



UEPB

**UNIVERSIDADE ESTADUAL DA PARAÍBA
CAMPUS I**

**CENTRO DE CIÊNCIAS E TECNOLOGIA
DEPARTAMENTO DE COMPUTAÇÃO
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

ÍTALO CAIO DE SOUSA CARVALHO

**DESEMPENHO DE APLICAÇÕES FRONT-ENDS E OS IMPACTOS EM
E-COMMERCE**

**CAMPINA GRANDE
2022**

ÍTALO CAIO DE SOUSA CARVALHO

**DESEMPENHO DE APLICAÇÕES FRONT-ENDS E OS IMPACTOS EM
E-COMMERCE**

Trabalho de Conclusão de Curso (Artigo) apresentado ao Departamento do Curso de Ciências da Computação da Universidade Estadual da Paraíba, como requisito parcial à obtenção do título de Graduado em Ciência da Computação.

Orientadora: Profa. Dra. Kézia de Vasconcelos Oliveira Dantas

**CAMPINA GRANDE
2022**

É expressamente proibido a comercialização deste documento, tanto na forma impressa como eletrônica. Sua reprodução total ou parcial é permitida exclusivamente para fins acadêmicos e científicos, desde que na reprodução figure a identificação do autor, título, instituição e ano do trabalho.

C331d Carvalho, Ítalo Caio de Sousa.

Desempenho de aplicações front-ends e os impactos em e-commerce [manuscrito] / Ítalo Caio de Sousa Carvalho. - 2022.
37 p. : il. colorido.

Digitado.

Trabalho de Conclusão de Curso (Graduação em Computação) - Universidade Estadual da Paraíba, Centro de Ciências e Tecnologia, 2022.

"Orientação : Profa. Dra. Kézia de Vasconcelos Oliveira Dantas, Coordenação do Curso de Computação - CCT."

1. Performance em computação. 2. E-commerce. 3. Comércio virtual. 4. Desenvolvimento de software. I. Título

21. ed. CDD 005.1

ÍTALO CAIO DE SOUSA CARVALHO

DESEMPENHO DE APLICAÇÕES FRONT-ENDS E OS IMPACTOS EM
E-COMMERCE

Trabalho de Conclusão de Curso de
Graduação em Ciência da Computação
da Universidade Estadual da Paraíba,
como requisito à obtenção do título de
Bacharel em Ciência da Computação.

Área de concentração: Engenharia de
Software.

Aprovado em 03 de Agosto de 2022.



Profa. Dra. Kézia de Vasconcelos Oliveira Dantas (DC - UEPB)
Orientador(a)



Profa. Dra. Sabrina de Figueiredo Souto (DC - UEPB)
Examinador(a)



Prof. Dr. Paulo Eduardo e Silva Barbosa (DC - UEPB)
Examinador(a)

À minha mãe pela dedicação, companheirismo e amizade,
DEDICO.

LISTA DE ILUSTRAÇÕES

Figura 1 –	Suporte ao formato webP.....	17
Figura 2 –	Tela inicial da loja.....	22
Figura 3 –	Tela de carrinho.....	23
Figura 4 –	Tela de cadastro.....	24
Figura 5 –	Tela de detalhes do produto.....	25
Figura 6 –	Tela de pagamento.....	25
Figura 7 –	Primeiro resultado do Lighthouse.....	28
Figura 8 –	Resultado do Lighthouse após mudança no formato das imagens.	29
Figura 9 –	Resultado do Lighthouse após dimensionamento correto das imagens	30
Figura 10 –	Resultado do Lighthouse após o carregamento lento dos componentes	31
Figura 11 –	Resultado do Lighthouse após o cacheamento.....	32

LISTA DE TABELAS

Tabela 1 –	Comparativo do antes e depois das métricas de performance.....	32
------------	--	----

SUMÁRIO

1	INTRODUÇÃO	9
2	FUNDAMENTAÇÃO TEÓRICA	10
2.1	E-commerce	10
2.2	Performance em aplicações web	11
2.2.1	Métricas de desempenho de web sites	12
2.3	Impacto das métricas de performance em lojas online	14
2.4	Técnicas para melhoria de performance	15
2.4.1	Formato das imagens	16
2.4.1.1	Webp.....	16
2.4.1.2	Suporte a navegadores	17
2.4.2	Redimensionar as imagens	17
2.4.3	Carregamento lento das imagens	18
2.4.4	Carregamento lento dos componentes	19
2.4.5	Cache	19
2.4.5.1	Service workers e api de cache.....	20
2.5	Ferramenta de avaliação de desempenho	20
3	DESENVOLVIMENTO DO E-COMMERCE	21
3.1	Página inicial	22
3.2	Tela de carrinho e tela de cadastro	22
3.3	Página de detalhes do produto	24
3.4	Tela de pagamento	25
4	ESTUDO DE CASO	26
4.1	Tecnologias de desenvolvimento	26
4.2	Análise dos resultados	27
4.2.1	Aplicação do webp para formato das imagens	28
4.2.2	Redimensionamento e carregamento lento das imagens	29
4.2.3	Carregamento lento dos componentes	30
4.2.4	Cacheamento	31
4.2.5	Comparativo do antes e depois das métricas de performance do Lighthouse	32
5	CONCLUSÃO	33

REFERÊNCIAS..... 35

DESEMPENHO DE APLICAÇÕES FRONT-ENDS E OS IMPACTOS EM E-COMMERCE

RESUMO

Esse trabalho tem como objetivo a construção de um e-commerce no front end utilizando as boas práticas de programação, arquitetura concisa de software e que seja rápido e performático. Assim, foram utilizados para a construção do modelo funcional alguns dos métodos recomendados para desenvolvimento, como padrões de projeto, bibliotecas auxiliares, frameworks, entre outros. Posterior ao desenvolvimento do estudo de caso, foram realizadas análises por meio da ferramenta Lighthouse, e verificações de performance para identificar os pontos de melhoria presentes na aplicação. Desta forma, a partir da identificação dos problemas existentes, foram aplicadas técnicas de performance para melhorar a velocidade das páginas. Os resultados obtidos foram: melhora na renderização de conteúdo, o que fez com que o início do carregamento da página fosse mais rápido, otimização no tempo de carregamento dos conteúdos maiores, como imagens, por exemplo, aprimoramento no tempo de interatividade com a página, ou seja, os usuários conseguem interagir mais rapidamente com a aplicação, melhora no tempo de bloqueio, na mudança de layout cumulativa e na velocidade com que o conteúdo é exibido visualmente durante o carregamento da página.

Palavras-chave: Performance. E-commerce. Desempenho. Lighthouse.

ABSTRACT

This work aims to build an e-commerce on the front end using good programming practices, concise software architecture and that is fast and performant. Thus, some of the methods recommended for development were used to build the functional model, such as design patterns, auxiliary libraries, frameworks, among others. After the development of the case study, analyzes were carried out using the Lighthouse tool, and performance checks were carried out to identify the points of improvement present in the application. In this way, from the identification of the existing problems, performance techniques were applied to improve the speed of the pages. The results obtained were: improvement in content rendering, which made the page load start faster, optimization in the loading time of larger content, such as images, for example, improvement in the interactivity time with the page, that is, users are able to interact more quickly with the application, improve the blocking time, the cumulative layout change and the speed with which the content is displayed visually during the page load.

Keywords: Performance. E-commerce. Lighthouse.

1 INTRODUÇÃO

Com os avanços tecnológicos e da internet, plataformas de comércio eletrônico, comumente conhecidas como e-commerce, vêm se tornando cada vez mais populares tanto para os consumidores quanto para as grandes empresas. O processo de compra de bens e serviços através da internet só cresce e se consolida dia após dia. Atualmente, os consumidores estão gradativamente mais dependentes dos dispositivos eletrônicos para acessar conteúdos e serviços online e também estão mais exigentes em relação à experiência do site ou aplicativo em que estão navegando.

Desta forma, o desempenho e velocidade de sites online têm sido pontos importantes para o sucesso de qualquer e-commerce, pois sites de alto desempenho envolvem mais os usuários e possuem taxas mais altas de retenção quando comparados com os de baixo desempenho. O desempenho é importante pois quanto menor for o tempo de carregamento do site, melhor será a experiência do usuário em qualquer conexão de internet. Melhorar a experiência do usuário resulta em um maior interesse por parte dele no site, pois páginas lentas fazem o usuário perder a paciência e, por consequência, a abandonam muito antes de ver o que ela tem a oferecer (WAGNER, 2016).

Estudos, como o de Sean Work¹, mostram que a maioria dos usuários se deparam com problemas relacionados à performance quando utilizam a internet, aplicação lenta, travando, com formatação ilegível e sem disponibilidade são comuns. Além disso, o trabalho de Gomez Peak² concluiu que mais da metade dos consumidores deixam um site lento por um mais rápido ao invés de esperar; que a maior parte dos consumidores são menos dispostos a retornar a uma página quando a experiência é ruim; que quase metade das pessoas passam a enxergar a empresa de forma negativa e que muitos compartilham essa experiência com outras pessoas.

