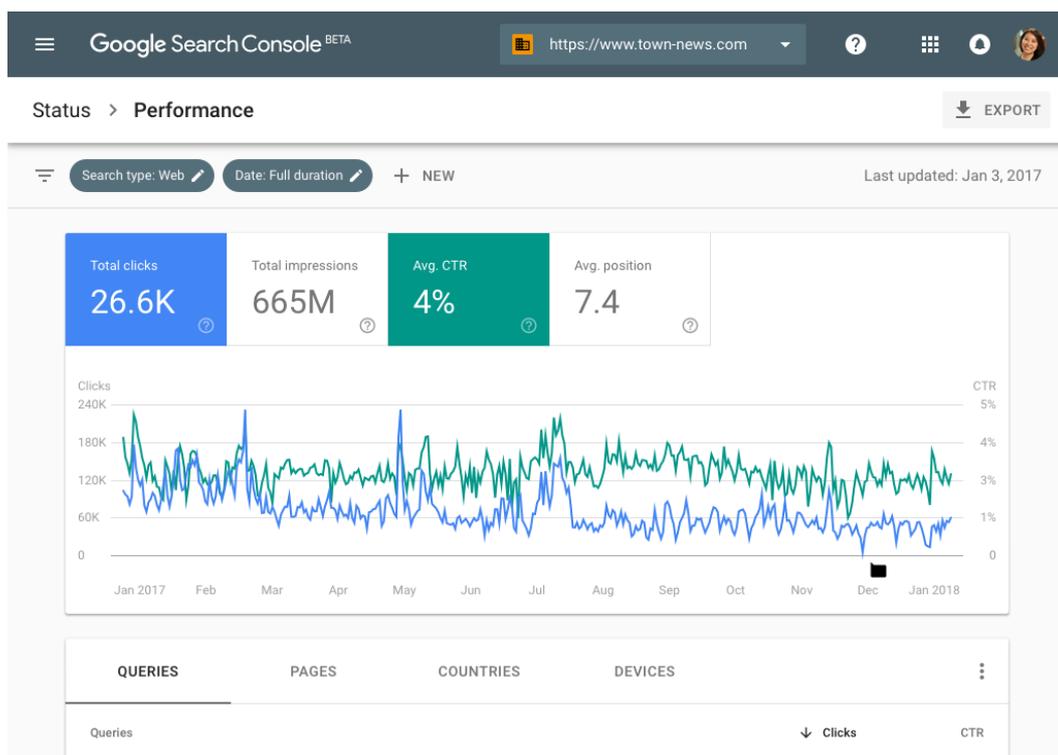
Em contraste a alguns pontos positivos em relação à aplicação de SEO para o desenvolvimento, existem impactos negativos que podem possivelmente atrapalhar o desenvolvimento responsivo, e entre alguns estão:

1. **Desempenho:** Às vezes, as práticas de SEO podem exigir a inclusão de elementos que podem prejudicar o desempenho de um site web, como scripts adicionais para rastreamento ou tags de acompanhamento. Se esses elementos não forem implementados corretamente, podem afetar negativamente a velocidade de carregamento e a experiência do usuário, o que pode impactar negativamente a classificação nos mecanismos de busca;
2. **Complexidade no Desenvolvimento:** A otimização para SEO muitas vezes envolve a inclusão de conteúdo adicional, como meta tags, descrições e tags de título, que devem ser adaptadas para diferentes dispositivos. Isso pode adicionar complexidade ao processo de desenvolvimento da aplicação e requer que a equipe de desenvolvimento seja cuidadosa em garantir que o conteúdo seja exibido corretamente em todos os dispositivos.

Uma ferramenta de análise de SEO é o Google Search Console, uma plataforma versátil que ajuda a obter diagnósticos completos sobre qualquer tipo de página que seu usuário ou desenvolvedor deseja analisar, porém com mecanismos de busca mais complexos do que os algoritmos de navegadores padrões.

Esse tipo de ferramenta ajuda a otimizar dinheiro e tempo para o time de desenvolvimento, além de poder ser usada em conjunto também com o Google Analytics, ferramenta utilizada para monitorar métricas em tempo real, como acessos à página, taxa de rejeição e aceitação ao conteúdo e o tempo de permanência do usuário no site, o que apoia a observação do desempenho e posicionamento em mecanismos de busca.

Figura 15 - Página inicial do Google Search Console baseado na análise de uma página na web



Fonte: <https://c2ti.com.br/blog/8-ferramentas-de-seo-gratuitas-para-analisar-e-otimizar-seu-site> (Samara Côra - C2TI)

5.2. BOAS PRÁTICAS PARA SEO (SEARCH ENGINE OPTIMIZATION)

Algumas práticas também devem ser consideradas pelos desenvolvedores ao integrar SEO em suas aplicações, algumas boas práticas de SEO com o desenvolvimento de sites web são:

1. **Priorizar a Experiência do Usuário:** Tanto o SEO quanto a responsividade devem ter como foco principal a experiência do usuário. Ao projetar e desenvolver um site, é importante considerar como cada elemento contribui para a experiência de usuário de forma positiva para todos os tipos de dispositivos;
2. **Testagem e Otimização Regular:** Realizar testes de desempenho e usabilidade em diferentes dispositivos é essencial para garantir que o site seja responsivo e ofereça uma experiência consistente em todas as plataformas. Além de monitorar regularmente as métricas de SEO para identificar áreas de melhoria e ajustar a estratégia inicial conforme necessário.

Práticas a serem evitadas também devem ser consideradas, visto que podem comprometer significativamente a eficácia tanto do SEO aplicado, quanto da responsividade do site. Ignorar ou negligenciar qualquer uma das duas pode resultar em uma experiência de usuário insatisfatória, uma classificação mais baixa nos mecanismos de busca e, em última análise, impactar negativamente o sucesso do site. A seguir está uma lista de algumas práticas a serem evitadas e possíveis alternativas para contornar estes problemas:

1. **Ignorar a Responsividade em Favor do SEO:** É crucial reconhecer que a responsividade é uma parte fundamental da otimização para mecanismos de busca. Priorizar exclusivamente o SEO em detrimento da responsividade pode levar a um site que não oferece uma experiência de usuário consistente em todos os dispositivos. Isso pode resultar em taxas de rejeição mais altas, menor tempo de permanência no site e, conseqüentemente, uma classificação mais baixa nos resultados de pesquisa.