Sendo assim, este trabalho se propõe a avaliar performance e desempenho de aplicações front end. Como modelo de avaliação, será desenvolvido um estudo de caso de uma plataforma de e-commerce, avaliando o seu desempenho, melhorando a sua performance e apresentando os resultados. A ferramenta utilizada

¹ <https://neilpatel.com/blog/loading-time/?wide=1>

² <http://munchweb.com/effect-of-website-speed>

para verificar a performance da aplicação foi o Lighthouse³, uma ferramenta automatizada e de código aberto para melhorar a qualidade das páginas da Web.

As métricas de performance analisadas⁴ no trabalho foram: First Contentful paint (FCP), ou primeira renderização de conteúdo, Largest Contentful Paint (LCP), ou maior pintura de conteúdo, Time to Interactive (TTI), ou tempo até interatividade, Total Blocking Time (TBT), ou tempo total de bloqueio, Cumulative layout shift (CLS), ou mudança de layout cumulativa e Speed Index, ou índice de velocidade.

2 FUNDAMENTAÇÃO TEÓRICA

Para um melhor entendimento acerca do desempenho de aplicações web, este trabalho é introduzido falando sobre o que é e-commerce, como surgiu; retrata sobre performance de aplicações web, definindo o que é e a sua importância; descreve as métricas de desempenho de web sites; pontua os impactos da performance em aplicações e as técnicas para melhoria de performance.

2.1 E-Commerce

O surgimento da internet provocou grandes transformações no mundo, inclusive na forma de efetivar vendas, o que ficou conhecido como e-commerce. O começo concreto do e-commerce se deu em 1991, quando a internet foi aberta para uso comercial (TASSABEHJI, 2003). Porém, somente após o desenvolvimento de protocolos de segurança como HTTP e SSL e sua implantação nos sistemas, as conexões e acessibilidade à internet foram aperfeiçoadas possibilitando o desenvolvimento de novos recursos online. Inicialmente, e-commerce significava o processo de execução de transações comerciais feitas em um ambiente eletrônico. Com o passar do tempo, com o seu desenvolvimento, o termo comércio eletrônico passou a ser definido como o processo de compra de bens e serviços disponíveis através da internet, utilizando conexões seguras e serviços de pagamento eletrônico (MAKELAINEN, 2006; MENDES, 2013).

Cada vez mais frequente, sendo amplamente adotado como estratégia entre empresas dos mais variados setores, o e-commerce tem se mostrado um aliado tanto dos interesses corporativos quanto dos consumidores. Sua adoção apresenta

³ <https://developer.chrome.com/docs/lighthouse/overview/>

⁴ <https://web.dev/user-centric-performance-metrics/>

diversas vantagens, tais como o pouco investimento inicial, grande crescimento do mercado, promoções, estoque, monitoramento de vendas, comodidade para os clientes, além do fato de que a loja não fecha, ou seja, ela trabalha 24 horas por dia e 7 dias na semana (ARAÚJO, 2014; MENDES, 2013).

De acordo com Moraes (2018), a estrutura de um comércio eletrônico possui diversas funcionalidades, e para torná-la reconhecida como boa, é preciso que ela possua segmentação por departamento e por marca, cupons de desconto, lista de presentes, integração por mídias sociais, busca por Tags, sistema de buscas, compras por ambiente, menus interativos, sistema de carrinho, layout exclusivo e integração com marketplace.

Entretanto, para o sucesso de um empreendimento online, apenas possuir as funcionalidades não são mais suficientes. Atualmente a velocidade do site representa um papel importante, pois sites de baixo desempenho envolvem menos os consumidores e possuem taxas mais baixas de retenção quando comparados com os de alto desempenho.

2.2 Performance em aplicações web

A performance ou desempenho web trata-se da velocidade de carregamento de uma página ou site. O site carregando rapidamente permite que o usuário comece a interagir com ele rapidamente e oferece feedback se algo estiver demorando para carregar.

O desempenho é importante pois quanto menor for o tempo de carregamento do site melhor será a experiência do usuário em qualquer conexão de internet. Melhorar a experiência do usuário resulta em um maior interesse por parte dele no site, pois páginas lentas fazem o usuário perder a paciência e, por consequência, a abandonam muito antes de ver o que ela tem a oferecer (WAGNER, 2016).

Conforme escreveu Jeremy Wagner (2016), sites rápidos proporcionam uma melhor experiência para o usuário. Fazer com que os sites possuam um bom desempenho, faz com que o usuário tenha uma experiência melhor e com isto acelera a entrega de conteúdo. É possível ver também que a medida que o site se torna mais rápido, os usuários têm um maior interesse pelo que está nele. Um site que não carrega rapidamente quase nenhum usuário possui interesse.

De acordo com a documentação para desenvolvedores do mozilla⁵, o desempenho da Web é a experiência percebida do usuário de um site ou aplicativo e ela pode ser medida. Isso inclui as seguintes áreas principais:

- Tempo de carregamento: O tempo gasto para os arquivos renderizar o site tende a ser afetado pela latência, pelo tamanho dos arquivos, quantos arquivos existem e outros fatores.
- Tempo de interação: significa carregar os recursos do site em uma ordem sensata para que o usuário possa começar a usá-lo realmente rapidamente. A medida de quanto tempo leva para o site chegar a um início utilizável após o início do carregamento é chamada de tempo de interação.
- Desempenho percebido: a rapidez com que um site parece ao usuário tem um impacto maior na experiência do usuário do que a velocidade real do site. Como um usuário percebe o desempenho é tão importante, ou talvez mais importante, do que qualquer estatística objetiva, mas é subjetivo e não tão prontamente mensurável.
- Medições de desempenho: o desempenho da Web envolve a medição das velocidades reais e percebidas de um aplicativo, otimizando sempre que possível e monitorando o desempenho. Isso envolve uma série de métricas e ferramentas para medir essas métricas.

Na próxima seção serão apresentadas as principais métricas para avaliar o desempenho de aplicações web.

2.2.1 Métricas de desempenho de web sites

De acordo com Ben Faw, cofundador da Pantheon (plataforma de WebOps, Operações de site), em seu artigo sobre as principais métricas de performance (Top Ten Website Performance Metrics, 2021) existem dez métricas essenciais para uma aplicação web como um eCommerce elas são: a velocidade da página, o Tempo para Título, a renderização, o Tempo para interagir, a Velocidade de pesquisa de DNS, a Taxa de rejeição, as Solicitações por segundo, a Taxa de erro, o Tempo para Primeiro Byte/Último Byte e Taxa de conversão.

Existem algumas ferramentas para avaliar a velocidade dos sites, elas são:

⁵ https://developer.mozilla.org/en-US/docs/Learn/Performance/What_is_web_performance

- Page Speed⁶, um complemento de código aberto para Firefox/Firebug que avalia o desempenho de páginas da web e oferece sugestões de melhorias;
- YSlow⁷, uma ferramenta do Yahoo! que sugere maneiras de melhorar a velocidade do site;
- O WebPagetest⁸, que mostra uma visualização em cascata do desempenho de carregamento de suas páginas, além de uma lista de verificação de otimização;
- Lighthouse⁹, que é uma ferramenta automatizada e de código aberto para melhorar a qualidade das páginas da Web.

Neste trabalho será utilizado como ferramenta o Lighthouse, pois as métricas de performance avaliadas por esta ferramenta abrangem mais as métricas essenciais para uma aplicação de e-commerce. Em sua plataforma de artigo para desenvolvedores (web.dev¹⁰), a Google afirma que o Lighthouse possui seis métricas de performance em sua análise de desempenho centradas no usuário elas são:

- **First contentful paint (FCP)**, ou primeira renderização de conteúdo, mede o tempo desde o início do carregamento da página até o momento em que qualquer parte do conteúdo da página é renderizada na tela.
- **Largest contentful paint (LCP)**, ou maior renderização de conteúdo mede o tempo desde o início do carregamento da página até o momento em que o maior bloco de texto ou elemento de imagem é renderizado na tela.
- **Time to Interactive (TTI)**, ou tempo até interatividade, mede o tempo desde o início do carregamento da página até o momento em que é renderizada visualmente, seus scripts iniciais (se houver) tenham sido totalmente carregados e que seja capaz de responder de forma confiável e rápida à entrada do usuário.
- **Total blocking time (TBT)**, ou tempo total de bloqueio mede a quantidade total de tempo entre a FCP e a TTI onde a thread principal foi bloqueada por tempo suficiente para evitar a responsividade de entrada.

⁶ <https://developers.google.com/speed?csw=1>

⁷ <http://yslow.org/>

⁸ <https://www.webpagetest.org/>

⁹ <https://developer.chrome.com/docs/lighthouse/overview/>

¹⁰ <https://web.dev/user-centric-performance-metrics/>

- **Cumulative layout shift (CLS)**, ou mudança de layout cumulativa, mede a pontuação cumulativa de todas as mudanças de layout inesperadas que ocorrem entre quando a página começa a ser carregada e quando seu estado de ciclo de vida muda para "oculta".
- **Speed Index**, ou índice de velocidade, que mede a rapidez com que o conteúdo é exibido visualmente durante o carregamento da página.

Segundo Philip Walton, engenheiro da Google, em sua publicação sobre métricas de performance (User-centric performance metrics, 2019), existem tipos de métricas que são relevantes para a forma como os usuários percebem o desempenho. As métricas consideradas por ele como importantes são as mesmas que o Lighthouse mensura, porém por mais que elas verifiquem diversos aspectos de desempenho úteis para os clientes finais elas não são completas pois existem outras métricas que podem ser relevantes como:

- suavidade, que verifica se as as transições e animações são renderizadas corretamente e se fluem com facilidade de um estado para o outro;
- responsividade, que mede a velocidade da página de responder à interação do usuário, após carregada.

De acordo com Philip(2019) as métricas de desempenho geralmente são medidas de duas maneiras:

- No laboratório: usando ferramentas para simular o carregamento de uma página em um ambiente consistente e controlado;
- No campo : em usuários reais carregando e interagindo com uma página no mundo real.

Este trabalho foca em avaliar as métricas no laboratório.

2.3 Impacto das métricas de performance em lojas online

No ano de 2008, a empresa Amazon criou uma métrica que passou a ser um modelo a respeito de desempenho de sites, que é a seguinte: a cada 100 milissegundos de latência, 1% das vendas da loja é perdida (KOHAVI; LONGBOTHAM, 2007). O Pinterest também revelou que reduziu os tempos de espera percebidos em 40% e isso aumentou o tráfego do mecanismo de pesquisa e as inscrições em 15%.

Em sua pesquisa Sean Work¹¹ aponta que 60% dos usuários encontram problemas relacionados à performance e usabilidade ao navegar pela internet e que as experiências em dispositivos móveis não são agradáveis. Os seguintes problemas foram apontados:

- O site estava muito lento para carregar
- O site travou, congelou ou gerou um erro
- O site foi formatado de forma ilegível
- O site não funcionou como esperado
- O site estava indisponível

Todas essas questões impactam diretamente na experiência do usuário, gerando assim uma frustração que não é desejada, além de levar a uma probabilidade menor de o cliente não finalizar o pedido, não retornar a loja para compras futuras e não recomendar para amigos e familiares. O Estudo de caso de Gomez Peak¹², que foi realizado com 1500 consumidores concluiu que:

- Mais de 75% dos consumidores deixaram um determinado site de compras para um concorrente em vez de sentar e esperar que uma página lenta fosse renderizada;
- 88% dos consumidores dizem que são menos propensos a retornar a um site após uma experiência ruim;
- Quase metade diz que vê a própria empresa de forma negativa após uma única experiência ruim com o site da empresa;
- Mais de 33% dos consumidores contaram a outras pessoas sobre uma experiência decepcionante que tiveram com um site.

2.4 Técnicas para melhoria de performance

Para melhorar a performance da aplicação construída, foi aplicado técnicas de performance no frontend para otimizar o tempo de carregamento das páginas. Assim, nas subseções subsequentes serão apresentadas as técnicas utilizadas, que são: Formato das imagens, Redimensionar as imagens, Carregamento lento das imagens, Carregamento lento dos componentes e Cache.

¹¹ <https://neilpatel.com/blog/loading-time/?wide=1>

¹² <http://munchweb.com/effect-of-website-speed>

2.4.1 Formato das imagens

As imagens são muito utilizadas para diversas funcionalidades no ambiente digital, principalmente em plataformas de e-commerce. No entanto, as imagens podem estar em diversos formatos digitais, como por exemplo:

- JPEG - que de acordo com a norma internacional(ISO 10918-1) é definido como uma família de algoritmos de compressão e descompressão, com e sem perda, para imagens de qualidade fotográfica, também designadas por imagens de tons contínuos.
- PNG (Portable Network Graphics) - que surgiu como resposta às limitações técnicas e às restrições legais derivadas dos direitos de propriedade do formato GIF, facilitando a troca de imagens pela internet.

Apesar desses formatos serem úteis, atualmente existem novos formatos que melhoram os recursos das imagens e a compressão e a fidelidade, eles são: WebP do Google, JPEG XR da Microsoft e o JPEG 200 (BENDELL et al., 2016).

A única desvantagem desses formatos é que atualmente nem todos os navegadores os suportam. Hoje os principais navegadores como Chrome, Mozilla, Edge e Safari já possuem suporte, pelo menos de forma parcial ao WebP que resolvemos utilizar na aplicação.

Os formatos das imagens utilizadas na loja desenvolvida a princípio foram JPEG e PNG. Entretanto, para melhorar o desempenho da aplicação, no contexto deste trabalho foi utilizado o formato das imagens webP.

2.4.1.1 Webp

O WebP foi desenvolvido e promovido pela Google. Esse formato se tornou a primeira imagem especialmente feita para navegador para ganhar adoção dos desenvolvedores web. Ele é fundamentado no *codec de vídeo VP8* da Google o formato realiza a codificação de imagem intraframe do *VP8* em um contêiner de imagem RIFF (BENDELL et al., 2016).

Atualmente, temos três variações distintas do WebP conforme Bendell et al. (2016):

- Básica - é simples, suporta codificação de apenas uma imagem opaca, assim como o JPEG.
- Estendida - ela adiciona suporte para compressão sem perdas e transparência total.

- Animada - é construída através da variação estendida e adiciona suporte de animação, fazendo elas serem substitutas dos GIFs.

As variações do WebP demonstram o quanto esse formato busca evoluir para melhorar e adicionar recursos, porém também apresentam um problema ainda de compatibilidade entre navegadores.

2.4.1.2 Suporte a navegadores

Através do site CanIUse, é possível analisar todos os navegadores que suportam o formato WebP. Na Figura 1 são apresentados todos os navegadores por colunas e cada célula na coluna corresponde a uma versão do navegador. Se o quadrado é verde tem compatibilidade, se for vermelho não tem e se for amarelo escuro tem parcialmente. O Internet Explorer e o KaiOS não oferecem suporte nas versões atuais e o Safari apresenta suporte apenas de forma parcial. Mas, todos os outros navegadores em suas versões atuais são suportáveis com o formato.

Figura 1 – Suporte ao formato webP

IE	Edge *	Firefox	Chrome	Safari	Opera	Safari on iOS *	Opera Mini *	Android Browser *	Opera Mobile *	Chrome for Android	Firefox for Android	Browser for Android	Samsung Internet	QQ Browser	Baidu Browser	KaiOS Browser
			4-8		10.1											
			9-22		11.5			2.1-3								
	12-17	2-64	23-31	3.1-13.1	12.1-18	3.2-13.7		4-4.1								
6-10	18-98	65-96	32-98	14-15.1	19-82	14-15.1		4.2-4.4.4	12-12.1				4-15.0			
11	99	97	99	15.3	83	15.3	all	98	64	98	96	12.12	16.0	10.4	7.12	2.5
		98-99	100-102	15.4-TP		15.4										

Fonte: (CanIUse, 2022)

2.4.2 Redimensionar as imagens

Na maioria das vezes as imagens são responsáveis pela maioria da carga total de um site, ou seja, elas são responsáveis pela maior parte do tamanho de uma página. Comumente, para que as imagens sejam apresentadas da melhor forma possível, elas são disponibilizadas em alta resolução para os dispositivos (WAGNER, 2016).

Porém, entregar uma imagem de alta resolução para todos os dispositivos pode afetar a performance da página, pois todos eles precisam utilizar uma imagem muito grande e redimensioná-la para caber na tela, sejam eles desktops ou mobiles.

Como resultado, a página poderia ter um tamanho maior e demorar mais tempo para ser baixada e renderizada.

Sendo assim, entregar uma imagem de boa qualidade é importante para a experiência do usuário, mas é preciso pensar também na capacidade dos dispositivos que a recebem. Possuir o entendimento de como entregar as imagens de forma correta faz com que os dispositivos com menor capacidade não sejam sobrecarregados e faz com que os dispositivos com maior poder de processamento continuem recebendo a melhor experiência possível (WAGNER, 2016).

Dessa forma, é necessário fornecer imagens de forma responsiva, isto é, no tamanho e no dimensionamento correto para cada dispositivo. À medida em que as imagens são planejadas para serem entregues de forma correta há, conseqüentemente, uma melhor experiência para os usuários.