Alternativa Proposta: Priorizar a abordagem "Mobile-First" no design e desenvolvimento do site. Isto significa começar pelo design e desenvolvimento para dispositivos móveis e, em seguida, expandir para dispositivos maiores. Ao adotar essa abordagem, se garante que o site seja responsivo desde o início. Além de utilizar ferramentas de testagem de compatibilidade móvel para verificar se a aplicação web é exibida corretamente em uma variedade de dispositivos e tamanhos de tela;

2. **Sobrecarregar o Site com Elementos de SEO Desnecessários:** Embora seja importante incluir elementos de SEO como meta tags, descrições e tags de título, é igualmente importante evitar sobrecarregar o site com esses elementos. Adicionar excesso de palavras-chave, tags ocultas ou texto irrelevante pode ser interpretado como uma tentativa de manipular os mecanismos de busca e resultar em penalidades, prejudicando a classificação do site.

Alternativa Proposta: A qualidade deve ser ponto focal ao invés da quantidade na inclusão de elementos de SEO. Ao invés de utilizar desnecessariamente palavras-chave, meta tags excessivas ou texto irrelevante, o desenvolvedor pode se concentrar em criar conteúdo valioso e relevante para os usuários, utilizando palavras-chave de forma natural e estratégica, mantendo o foco na experiência do usuário;

3. **Desempenho Comprometido devido a Excesso de Recursos de SEO:** Alguns recursos de SEO, como scripts adicionais para rastreamento ou tags de acompanhamento, podem comprometer o desempenho do site, especialmente em dispositivos móveis. Se esses recursos não forem implementados corretamente, podem aumentar o tempo de carregamento da página, afetando negativamente a experiência do usuário e a classificação nos mecanismos de busca.

Alternativa Proposta: Otimizar o carregamento de recursos de SEO, como scripts de rastreamento e tags de acompanhamento, através da minimização do uso de scripts desnecessários, optando por soluções de terceiros que ofereçam desempenho otimizado. Utilizar técnicas de carregamento assíncrono para evitar atrasos no carregamento da página também pode ser implementado de acordo com a necessidade do usuário e escopo inicial do projeto;

4. **Falta de Atualização e Manutenção:** A falta de atualização e manutenção do site pode levar a problemas de compatibilidade e desempenho, prejudicando tanto o SEO quanto a responsividade. É importante manter-se atualizado com as melhores práticas de SEO e tecnologias de desenvolvimento web, e realizar atualizações regulares para garantir que o site permaneça responsivo e otimizado para os mecanismos de busca.

Alternativa Proposta: Implementar um plano de atualização e manutenção regular para a aplicação, o que inclui a verificação periódica de compatibilidade com novas tecnologias e diretrizes para a web, bem como a aplicação de atualizações de segurança e correções de bugs;

Ao evitar essas práticas prejudiciais, os desenvolvedores podem garantir que seus sites ofereçam uma experiência de usuário consistente e de alta qualidade em todos os dispositivos, ao mesmo tempo em que alcançam uma classificação mais alta nos mecanismos de busca.

6. EVOLUÇÃO DAS DIRETRIZES DE RESPONSABILIDADE

A responsividade para a web e dispositivos móveis é um componente essencial para o desenvolvimento responsivo a partir da criação de experiências de usuário eficientes e acessíveis, adaptando o conteúdo da aplicação desenvolvida a diferentes tamanhos de tela e capacidades dos dispositivos.

No início da web, as páginas eram estáticas e projetadas principalmente para monitores de computador. Com o crescimento da diversidade de dispositivos, como smartphones e tablets, tornou-se evidente a necessidade de criar layouts que pudessem se adaptar a diferentes tamanhos e resoluções de tela, dando partida para a evolução das diretrizes de responsividade, cujo começou a atender essas demandas, oferecendo soluções técnicas para criar experiências de usuário consistentes e acessíveis em uma variedade de dispositivos, com isso foram surgindo organizações com o fim de padronizar e criar diretrizes para o desenvolvimento de novas aplicações, aprimorando o conceito de design responsivo evoluiu significativamente desde os primórdios da web, o que foi muito impulsionado por organizações como o W3C e o WCAG.

6.1. W3C (WORLD WIDE WEB CONSORTIUM)

A W3C foi uma das primeiras organizações a reconhecer a necessidade de diretrizes que suportassem a criação de websites responsivos. Desde a sua fundação em 1994, a W3C tem desempenhado um papel crucial na definição de padronização para a web que garantem a interoperabilidade, responsividade e acessibilidade para a aplicação. Com isso foi criado com o tempo diversas práticas e diretrizes que corroboraram para a evolução da comunidade programadora, algumas das criações mais famosas foram:

1. Primeiros Padrões HTML e CSS: As primeiras versões permitiam apenas layouts estáticos, sem suporte significativo para responsividade;

2. HTML4 e CSS2 (1998): Com a introdução do HTML4 e CSS2, os desenvolvedores ganharam mais controle sobre a apresentação do conteúdo. O CSS2 introduziu conceitos como floats e posicionamento, que permitiam layouts mais flexíveis, mas ainda não totalmente responsivos;
3. HTML5 (2014): HTML5 introduziu novas tags semânticas, como `<header>`, `<footer>`, `<article>`, e `<section>`, que melhoraram a estrutura e a acessibilidade das páginas web. Também trouxe APIs para gráficos, áudio e vídeo que suportaram interatividade avançada, essencial para aplicações web modernas;
4. CSS3 (2011): A introdução desta nova versão do CSS possibilitou a criação de diversas ferramentas como: Media Queries, Flexbox, Grids, Tags de Viewport e entre muitas outras que também já foram analisadas anteriormente.

6.2. WCAG

As Web Content Accessibility Guidelines (WCAG) foram criadas para garantir que o conteúdo web seja acessível a pessoas com deficiências. Desde a sua primeira versão, as WCAG têm evoluído para incluir uma gama mais ampla de tecnologias e necessidades dos usuários. A primeira versão da WCAG, a 1.0, criada em 1999, focava em tornar o conteúdo web acessível através de diretrizes específicas e técnicas recomendadas. As principais áreas de foco incluíam:

1. Alternativas Textuais: Fornecer texto alternativo para imagens, gráficos e outros conteúdos não textuais, como por exemplo a propriedade `alt` dentro do HTML;
2. Estrutura de Conteúdo: Utilizar elementos estruturais HTML (como `<h1>`, `<h2>`, etc.) para organizar o conteúdo de maneira lógica.