2.4.3 Carregamento lento das imagens

Segundo Mjelde e Opdahl (2017), uma forma final de reduzir os tempos de carregamento da página é atrasar o carregamento de recursos não críticos. Os recursos não críticos são aqueles que não estão visíveis inicialmente na página, que precisa necessariamente rolar a barra para vê-los. O Lazy Loading ou carregamento lento é uma técnica que inicialmente carrega apenas os recursos que são exibidos dentro da janela de visualização do navegador. O carregamento de outros recursos incorporados (o chamado conteúdo abaixo da dobra) é atrasado até que um elemento de espaço reservado que representa a imagem não baixada seja deslocado para dentro da janela de visualização do navegador.

Em seu trabalho, Mjelde e Opdahl (2017) afirmam que a vantagem central do carregamento lento é que ele causa menos solicitações HTTP no carregamento inicial da página, porque as solicitações serão feitas apenas para imagens dentro da janela de visualização. Por consequência, o carregamento lento também reduz o tamanho da página no carregamento inicial da página, porque as imagens compreendem a maior parte dos tamanhos médios de página da web. Um tamanho de página menor pode, por sua vez, levar a tempos de carregamento mais rápidos, especialmente em conexões de baixa largura de banda, como redes de telefonia móvel (celular). Além disso, o carregamento lento evita o download de imagens que nunca são visualizadas, o que é um problema ao pagar por byte baixado em redes de telefonia móvel.

2.4.4 Carregamento lento dos componentes

O carregamento lento é uma técnica que pode ser usada para qualquer recurso obtido de forma assíncrona (MJELDE; OPDAHL, 2017). Dessa forma, esta técnica também pode ser usada para os componentes da aplicação.

Para aplicar o carregamento lento nos componentes é preciso utilizar outra técnica que é a divisão de código. A divisão de código divide a aplicação em pedaços e uma vez dividida torna-se possível carregar os seus pedaços de forma lenta. Em muitos cenários, a página não irá precisar de todo o código do pacote javascript de forma imediata. Assim, o carregamento lento permite a divisão do pacote e disponibilizar apenas as partes que de fato são importantes, fazendo assim que os usuários não percam tempo baixando e analisando o código não utilizado (RAKOWSKI, 2019).

2.4.5 Cache

O Armazenamento em cache é um método de armazenamento de dados geralmente utilizado numa memória de acesso rápido em que é possível usar logo após a solicitação. Sendo assim, os navegadores muitas vezes omitem a rede e acessam os arquivos da máquina local (RAKOWSKI, 2020).

Os navegadores têm a capacidade de guardar os dados em *Memory Cache*, que é o armazenamento em memória RAM, esta forma de persistir as informações acontece enquanto o navegador está aberto. Outro tipo de armazenamento é o em *Disk Cache*, que é no disco rígido e que leva bem mais tempo para ser acessado. Os navegadores já realizam o cacheamento dos ativos estáticos da página, como CSS, HTML, arquivos JavaScript e também imagens (RAKOWSKI, 2020).

Entretanto, o cacheamento feito possui várias limitações. Uma delas é relacionada ao tamanho, o cache de navegador é muito pequeno e por causa disso é comum os ativos armazenados em cache serem substituídos por novos. Outro ponto é o seu comportamento incerto. Cada navegador possui técnicas diferentes de cacheamento. Vale ressaltar que algumas configurações de dispositivo são capazes de impactar o funcionamento. Outro ponto negativo é que apenas é possível recuperar os arquivos em cache enquanto houver conexão com a internet. Se não houver conexão apresentaram a imagem de sem conectividade ou offline (RAKOWSKI, 2020).

2.4.5.1 Service workers e api de cache

O Service Workers e o Cache API são funcionalidades do navegador que podem resolver as limitações do cache dos navegadores. Segundo LePage (2020), a API de cache é um sistema para armazenar e recuperar solicitações de rede e suas respostas correspondentes. Podem ser solicitações e respostas regulares criadas durante a execução de seu aplicativo ou podem ser criadas apenas para armazenar dados para uso posterior. Assim, a API de cache pode ser utilizada para manter as respostas de rede e assim temos controle sobre elas.

Segundo a documentação para desenvolvedores da Google¹³, Service Workers são ativos JavaScript especializados que atuam como proxies entre navegadores da web e servidores da web. Eles visam melhorar a confiabilidade fornecendo acesso offline, bem como aumentar o desempenho da página.

A junção de Service Workers e Cache API permite servir de forma programática alguns arquivos. Fazendo com que seja possível o controle sobre o que deve ser armazenado em cache e quando deve. Com o Service Worker atuando como um proxy, também é possível fazer a aplicação funcionar no modo offline (RAKOWSKI, 2020).

2.5 Ferramenta de avaliação de desempenho

A ferramenta utilizada para verificar a performance da aplicação foi o Lighthouse. De acordo com a Google, empresa gerenciadora deste recurso, o Lighthouse¹⁴ é uma ferramenta automatizada e de código aberto para melhorar a qualidade das páginas da Web. Ela permite a execução em todo tipo de página da Web, público ou que exija autenticação. Possui auditorias para desempenho, acessibilidade, aplicativos web progressivos, SEO, entre outros.

Na página Developers Google que contém artigos para desenvolvedores, este recurso já vem incorporado no navegador do Chrome. Mas, além de executar o Lighthouse no Chrome DevTools, pode-se executar por linha de comando, ou como um módulo Node. Apenas é necessário fornecer ao Lighthouse uma URL para auditar, ela executa uma série de auditorias contra a página e, em seguida, gera um relatório sobre a sua performance. Com isso, podemos usar as suas auditorias que

¹³ <https://developer.chrome.com/docs/workbox/service-worker-overview/>

¹⁴ <https://developer.chrome.com/docs/lighthouse/overview/>

falharam para otimizar a página. Em cada análise que é feita um documento de referência explicando por que a auditoria é importante, bem como como corrigi-la.

O Lighthouse faz parte do painel de auditorias do Chrome DevTools. Segue os passos para executar um relatório na ferramenta:

1. Abrir o Google Chrome para desktop;
2. No Google Chrome, ir para a URL que deseja auditar;
3. Abrir o Chrome DevTools;
4. Clicar na guia **Lighthouse**;
5. Clicar em **Gerar relatório**. Depois de 30 a 60 segundos, o Lighthouse apresentará um relatório da página.

3 DESENVOLVIMENTO DO E-COMMERCE

Nesta seção é apresentado o e-commerce desenvolvido, apresentando as funcionalidades e aspectos para uma loja online, assim como Moraes (2018) apresentou em seu trabalho. Dessa forma, foi possível definir os requisitos funcionais do projeto, eles são:

- O sistema deverá possuir segmentação por departamento;
- O sistema deverá permitir a busca por produtos através do seu nome;
- O sistema deverá permitir adição de itens ao carrinho e também permitirá que se exclua itens do carrinhos;
- O sistema deverá permitir manter o usuário, fazendo o cadastro do cliente;
- O sistema deverá permitir o login dos usuários cadastrados;
- O sistema deverá permitir a listagens de produtos;
- O sistema deverá permitir a apresentação dos detalhes de cada produto;
- O sistema deverá permitir o pagamento dos produtos;
- O sistema deverá permitir a apresentação da tela de conclusão de pedido;
- O produto deverá possuir nome, descrição, foto, avaliação e preço.

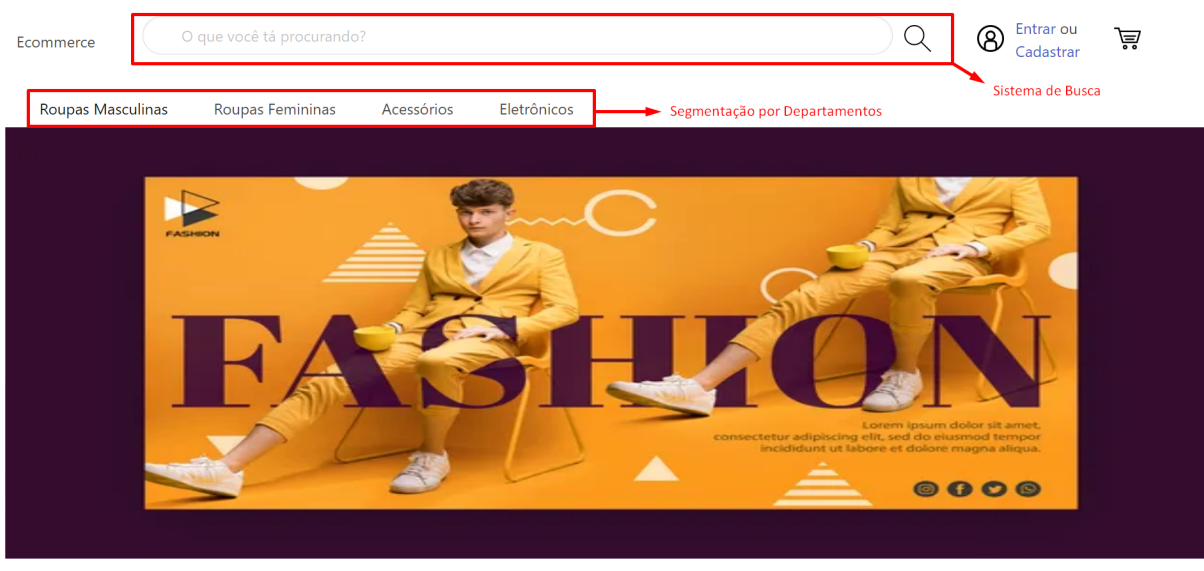
Nas subseções seguintes são apresentadas as páginas da loja, elas são: página inicial, tela de carrinho, página de cadastro, página de detalhes do produto e tela de pagamento.

3.1 Página inicial

Na página inicial do e-commerce desenvolvido (ver Figura 2), são apresentados dois requisitos dos definidos para o projeto: o sistema de pesquisa e a segmentação por departamentos.

Um sistema de busca que funcione corretamente é essencial numa loja, pois possibilita que os compradores encontrem os produtos que procuram de forma rápida. E o sistema de departamentos faz a separação dos produtos por departamentos e com isso, o e-commerce promove uma maior usabilidade, uma melhor experiência para o usuário e mais praticidade também.

Figura 2 – Tela inicial da loja






Fonte: Elaborado pelo autor, 2022.

3.2 Tela de carrinho e tela de cadastro

Outro requisito fundamental em um comércio eletrônico é o sistema de carrinho. A Figura 3 apresenta a tela de carrinho do projeto.

Figura 3 – Tela de carrinho

Ecommerce   [Entrar ou Cadastrar](#) 

Roupas Masculinas Roupas Femininas Acessórios Eletrônicos

CARRINHO INFORMAÇÃO DO USUÁRIO PAGAMENTO CONFIRMAÇÃO

Minha Cesta		Resumo do Pedido	
Produto	Preço	Produto	Preço
Fjallraven - Foldsack No. 1 Backpack, Fits 15 Laptops	109.95	Fjallraven - Foldsack No. 1 Backpack, Fits 15 Laptops	109.95
		Total: R\$109.95	em até 12x sem juros
		Continuar	
		Adicionar mais produtos	

Calcule o frete e prazo

[Não sei meu CEP](#)

Fonte: Elaborado pelo autor, 2022.

Para clientes que não estão logados, após adicionar o produto no carrinho e continuar a compra, o usuário é encaminhado para a tela de cadastro para a conclusão de pedido, conforme pode ser visualizado na figura abaixo (Figura 4). Com isso, foi implementado um cadastro em uma única página, exigindo dos clientes somente os dados que são necessários para finalizar a compra.

Figura 4 – Tela de cadastro

Ecommerce [Entrar ou Cadastrar](#)

[Roupas Masculinas](#) [Roupas Femininas](#) [Acessórios](#) [Eletrônicos](#)

Criar seu Cadastro

Veja seus pedidos de forma fácil, compre mais rápido e tenha uma experiência personalizada

Nome Completo

Gênero

Masculino Feminino

Data de nascimento

CPF

Necessário para emissão das notas fiscais

Telefone

Caso a gente precise entrar em contato sobre seus pedidos

Receber notificações por WhatsApp

Email

Receber e-mails promocionais

Senha

precisa ter entre 6 a 20 caracteres

[Criar Cadastro](#)

Já tem cadastro? [Entrar](#)

Fonte: Elaborado pelo autor, 2022.

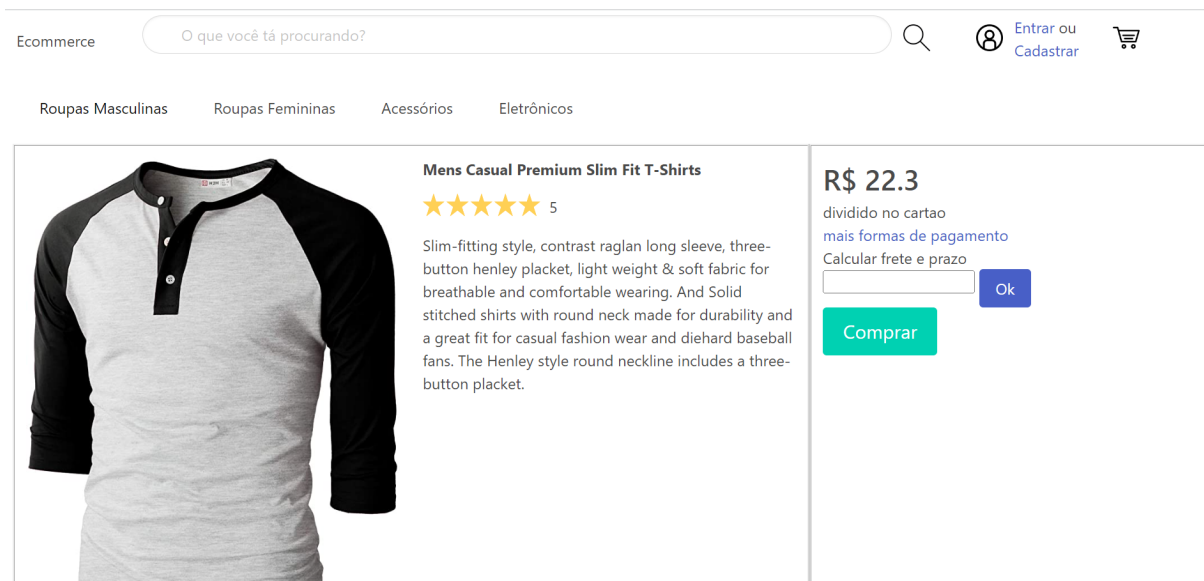
3.3 Página de detalhes do produto

Na página de detalhes do produto, são apresentadas as informações do produto. Abaixo a Figura 5 mostra esta página exibindo o nome, descrição, foto, avaliação, preço, cálculo de frete, formas de pagamento e botão de compra.

Dois pontos importantes que fazem parte dos requisitos são:

- O produto possuir imagem, pois a maioria dos consumidores preferem produtos que tenham fotos;
- A descrição do produto, pois a maioria dos compradores gostam de uma descrição bem definida para decidirem a compra do produto.

Figura 5 – Tela de detalhes do produto



Fonte: Elaborado pelo autor, 2022.

3.4 Tela de pagamento

Na Figura 6 é apresentada a tela de pagamento, projetada levando em consideração todo o público alvo, os que possuem cartão de crédito/débito e os que não possuem, oferecendo diversas opções de pagamento.

Figura 6 – Tela de pagamento

The screenshot shows a payment page titled 'Formas de pagamento'. At the top, there is a navigation bar with a search bar, a user profile icon with 'Entrar ou Cadastrar', and a shopping cart icon. Below this is a secondary navigation bar with categories: 'Roupas Masculinas', 'Roupas Femininas', 'Acessórios', and 'Eletrônicos'. The main heading is 'Formas de pagamento', with sub-links for 'Cartão de crédito' and 'Boleto'. The 'Cartão de crédito' section is active and contains the following fields: 'Cadastrar Cartão', 'Número do Cartão', 'Nome Completo', 'Validade', and 'Código de segurança'. Below these fields is a checkbox labeled 'Salvar o cartão para futuras compras'. At the bottom left, there is a 'Total:' label and a 'Fechar Pedido' button. At the bottom right, the total amount is displayed as 'R\$ 109.95'.

Fonte: Elaborado pelo autor, 2022.

4 ESTUDO DE CASO

Neste capítulo serão apresentadas as tecnologias utilizadas para desenvolvimento do projeto do estudo de caso, os resultados obtidos a partir da utilização da ferramenta Lighthouse para estudo da performance, bem como as ações realizadas para melhoria da performance.

4.1 Tecnologias de desenvolvimento

O projeto foi desenvolvido utilizando o framework Nuxt como base. Este framework tem como foco tornar o desenvolvimento web intuitivo e de alto desempenho com uma excelente experiência para o desenvolvedor.

O Nuxt foi escolhido porque era preciso uma ferramenta que suprisse as necessidades da aplicação web. Estas necessidades foram:

1. Um Framework JavaScript que ofereça reatividade e componentes web, o Nuxt dispõe do Vue.js;
2. Algum empacotador que forneça suporte à troca de módulos no desenvolvimento e que agrupe o código para produção, ele oferece suporte a dois no qual foi optado pelo Webpack no projeto;
3. Um transpilador para escrever a sintaxe JavaScript atual e também dar suporte a navegadores legados, nele é disponibilizado o esbuild;
4. Um servidor para rodar a aplicação em ambiente de desenvolvimento e que também consiga lidar com renderização do lado do servidor ou rotas de API, ele utiliza o vue-router.

Segundo a documentação, o Nuxt também oferece uma estrutura de pasta na qual foi seguida para estruturar todo o projeto desenvolvido, focada em recursos específicos, facilitando o desenvolvimento.

Como biblioteca de estilos foi utilizado o Bulma que é um framework CSS moderno. Ela é gratuita e o código é aberto, dispondo de componentes prontos para aplicação facilitando a criação de interfaces responsivas.