A versão 2.0 da WCAG lançada em 2008 introduziu um modelo mais robusto baseado em princípios, diretrizes e critérios de sucesso, estes princípios fundamentais (POUR) incluem:

1. Perceptibilidade: O conteúdo deve ser apresentado de forma que possa ser percebido pelos usuários. Exemplos:
 - a) Texto Alternativo: Continuar a fornecer alternativas textuais para conteúdo não textual;
 - b) Legenda e Transcrições: Fornecer legendas para vídeos e transcrições para conteúdo de áudio.

2. Operabilidade: A interface e a navegação devem ser operáveis. Exemplos:
 - a) Acessibilidade via Teclado: Garantir que todas as funcionalidades sejam acessíveis via teclado;
 - b) Tempo Suficiente: Fornecer tempo suficiente para que os usuários possam ler e usar o conteúdo.
3. Compreensibilidade: A informação e a operação da interface devem ser compreensíveis. Exemplos:
 - a) Leitura e Previsibilidade: Tornar o texto legível e previsível;
 - b) Ajuda com Erros: Ajudar os usuários a evitarem e corrigir erros.
4. Robustez: O conteúdo deve ser robusto o suficiente para ser interpretado por uma ampla variedade de agentes de usuário, incluindo tecnologias assistivas.

A versão mais atual da WCGA 2.1, lançada em 2018, ajudou a expandir os critérios de sucesso para cobrir melhor a acessibilidade em dispositivos móveis e para pessoas com deficiências cognitivas e de baixa visão, alguns exemplos incluem:

1. Modalidades de Input: Melhor suporte para interfaces sensíveis ao toque, como assegurar que alvos de toque sejam suficientemente grandes e espaçados;
2. Contraste: Critérios mais rigorosos para contraste de cores, garantindo melhor legibilidade em diferentes condições de iluminação;
3. Espaçamento de Texto: Diretrizes para assegurar que o espaçamento de texto possa ser ajustado para melhorar a legibilidade.

O objetivo da nova versão WCAG 2.2 é dar continuidade às versões anteriores, melhorando as recomendações de acessibilidade para três principais grupos: pessoas com deficiências cognitivas ou dificuldade de aprendizagem, com baixa visão e com deficiências em dispositivos móveis. Os sites que seguem WCAG 2.0 e 2.1 continuarão estando de acordo com as novas diretrizes. É importante ressaltar que a WCAG possui níveis de conformidade que representam os desafios a serem ultrapassados, entre eles estão:

1. Nível A: Critérios básicos que representam as barreiras mais significativas de acessibilidade;
2. Nível AA: Garante acesso à grande maioria dos conteúdos;
3. Nível AAA: Especificações mais detalhadas e sofisticadas para um nível mais elevado de acessibilidade.

Alguns novos critérios de sucesso no WCAG 2.2 foram projetados, com o intuito de melhorar a experiência do usuário, fornecendo diretrizes mais detalhadas e específicas. Eles são organizados em três níveis de conformidade (A, AA e AAA) como foi explicado anteriormente, ou seja, tem o objetivo de atender às diversas necessidades dos usuários e facilitar a implementação prática dessas recomendações por parte dos desenvolvedores.

A seguir, são apresentados os novos critérios de sucesso do WCAG 2.2, com uma breve explicação sobre o que cada um deles busca alcançar e como contribuem para uma web mais inclusiva e acessível:

1. Autenticação Acessível (Nível A): Alternativas de autenticação mais acessíveis para pessoas com deficiências cognitivas, facilitando o processo de login sem a necessidade de memorização de senhas complexas;
2. Arrastando (Nível AA): Interações que envolvem arrastar o ponteiro na tela devem ser substituídas por ações de toque único ou cliques, exceto quando o movimento de arrastar for essencial. Isso melhora a usabilidade para pessoas com mobilidade reduzida;
3. Ajuda Localizável (Nível A): Disponibilização de formas de ajuda para localizar informações ou executar tarefas no site, como mecanismos de contato automatizado ou opções de autoajuda;
4. Pontos de Referência Fixos (Nível A): Facilita a localização de referências de conteúdo dentro das páginas, mantendo a informação acessível independentemente do dispositivo ou formatação;
5. Foco Visível Mínimo (Nível AA): Especificações para a visibilidade do foco durante a navegação sem mouse, com tamanho, contraste e espessura da borda adequados;

6. Foco Visível Aprimorado (Nível AAA): Diretrizes para um foco ainda mais visível, ideal para pessoas com deficiências visuais e motoras, melhorando o contraste e o tamanho dos elementos focados;
7. Controles Escondidos (Nível AA): Controles necessários para a navegação devem estar totalmente visíveis, em vez de aparecerem apenas quando a função de foco no teclado é ativada;
8. Espaçamento de Alvo do Ponteiro (Nível AA): Elementos interativos devem ter um tamanho mínimo de 44 pixels de altura e largura, facilitando o uso para pessoas com deficiências motoras e evitando cliques acidentais;
9. Entrada Redundante (Nível A): Redução da solicitação repetida de dados em formulários, favorecendo a usabilidade para pessoas com deficiências cognitivas, ao reutilizar informações previamente fornecidas, exceto quando necessário por questões de segurança.

Esses novos critérios refletem o compromisso do WCAG em criar uma web mais acessível e amigável, abordando de forma abrangente e detalhada as barreiras enfrentadas por diversos usuários.

A implementação dessas diretrizes não só melhora a acessibilidade, mas também promove uma melhor experiência geral para todos os usuários, independentemente de suas habilidades ou dispositivos utilizados, além de aprimorar as diretrizes anteriores, garantindo uma web mais inclusiva e acessível para uma gama ainda maior de usuários, mantendo a compatibilidade com as versões WCAG 2.0 e 2.1.

6.3. OUTRAS DIRETRIZES E PADRÕES INTERNACIONAIS

Além de diretrizes propostas por organizações como a W3C e a WCAG, várias outras organizações e padrões internacionais contribuem para a evolução da responsividade para a web e em dispositivos móveis também. A seguir, será apresentado e detalhado alguns exemplos significativos:

1. ISO/IEC 40500: A ISO/IEC é um padrão internacional adotada para várias práticas, não só para o desenvolvimento de aplicações, em 2012, a WCAG 2.0 foi adotada como um padrão internacional sob a designação ISO/IEC 40500, esta padronização fortaleceu seu

reconhecimento global, incentivando a implementação de práticas acessíveis e responsivas em todo o mundo, o que resultou em uma aplicação dessas práticas de forma mais consistente em diferentes países e contextos;

2. Apple Human Interface Guidelines (HIG): Essas diretrizes foram criadas pela Apple em 1984 junto ao lançamento do primeiro Macintosh, desde então tem sido fornecido um conjunto abrangente de diretrizes que enfatizam a importância da responsividade e acessibilidade para aplicativos com sistema operacional iOS, nativos do Iphone. Essas diretrizes incluem recomendações para criar layouts flexíveis, utilizar tipografia de texto adequada e garantir que os elementos de toque sejam de tamanho adequado;
3. Google Material Design: O Material Design do Google foi criado pela própria Google em 2014 como parte de uma ferramenta extra para designers que utilizavam seus produtos, o Material Design é um sistema de design que visa criar experiências de usuário coesas em diferentes dispositivos e plataformas, incluindo orientações detalhadas sobre como criar layouts responsivos que se adaptam automaticamente a diferentes tamanhos de tela e resoluções. Além disso, o Material Design ressalta a importância da acessibilidade, assegurando que todos os usuários, independentemente de suas habilidades, possam interagir com os aplicativos de forma eficaz e agradável.

Esses exemplos destacam como diferentes organizações e padrões internacionais trabalham juntos para promover a responsividade e acessibilidade na web e em dispositivos móveis, garantindo uma experiência de usuário uniforme e inclusiva em todas as plataformas.

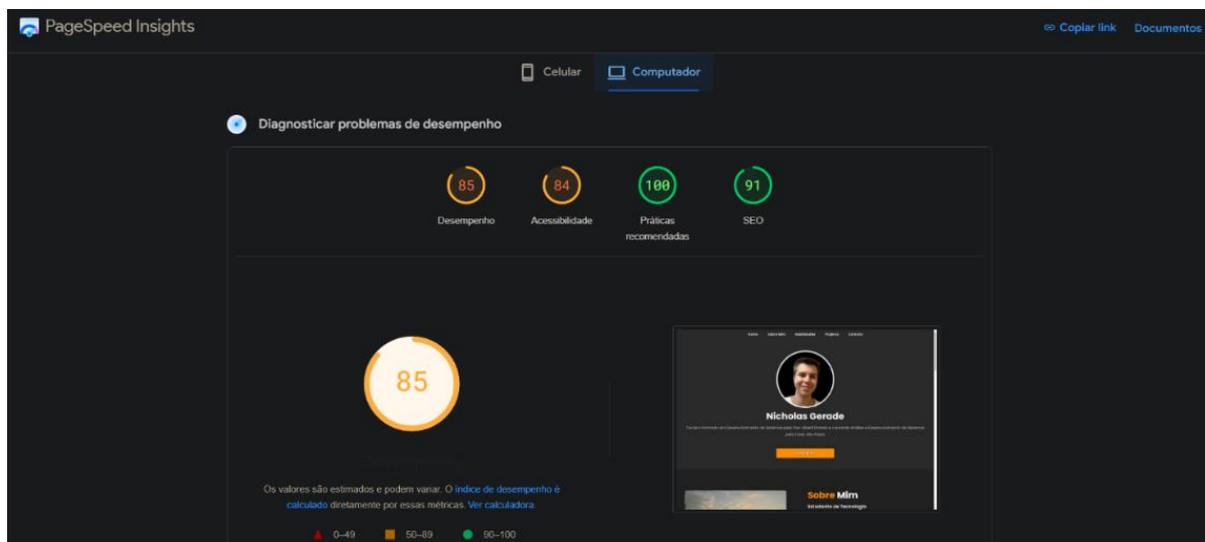
7. CASO DE ESTUDO

Neste tópico será realizado um estudo de caso em um site de portfólio pessoal com base nas técnicas e ferramentas analisadas no presente trabalho, utilizando a ferramenta Google Lighthouse. Esta ferramenta é usada para ajudar desenvolvedores a melhorar a qualidade de suas páginas web e pode ser executada como uma extensão para navegadores. O Google Lighthouse fornece uma série de auditorias em diferentes aspectos de uma página web, tais como análise de SEO, acessibilidade, desempenho e práticas recomendadas. Com isso, algumas práticas analisadas no presente trabalho serão implementadas para melhorar as métricas impostas pelo Google Lighthouse.

É importante ressaltar que o site analisado é um projeto já finalizado e bem estabelecido, com características inerentes que, de certa forma, já contribuem naturalmente para as métricas do Google Lighthouse. No entanto, ciente da importância contínua de aprimorar a experiência do usuário e garantir um desempenho excepcional, uma série de mudanças avançadas serão planejadas e aplicadas meticulosamente. Essas alterações serão concebidas não apenas de forma complementar, mas também para impulsionar as propriedades já bem estruturadas do site. Assim, algumas medidas mais avançadas serão cuidadosamente projetadas para se integrarem harmoniosamente às características existentes, resultando em uma harmonia maior dentro da aplicação e ajudando a elevar a experiência do usuário a novos patamares de excelência.

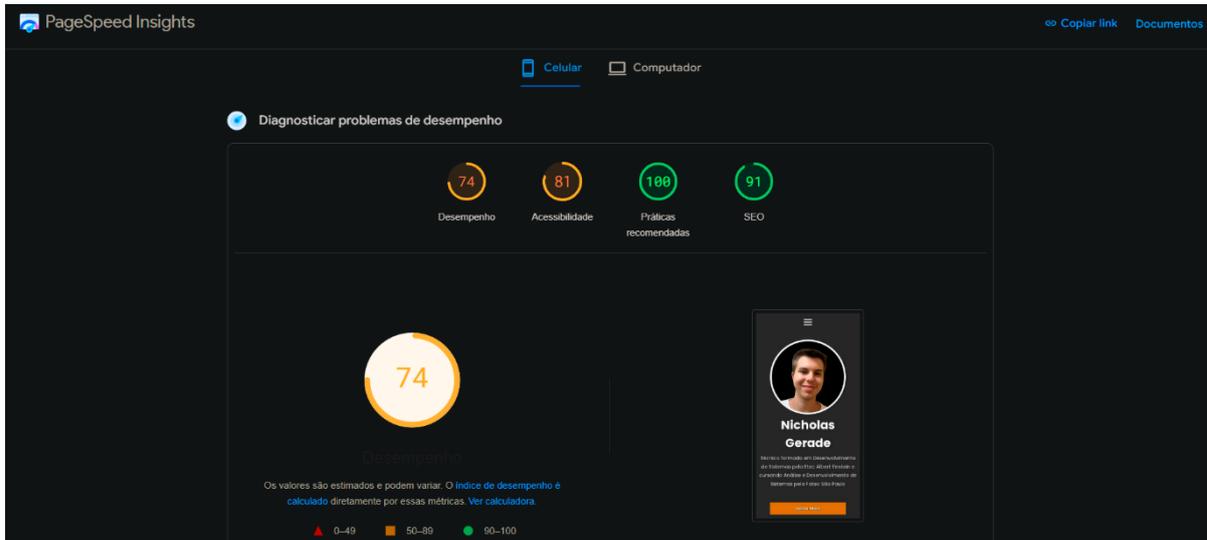
Abaixo são apresentadas as métricas para computadores e, logo em seguida, as métricas para celular do site escolhido antes da aplicação das práticas que serão adotadas. No tópico seguinte, será detalhado todas as práticas que foram aplicadas.