Para simular os dados da aplicação foi usado o FakeStoreAPI. Ela é uma API REST gratuita, o que permitiu a utilização de dados pseudo-reais para a loja online. Ela é utilizada apenas para estudo, códigos de amostra, testes entre outros.

A utilização do FakeStoreAPI se deu pela necessidade de utilizar dados falsos, com ela não foi preciso criar dados lorem ipsum, muito menos um arquivo JSON.

Assim, o desenvolvimento do template se resumiu nas seguintes linguagens: HTML, CSS e JavaScript.

Após o desenvolvimento da loja, foi preciso analisar sua performance e mensurar o seu tempo de carregamento para que os resultados de vendas e os lucros do ecommerce não sofressem por conta do baixo desempenho das páginas.

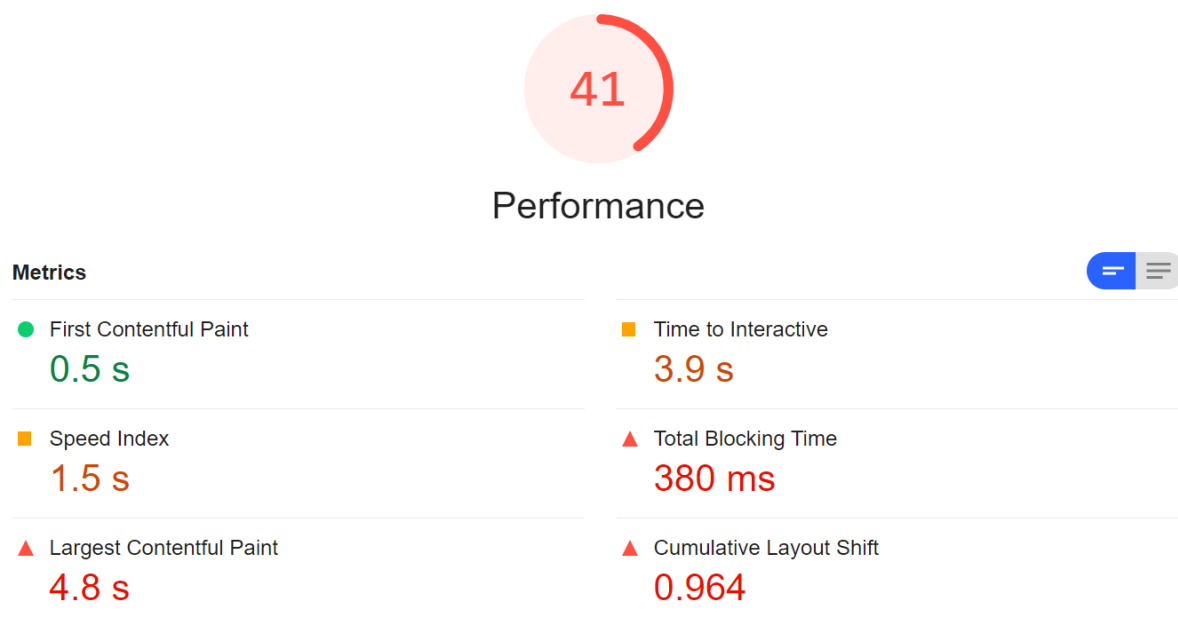
4.2 Análise dos resultados

Para medir a performance da aplicação criada foi executado o Lighthouse 3 vezes na página. É recomendável executar mais de uma vez pois podem haver variações de rede e de recursos do dispositivo utilizado. O resultado foi a média do índice de performance dado pela ferramenta.

A máquina utilizada para teste foi um notebook Thinkpad da Lenovo com 16 GB de memória RAM, processador Core i7 vpro da 8ª geração e 254 GB de SSD. Inicialmente, após executar os testes sem ter aplicado nenhuma técnica de performance, o resultado foi de 41 pontos de performance (Figura 7). Sabendo que a nota máxima da ferramenta são 100 pontos, o projeto estava muito abaixo do desejado.

As métricas de performance que a Ferramenta analisa são: *First Contentful paint* (FCP), ou primeira renderização de conteúdo, *Speed Index*, ou índice de velocidade, *Largest Contentful Paint* (LCP), ou maior pintura de conteúdo, *Time to Interactive* (TTI), ou tempo de interatividade, *Total Blocking Time* (TBT), ou tempo total de bloqueio e *Cumulative layout shift* (CLS), ou mudança de layout cumulativa.

Figura 7 - Primeiro resultado do Lighthouse

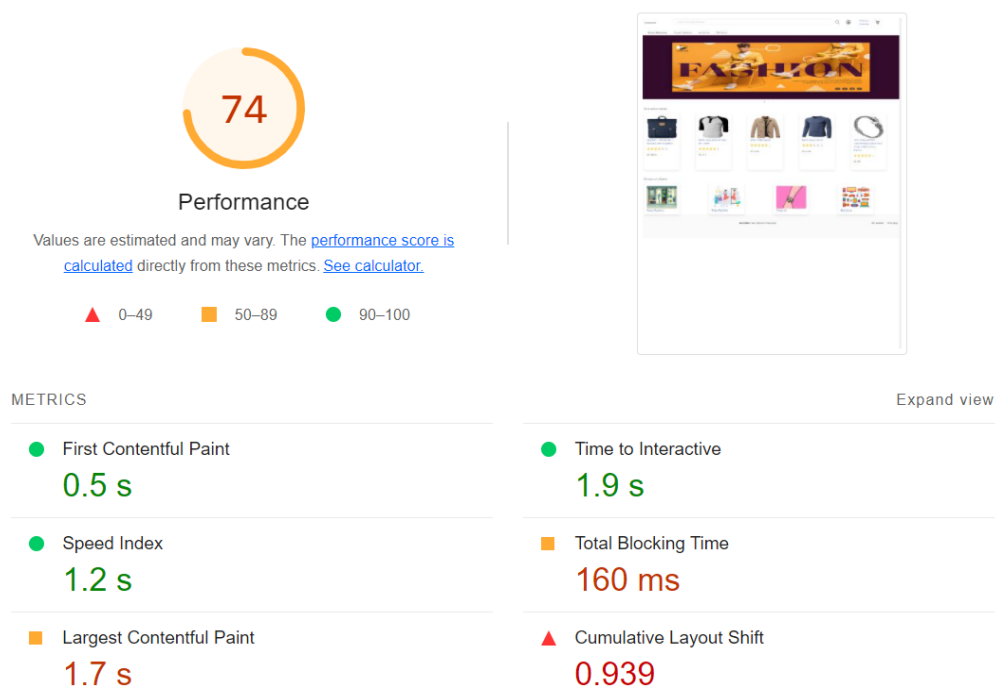


Fonte: Elaborado pelo autor, 2022.

4.2.1 Aplicação do webp para formato das imagens

Primeiramente, foi alterado o formato das imagens para WebP, após isto, foi realizada a auditoria do Lighthouse, rodando três vezes. A Figura 8 logo abaixo apresenta o resultado em média:

Figura 8 - Resultado do Lighthouse após mudança no formato das imagens



Fonte: Elaborado pelo autor, 2022.

Quando comparado a primeira execução do Lighthouse, houve uma melhora nas seguintes métricas:

1. *Speed Index*, houve uma otimização na velocidade em que o conteúdo é exibido durante o carregamento da página;
2. *Largest Contentful Paint*, o tempo do início do carregamento da página até o momento em que o maior bloco de texto ou elemento foi reduzido;
3. *Time to Interactive*, o tempo de interação com a página foi melhorado;
4. *Total Blocking Time*, o tempo que a página fica bloqueada foi diminuído;
5. *Cumulative Layout Shift*, a pontuação de todas as mudanças de layout inesperadas foram aperfeiçoadas.

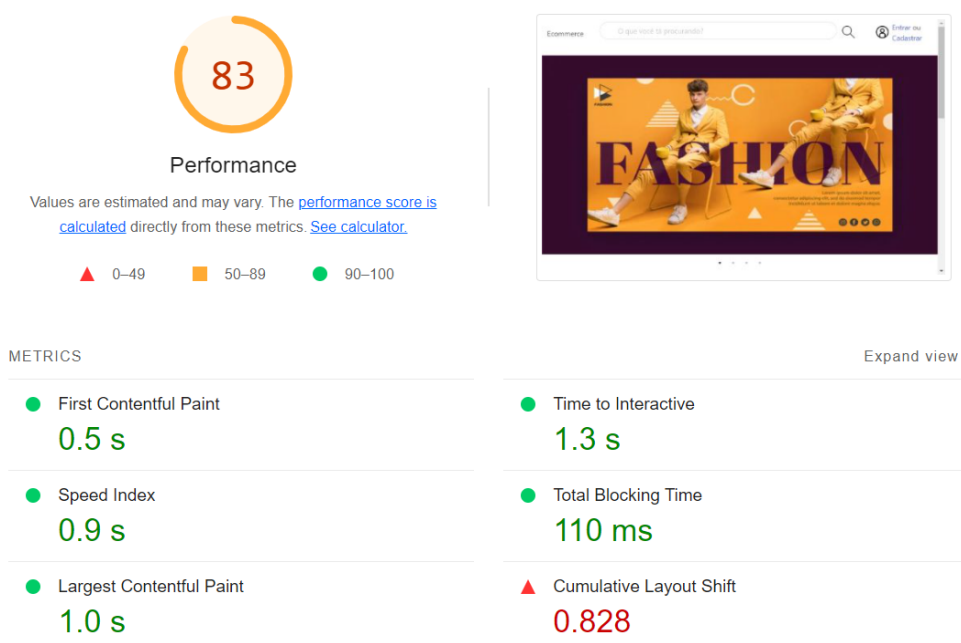
De forma mais expressiva, as melhoras foram no Time to Interactive que foi otimizado em 49% e no Largest Contentful Paint que melhorou 35%.