Figura 16 - Métricas para Computador do site de Portfólio analisado pela ferramenta Google Lighthouse antes da aplicação das práticas estudadas.



Fonte: De autoria própria

Figura 17 - Métricas para Celular do site de Portfólio analisado pela ferramenta Google Lighthouse antes da aplicação das práticas estudadas.



Fonte: De autoria própria

7.1. TÉCNICAS APLICADAS

Para melhorar as métricas do site conforme as recomendações do Google Lighthouse, diversas mudanças significativas foram implementadas. Primeiramente, as imagens foram convertidas para formatos modernos como WebP e AVIF, resultando em uma economia substancial de dados e tempos de carregamento mais rápidos. Além disso, as dimensões de todas as imagens foram definidas explicitamente, o que ajudou a reduzir mudanças de layout inesperadas e, conseqüentemente, melhorou a métrica de Cumulative Layout Shift (CLS), o CLS mede a estabilidade visual de uma página, avaliando o quanto os elementos se movem inesperadamente durante o carregamento.

A adequação do tamanho das imagens para suas respectivas exibições também foi ajustada, otimizando ainda mais o uso de dados de rede. Recursos estáticos agora possuem políticas de cache eficientes, o que contribui para um desempenho superior em visitas subseqüentes. O CSS crítico foi incluído inline, visando acelerar a renderização inicial da página, enquanto scripts e estilos não críticos foram adiados, eliminando bloqueios de renderização.

Além disso, a propriedade font-display foi adicionada aos arquivos CSS de webfonts, assegurando que o texto permaneça visível enquanto as fontes carregam, melhorando a

experiência do usuário. Para completar, as imagens foram otimizadas para uma codificação mais eficiente, resultando em uma economia adicional de bytes.

Outras melhorias estratégicas também foram realizadas. Implementou-se a propriedade lazy loading para imagens que estão fora da visualização inicial, fazendo com que estas imagens só sejam carregadas quando o usuário descer a tela ao encontro destas imagens, o que ajudou a melhorar o Largest Contentful Paint (LCP) e o First Contentful Paint (FCP). O LCP mede o tempo até que o maior conteúdo visível na tela seja carregado, enquanto o FCP mede o tempo até que o primeiro elemento de conteúdo seja renderizado.

Também se otimizou o carregamento de CSS e JavaScript, minimizando e adiando recursos não críticos, o que melhorou significativamente o Total Blocking Time (TBT) e a métrica de First Input Delay (FID). O TBT mede o tempo durante o qual a página está bloqueada e o usuário não pode interagir com ela, enquanto o FID mede o tempo desde a primeira interação do usuário com a página até que o navegador responda a essa interação.

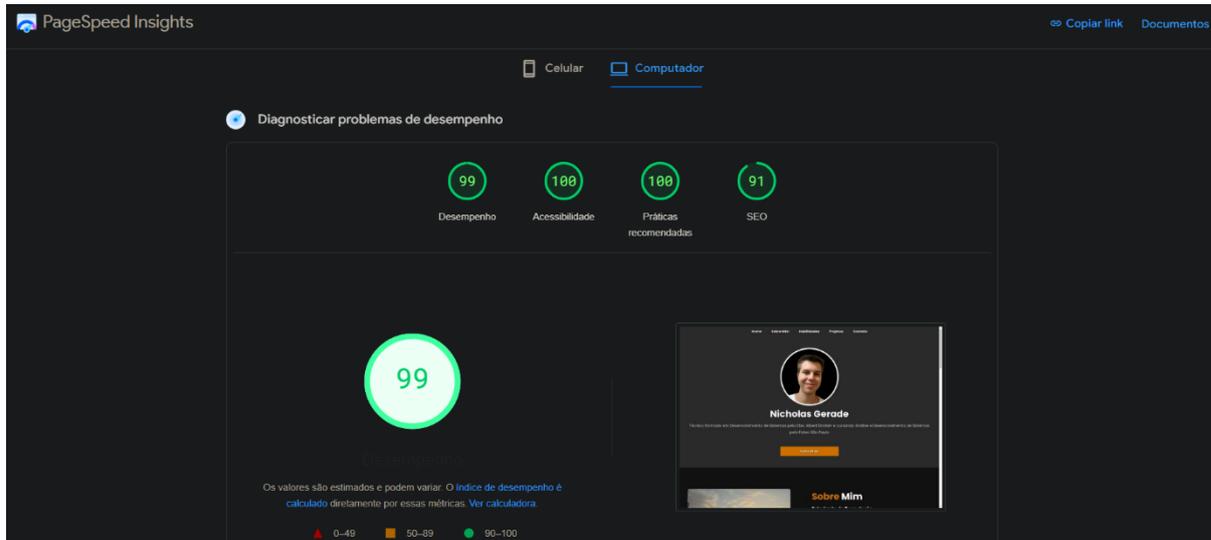
Adicionalmente, todos os recursos foram comprimidos com Gzip, minimizando o tamanho dos arquivos CSS e JavaScript, reduzindo o tempo de transferência de dados e melhorando o tempo de resposta do servidor. Essas ações abrangentes resultaram em melhorias notáveis nas pontuações de desempenho, acessibilidade, melhores práticas e SEO na plataforma Google Lighthouse. Os resultados dessas práticas aplicadas serão apresentados no próximo tópico.

7.2. RESULTADOS OBTIDOS PARA DESKTOP

Este tópico focará na implementação das práticas estudadas e otimizações exclusivamente para o ambiente de computadores. Os resultados da avaliação pelo Google Lighthouse revelaram um progresso notável em várias métricas-chave. Cada etapa da análise e otimização foi conduzida com um foco específico em aprimorar a performance e a acessibilidade.

Essa abordagem direcionada foi fundamental para assegurar que todas as melhorias implementadas contribuíssem significativamente para a experiência do usuário, resultando em um ambiente online mais eficiente, acessível e agradável para todos os visitantes, já que as métricas de Práticas Recomendadas e SEO já possuíam um nível aceitável.

Figura 18 - Métricas para Computador do site de Portfólio analisado pela ferramenta Google Lighthouse depois da aplicação das práticas estudadas.



Fonte: De autoria própria

Inicialmente, o desempenho do site viu uma melhoria significativa, com as métricas de Desempenho passando de 85 para 99. Essa mudança foi impulsionada por uma série de otimizações, incluindo a conversão de imagens para formatos mais eficientes, definição de dimensões explícitas e a implementação de políticas de cache eficientes, resultando em tempos de carregamento mais rápidos e uma experiência geral mais ágil para o usuário.

Além disso, a acessibilidade do site também apresentou melhorias notáveis, com a métrica subindo de 84 para 100. As práticas adotadas para garantir a navegabilidade por teclado, compatibilidade com leitores de tela e inclusão de alternativas textuais para conteúdos visuais contribuíram significativamente para tornar o site mais acessível e inclusivo para todos os usuários.

Em relação às práticas recomendadas, o site manteve uma pontuação perfeita de 100, demonstrando conformidade contínua com as melhores práticas de desenvolvimento web, especialmente quando acessado por meio de dispositivos desktop. Esses resultados refletem não apenas um compromisso contínuo com a qualidade técnica e relevância do conteúdo oferecido pelo site, mas também uma busca constante pela excelência em todos os aspectos do desenvolvimento web.

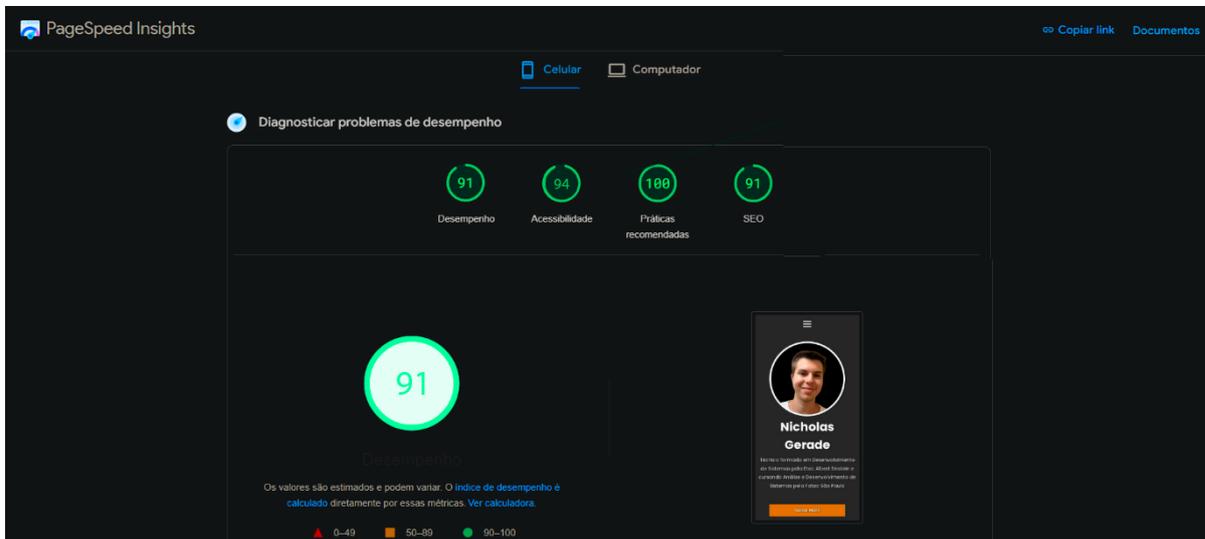
No que diz respeito ao SEO, o site também obteve uma pontuação consistente de 91, destacando uma otimização eficaz para mecanismos de busca e uma presença online robusta e bem classificada.

7.3. RESULTADOS OBTIDOS PARA DISPOSITIVOS MÓVEIS

Este tópico focará na implementação das práticas estudadas e otimizações exclusivamente para o ambiente de dispositivos móveis. É importante ressaltar que as práticas e códigos aplicados são idênticos às práticas e códigos aplicados para desktop.

Os resultados da avaliação para dispositivos móveis através do Google Lighthouse mostraram melhorias significativas em várias métricas-chave, demonstrando o impacto positivo das otimizações aplicadas tanto para desktop quanto para dispositivos móveis.

Figura 19 - Métricas para Celular do site de Portfólio analisado pela ferramenta Google Lighthouse depois da aplicação das práticas estudadas.



Fonte: De autoria própria

Inicialmente, o desempenho do site em dispositivos móveis melhorou significativamente, com a métrica de Desempenho subindo de 74 para 91, pontuação que estava ainda menor do que para computadores. Essa mudança foi impulsionada por uma série de otimizações, incluindo a conversão de imagens para formatos mais eficientes, definição

explícita das dimensões das imagens e a implementação de políticas de cache eficientes, resultando em tempos de carregamento mais rápidos e uma experiência geral mais ágil para o usuário, práticas que foram mais detalhadas no tópico anterior.

Além disso, a acessibilidade do site em dispositivos móveis também apresentou uma melhoria notável, com a métrica subindo de 81 para 94. As práticas adotadas para garantir a navegabilidade por teclado, compatibilidade com leitores de tela e inclusão de alternativas textuais para conteúdos visuais contribuíram significativamente para melhorar esta métrica.

Em relação às Práticas Recomendadas, o site manteve uma pontuação perfeita de 100, demonstrando conformidade contínua com as melhores práticas de desenvolvimento web, especialmente quando acessado por meio de dispositivos móveis. No que diz respeito ao SEO, o site manteve uma pontuação consistente de 91, destacando uma otimização eficaz para mecanismos de busca e uma presença online robusta e bem classificada. Esses resultados validam a aplicação de todas as práticas estudadas e analisadas tanto para celulares quanto para computadores.

7.4. RESULTADOS GERAIS

Durante o desenvolvimento deste estudo, uma série de técnicas e otimizações foram aplicadas ao site de portfólio analisado, com o objetivo de melhorar métricas como: performance, acessibilidade, práticas recomendadas e SEO. A análise detalhada das métricas pelo Google Lighthouse revelou áreas de oportunidade para aprimoramento, guiando como deveriam as práticas deveriam ser aplicadas para garantir uma experiência de usuário excepcional em todo tipo de plataforma.