4.2.2 Redimensionamento das imagens e carregamento lento das imagens

Após a mudança no formato das imagens, foi aplicado o redimensionamento nas imagens da página e também o carregamento lento das imagens. Feito isto, foi

realizada a verificação do Lighthouse, rodando três vezes. O resultado em média é apresentado na figura abaixo (Figura 9):

Figura 9 - Resultado do Lighthouse após dimensionamento correto das imagens



Fonte: Elaborado pelo autor, 2022.

Comparando com o resultado anterior, houve uma melhora significativa nas métricas:

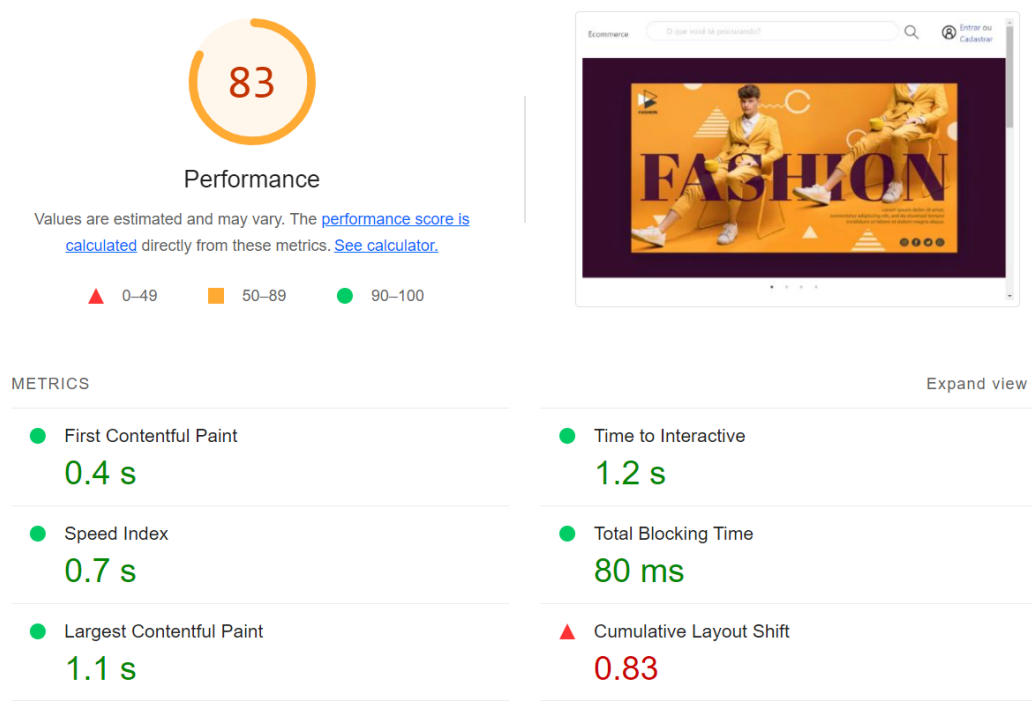
1. *Speed Index*, indicando que a velocidade em que o conteúdo é exibido durante o carregamento da página foi aprimorada;
2. *Largest Contentful Paint*, mostrando que o tempo do início do carregamento da página até o momento em que o maior bloco de texto ou elemento foi otimizado;
3. *Time to Interactive*, apontando que o tempo desde o início do carregamento da página até o momento em que é renderizada foi reduzido.

4.2.3 Carregamento lento dos componentes

Depois da aplicação da técnica de carregamento lento dos componentes, também foi realizada a análise dos componentes. Por ser uma técnica específica e aplicada em recursos menores, os ganhos não foram tão grandes quanto as técnicas anteriores. No entanto, ainda sim houve ganhos nas seguintes métricas:

First Content Paint, Speed Index, Time to Interactive e Total Blocking Time. Sendo destes o mais considerável o *Speed Index*, mostrando que os conteúdos são exibidos mais rapidamente durante o carregamento da página, como visto na Figura 10.

Figura 10 - Resultado do Lighthouse após o Carregamento lento dos componentes

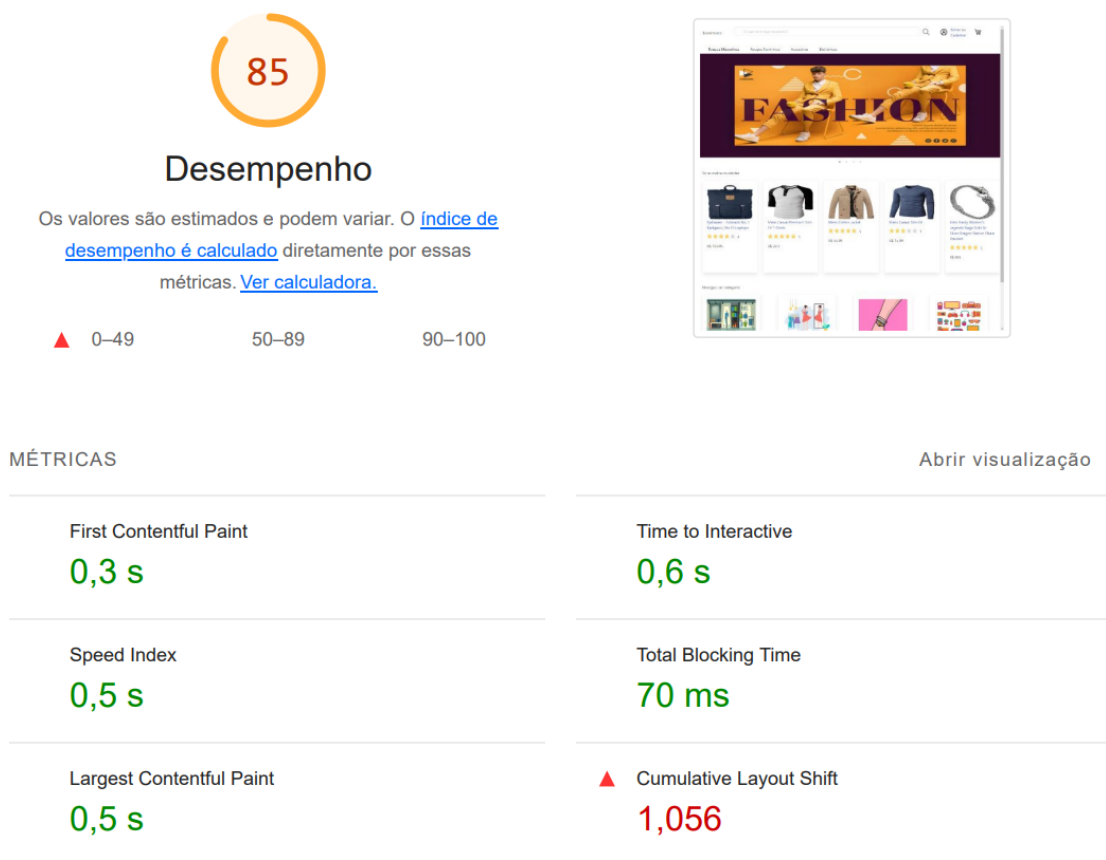


Fonte: Elaborado pelo autor, 2022.

4.2.4 Cacheamento

Em seguida, também houve a aplicação do cacheamento dos arquivos e foi feita a verificação com a Ferramenta. Como mostra a Figura 11, a pontuação de Performance aumentou em 2 pontos, chegando a 85 pontos a média. A melhora mais expressiva foi no tempo de interatividade, indicando que a página ficou disponível para interação mais rapidamente.

Figura 11 - Resultado do Lighthouse após o cacheamento



Fonte: Elaborado pelo autor, 2022.

4.2.5 Comparativo do antes e depois das métricas de performance do Lighthouse

Após a aplicação de todas as técnicas de performance, pôde-se fazer o comparativo de cada métrica de performance da aplicação inicial e da final como mostra a Tabela 1.