Inicialmente, diversas mudanças significativas foram implementadas para melhorar o desempenho do site. Isso incluiu a conversão de imagens para formatos mais eficientes, definição explícita das dimensões das imagens e a implementação de políticas de cache eficientes. Essas alterações resultaram em tempos de carregamento mais rápidos e uma experiência geral mais ágil para os usuários. A acessibilidade do site também foi aprimorada, garantindo a navegabilidade por teclado, compatibilidade com leitores de tela e inclusão de alternativas textuais para conteúdos visuais.

Os resultados obtidos foram notáveis em todas as métricas avaliadas pelo Google Lighthouse. Tanto para dispositivos desktop quanto para dispositivos móveis, as pontuações de desempenho, acessibilidade, práticas recomendadas e SEO melhoraram significativamente.

Todas essas melhorias não apenas garantiram uma experiência de usuário superior à média, mas também ajudam a posicionar o site de uma forma mais competitiva nos resultados de busca em relação a outras aplicações. Em conclusão, a aplicação das novas técnicas e otimizações mostrou-se altamente eficaz na melhoria do site de portfólio analisado. Este estudo ressalta a importância da melhoria contínua no desenvolvimento web e destaca a necessidade de uma abordagem integrada e focada para garantir um desempenho excepcional e uma experiência de usuário de alta qualidade para todos os tipos de plataformas e telas.

8. CONCLUSÃO

Ao longo deste trabalho, foi explorado diversas técnicas, ferramentas e padrões relacionados à responsividade para a web e mobile, tais como: HTML, CSS, Javascript, Figma e outras. Seu objetivo primordial foi investigar como padrões, diretrizes, e práticas comuns podem ser implementadas de forma eficaz para melhorar a experiência do usuário em sites e aplicações web, com um enfoque especial para dispositivos móveis.

Ainda que existam diversas organizações que criam e atualizam novas diretrizes e padrões para a criação de novas aplicações, como a W3C e WCAG, existe muito espaço para novas ideias e a criação de boas práticas para desenvolvedores. Essas organizações desempenham um papel fundamental ao estabelecerem as bases para a responsividade na web, fornecendo diretrizes e padrões reconhecidos internacionalmente. No entanto, à medida que a tecnologia avança e as necessidades dos usuários evoluem, novas abordagens e soluções inovadoras também se fazem necessárias. Nesse sentido, essas organizações têm o desafio de acompanhar estas mudanças e adaptar suas diretrizes e padronizações visando garantir que permaneçam relevantes e eficazes. Ao mesmo tempo, há espaço para o surgimento de novas organizações, iniciativas, e diretrizes que possam complementar o trabalho das já existentes, trazendo perspectivas novas e soluções criativas para os desafios da responsividade dentro da web. Essa dinâmica entre organizações já pré-estabelecidas e novos atores é essencial para impulsionar a inovação e criatividade, promovendo um ambiente digital mais inclusivo e acessível para todos.

Mesmo com a diversidade de organizações que estabelecem e atualizam diretrizes e padrões para a criação de novas aplicações, comunidades online também desempenham um papel crucial no apoio aqueles que desejam desenvolver ou aprender mais sobre responsividade tanto para a web como para mobile. Plataformas como Stack Overflow e GitHub são plataforma oferecem ambientes onde desenvolvedores podem colaborar, compartilhar conhecimentos e discutir questões relacionadas ao tema, também há grupos dentro do próprio LinkedIn, como "Web Accessibility Initiative" e "Responsive Web Designers", assim como fóruns dedicados a esses tópicos em plataformas como Reddit e Discord, proporcionando um espaço valioso para a troca de ideias e resolução de problemas. Essas comunidades fortalecem o engajamento e a colaboração entre profissionais e entusiastas da web, impulsionando a criação de aplicações mais inclusivas e adaptáveis.

Durante o desenvolvimento deste trabalho, foi proposto um estudo de caso, onde alguns desafios surgiram, especialmente no que diz respeito à otimização do desempenho e acessibilidade para dispositivos móveis. Para contornar estes desafios, foi adotado uma abordagem abrangente que incluiu a aplicação de diversas práticas para melhorar essas métricas, a implementação dessas práticas não apenas tornou o site mais acessível e eficiente, mas também contribuiu para uma experiência de usuário mais inclusiva e consistente.

O estudo de caso realizado foi essencial para aplicar na prática os conhecimentos adquiridos ao longo deste trabalho. A análise detalhada das métricas fornecidas pelo Google Lighthouse revelou áreas de oportunidade para melhoria e guiou a implementação de técnicas específicas para otimizar o desempenho, acessibilidade, práticas recomendadas e SEO do site, os resultados obtidos por meio desta prática demonstrou que a aplicação dessas técnicas teve um impacto significativo nas métricas avaliadas, melhorando a experiência do usuário e posicionando o site de forma mais competitiva nos resultados de busca.

Em conclusão, este trabalho ressalta a importância da melhoria contínua no desenvolvimento web e destaca a necessidade de uma abordagem integrada e focada para garantir um desempenho excepcional e uma experiência de usuário de alta qualidade para todo tipo de plataforma e tela.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] Rimar Bezerra, WHESLLEY. **“ACESSIBILIDADE DIGITAL”**. 2019. Acesso em: 15. set de 2023.

[2] George, JAMES. Jason, BEAIRD. **“PRINCÍPIOS DO WEB DESIGN MARAVILHOSO”**. 2016. Acesso em: 16. set de 2023.

[3] G. D. S. Santos, GABRIEL. Cordeiro Gomes, ALINE. Paulo A. Ferreira, PEDRO. **“ESTUDO DA ACESSIBILIDADE DE WEBSITES COM FOCO NA PERCEPÇÃO DE USUÁRIOS COM DEFICIÊNCIA VISUAL”**. 2018. Disponível em < https://sol.sbc.org.br/index.php/sbsi_estendido/article/download/6201/6099/ >. Acesso em: 30. set de 2023.

[4] Severo Maciel, GISELE. **“O ESTUDO DO DESIGN RESPONSIVO PARA INTERFACES DE DISPOSITIVOS MÓVEIS: O REDESIGN DO SITE ROYAL PLAZA SHOPPING”**. 2018. Disponível em < <https://repositorio.animaeducacao.com.br/bitstream/ANIMA/3986/1/GisieleSeveroMacielArtigoP%C3%B3sDefesa.pdf> >. Acesso em: 30. set de 2023.

[5] W3C (World Wide Web Consortium). **“MOBILE WEB BEST PRACTICES 1.0”**. 2008. Disponível em < <https://www.w3.org/TR/mobile-bp/> >. Acesso em: 30. set de 2023.

[6] W3C (World Wide Web Consortium). **“Web Content Accessibility Guidelines (WCAG) 2.1”**. 2023. Disponível em < <https://www.w3.org/TR/WCAG21/> >. Acesso em: 03 de fevereiro de 2024.

[7] W3C (World Wide Web Consortium). **“Our Mission”**. 2024. Disponível em < <https://www.w3.org/mission/> >. Acesso em: 06 de fevereiro de 2024.

[8] W3C (World Wide Web Consortium). “**About us**”. 2024. Disponível em < <https://www.w3.org/about/> >. Acesso em: 06 de fevereiro de 2024.

[9] W3C (World Wide Web Consortium). “**Web Standards**”. 2024. Disponível em < <https://www.w3.org/standards/> >. Acesso em: 06 de fevereiro de 2024.

[10] MARCOTTE, Ethan. “**Responsive Web Design**”. 2010. Disponível em < <https://alistapart.com/article/responsive-web-design/> >. Acesso em 17 de março de fevereiro de 2024.

[11] W3C (World Wide Web Consortium). “**History**”. 2024. Disponível em < <https://www.w3.org/about/history/> >. Acesso em: 22 de março de 2024.

[12] IFRS (Instituto Federal do Rio Grande do Sul). “**Web Content Accessibility Guidelines – WCAG 2.1**”. 2021. Disponível em < <https://cta.ifrs.edu.br/web-content-accessibility-guidelines-wcag-2-1/> >. Acesso em 22 de março de 2024.

[13] MARTINS, Diana. “**Experiência do usuário: o guia completo para agradar seu público digital em 2024**”. 2022. Disponível em < <https://rockcontent.com/br/blog/experiencia-do-usuario/#:~:text=A%20experi%C3%Aancia%20do%20usu%C3%A1rio%20%C3%A9,valor%20gerado%20pelo%20conte%C3%BAdo%20acessado> >. Acesso em 31 de março de 2024.

[14] BBC (British Broadcast Channel). “**Tecnologia deixa humanos com atenção mais curta que de peixinho dourado, diz pesquisa**”. 2015. Disponível em < https://www.bbc.com/portuguese/noticias/2015/05/150515_atencao_peixinho_tecnologia_fn >. Acesso em 01 de abril de 2024.

[15] SUZUKI, Akira. “**Hoje, o tempo de atenção de um humano é menor que o de um peixinho-dourado**”. 2015. Disponível em < <https://www.tecmundo.com.br/comportamento/80090-tempo-atencao-humano-menor-o-de-um-peixinho-dourado.htm> >. Acesso em 01 de abril de 2024.

[16] DINAMIZE. “**O que é mobile first e por que é importante no marketing digital?**”. 2024. Disponível em < <https://www.dinamize.com.br/blog/o-que-e-mobile-first/> >. Acesso em 02 de abril de 2024.

[17] TERRA, Rafael. “**Conheça os 14 frameworks CSS mais usados por desenvolvedores**”. 2021. Disponível em < <https://rockcontent.com/br/talent-blog/frameworks-css/> >. Acesso em 15 de abril de 2024.

[18] Mozilla Foundation. “**Flexbox**”. 2023. Disponível em < https://developer.mozilla.org/pt-BR/docs/Learn/CSS/CSS_layout/Flexbox >. Acesso em 21 de abril de 2024.

[19] Mozilla Foundation. “**Grids**”. 2023. Disponível em < https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Grids >. Acesso em 21 de abril de 2024.

[20] Mozilla Foundation. “**Viewport**”. 2023. Disponível em < <https://developer.mozilla.org/pt-BR/docs/Glossary/Viewport> >. Acesso em 21 de abril de 2024.

[21] REBECA P. C. SILVA, Cleviane. MOREIRA, O. Leonardo. “**Um Estudo de Frameworks de Design Responsivo e Avaliação na Perspectiva da Acessibilidade na Web**”. 2019. Disponível em < https://repositorio.ufc.br/bitstream/riufc/45934/3/2019_art_crpcsilva.pdf >. Acesso em 28 de abril de 2024.

[22] Bootstrap. **“Breakpoints”**. Disponível em < <https://getbootstrap.com/docs/5.3/layout/breakpoints/#available-breakpoints> >. Acesso em 28 de abril de 2024.

[23] Wikipedia. **“W3C Markup Validation Service”**. 2023. Disponível em < https://en.wikipedia.org/wiki/W3C_Markup_Validation_Service >. Acesso em 28 de abril de 2024.

[24] Web Design Museum. **“The first HTML validator”**. 2024. Disponível em < <https://www.webdesignmuseum.org/web-design-history/the-first-html-validator-1994> >. Acesso em 28 de abril de 2024.

[25] LOPES, Michele. **“Figma, Sketch e Adobe XD: qual ferramenta é melhor, comparação”**. 2023. Disponível em < <https://ebaonline.com.br/blog/figma-sketch-adobe-xd-comparacao#top> >. Acesso em 03 de maio de 2024.

[26] LinkedIn. **“Como você superou os desafios comuns de web design responsivo?”**. Disponível em < <https://www.linkedin.com/advice/3/how-did-you-overcome-common-responsive-web-design-challenges?lang=pt&originalSubdomain=pt> >. Acesso em 08 de maio de 2024.

[27] CÔRA, Samara. **“8 ferramentas de SEO gratuitas para analisar e otimizar seu site”**. 2024. Disponível em < <https://c2ti.com.br/blog/8-ferramentas-de-seo-gratuitas-para-analisar-e-otimizar-seu-site> >. Acesso em 13 de maio de 2024.

[28] Google. **“Guia de otimização de mecanismos de pesquisa (SEO) para iniciantes”**. Disponível em < <https://developers.google.com/search/docs/fundamentals/seo-starter-guide?hl=pt-br> >. Acesso em 13 de maio de 2024.

[29] WCAG. **“Web Content Accessibility Guidelines (WCAG) 2.2”**. 2023. Disponível em < <https://www.w3.org/TR/WCAG22/> >. Acesso em 15 de maio de 2024.

[30] GALA, Ana Sofia. **“WCAG 2.2: o que esperar da atualização de Diretrizes de Acessibilidade de Conteúdo da Web”**. Disponível em < <https://www.handtalk.me/br/blog/wcag-2-2/> >. Acesso em 15 de maio de 2024.