Tabela 1 – Comparativo do antes e depois das métricas de performance

Métricas de Performance	Inicial	Final
<i>First Contentful paint</i>	0,5 s	0,3 s
Speed Index	1,5 s	0,5 s
<i>Largest Contentful Paint</i>	4,8 s	0,5 s
Time to Interactive	3,9 s	0,6 s

Total Blocking Time	380 ms	70 ms
Cumulative layout shift	0,964	1,056

Fonte: Elaborado pelo autor, 2022.

5 CONCLUSÃO

A partir do trabalho realizado, é possível concluir que avaliar a performance e desempenho de aplicações web, principalmente de e-commerce, é crucial para o seu êxito. Atualmente, existem ferramentas, como o Lighthouse, por exemplo, que podem auxiliar nas avaliações de velocidade e, com isto, permitir analisar os pontos fortes e fracos do site e trabalhar em cima deles. À medida que o site se torna mais rápido, conseqüentemente os usuários têm um maior interesse pelo que está nele. Uma aplicação que não carrega rapidamente quase nenhum usuário possui interesse.

Estudos mostram que, em cenário de comércio eletrônico, reduzir o tempo de espera, ou seja, melhorar o desempenho da página, resulta em maior conversões em vendas, taxa de retenção maior, melhora nas pesquisas e também no número de cadastros.

Entretanto, pesquisas também apontam que a maioria dos usuários encontram problemas relacionados à performance e usabilidade ao navegar pela internet e que as experiências em dispositivos móveis não são agradáveis. Esses problemas em ambiente de e-commerce impactam diretamente na experiência do usuário, gerando assim uma frustração que não é desejada, além de levar a uma probabilidade menor de o cliente não finalizar o pedido, não retornar a loja para compras futuras e não recomendar para amigos e familiares.

Dessa forma, este trabalho se propôs a resolver os problemas relacionados a performance e usabilidade, melhorando a experiência do cliente dentro do contexto de loja online. Para otimizar a velocidade da página foi aplicadas técnicas de performance, elas foram: Formato das imagens, Redimensionar as imagens, Carregamento lento das imagens, Carregamento lento dos componentes e Cache.

A partir das técnicas aplicadas, foi possível avaliar as seguintes métricas de performance: primeira renderização de conteúdo, maior pintura de conteúdo, tempo de interatividade, tempo total de bloqueio e mudança de layout cumulativa.

Assim, os resultados foram os seguintes: houve uma melhora na renderização de conteúdo, o que fez com que o início do carregamento da página

fosse mais rápido, houve uma otimização na maior pintura de conteúdo (LCP), fazendo o tempo de carregamento dos maiores conteúdos como imagens, por exemplo, se tornasse menor, o tempo de interatividade com a página foi diminuído, ou seja, os usuários conseguem interagir mais rapidamente com a aplicação, também aconteceu uma melhora no tempo de bloqueio e na mudança de layout cumulativa (CLS) e na velocidade com que o conteúdo é exibido visualmente durante o carregamento da página.

Por fim, recomenda-se que trabalhos futuros busquem avaliar as métricas de desempenho também em campo, ou seja, com usuários reais interagindo e utilizando a página no mundo real.

REFERÊNCIAS

Analyze and optimize your website with PageSpeed tools. **Make the web faster.** Disponível em: <<https://developers.google.com/speed?csw=1>> Acesso em: 26 jul. 2022.

ARAÚJO, Laryssa Ayalla Dantas. Análise do modelo de negócios do e-commerce do Sacolão. 2014.

BENDELL, Colin et al. High Performance Images: Shrink, Load, and Deliver Images for Speed. " **O'Reilly Media, Inc.**", 2016.

Bulma: the modern CSS framework that just works. **BULMA.** Disponível em: <<https://bulma.io/>> Acesso em: 02 fev. 2022.

Code Splitting. **webpack.** Disponível em: <<https://webpack.js.org/guides/code-splitting>> Acesso em: 10 mar. 2022.

Driving user growth with performance improvements. **Pinterest Engineering.** Disponível em: <https://medium.com/@Pinterest_Engineering/driving-user-growth-with-performance-improvements-cfc50dafadd7> Acesso em: 26 jul. 2022.

Effect of Website Speed on Users. **Munchweb.** Disponível em: <<http://munchweb.com/effect-of-website-speed>> Acesso em: 23 mai. 2022.

MORAES, Thiago. Estrutura de loja virtual. **Agência e-plus.** Disponível em: <agenciaeplus.com.br/estrutura-de-loja-virtual-funcionalidades>. 2018. Acesso em: 26 jul. 2022.

Fake Store API. Disponível em: <<https://fakestoreapi.com>> Acesso em: 02 fev. 2022.

Find out how you stack up to new industry benchmarks for mobile page speed. **Think with Google.** Disponível em: <thinkwithgoogle.com/marketing-strategies/app-and-mobile/mobile-page-speed-new-industry-benchmarks> Acesso em: 26 jul. 2022.

How can performance improve conversion? **web.dev.** Disponível em: <<https://web.dev/how-can-performance-improve-conversion>> Acesso em: 26 jul. 2022.

How focusing on web performance improved Tokopedia's click-through rate by 35%. **web.dev.** Disponível em: <<https://web.dev/tokopedia/>> Acesso em: 26 jul. 2022.

How Loading Time Affects Your Bottom Line. **NEILPATEL.** Disponível em: <<https://neilpatel.com/blog/loading-time/?wide=1>> Acesso em: 23 mai. 2022.

Is Your Site Driving Customers Away? **Gomez Inc.** Disponível em: <<https://ohmedia.ca/pub/images/blog/Why-Performance-Matters.pdf>> Acesso em: 23 mai. 2022.

KOHAVI, Ron; LONGBOTHAM, Roger. Online experiments: Lessons learned. **Computer**, v. 40, n. 9, p. 103-105, 2007.

LEPAGE, Pete. What is speed?. **web.dev**. Disponível em: <<https://web.dev/what-is-speed>> Acesso em: 26 jul. 2022.

LOPES, João Manuel Brisson. Formatos de imagem. **Computação Gráfica**, v. 3, 2003.

MAKELAINEN, S. From B2B to B2C e-commerce. 15 p. **Masters—University of Helsinki**, Helsinki, 2006.

Mastering Browser Cache. **Vue School.io**. Disponível em: <vueschool.io/articles/vuejs-tutorials/vue-js-performance-mastering-cache> Acesso em: 10 mar. 2022.

MENDES, Laura Zimmermann Ramayana. E-commerce: origem, desenvolvimento e perspectivas. 2013.

MJELDE, Eivind; OPDAHL, Andreas L. Load-time reduction techniques for device-agnostic web sites. **Journal of Web Engineering**, p. 311-346, 2017.

PAVIC, Bojan; ANSTEY, Chris; WAGNER, Jeremy. Why does speed matter?. **web.dev**. Disponível em: <<https://web.dev/why-speed-matters/>> Acesso em: 26 jul. 2022.

PÍCOLO, Mariana Nunes. Otimização de desempenho orientado ao tempo de carregamento em aplicações web front-end. 2021.

RAKOSWISKI, Philip. Mastering Browser Cache. **Vue School.io**. Disponível em: <<https://vueschool.io/articles/vuejs-tutorials/vue-js-performance-mastering-cache>>. 2020. Acesso em: 26 jul. 2022.

RAKOWSKI, Filip. Lazy loading and code splitting in Vue.js. **Vue School.io**. Disponível em: <<https://vueschool.io/articles/vuejs-tutorials/lazy-loading-and-code-splitting-in-vue-js/>>. 2019. Acesso em: 10 mar. 2022.

Service worker overview. **Chrome Developers**. Disponível em: <<https://developer.chrome.com/docs/workbox/service-worker-overview/>> Acesso em: 26 jul. 2022.

Speed Index. **web.dev**. Disponível em: <https://web.dev/speed-index/?utm_source=lighthouse&utm_medium=devtools> Acesso em: 26 jul. 2022.

TASSABEHJI, Rana. Applying e-commerce in business. **Applying E-Commerce in Business**, p. 1-326, 2003.

The Cache API: A quick guide. Disponível em:
<<https://web.dev/cache-api-quick-guide/>> Acesso em: 21 jun. 2022.

Using site speed in web search ranking. **Google Search Central**. Disponível em:
<<https://developers.google.com/search/blog/2010/04/using-site-speed-in-web-search-ranking>> Acesso em: 23 mai. 2022.

Using site speed in web search ranking. **Google Search Central**. Disponível em:
<<https://developers.google.com/search/blog/2010/04/using-site-speed-in-web-search-ranking>> Acesso em: 26 jul. 2022.

WAGNER, Jeremy. Web Performance in Action: Building Fast Web Pages. **Simon and Schuster**, 2016.

WALTON, Philip. User-centric performance metrics. **web.dev**. Disponível em:
<<https://web.dev/user-centric-performance-metrics>> Acesso em: 26 jul. 2022.

What is ecommerce? **AMAZON**. Disponível em:
<<https://sell.amazon.com/learn/what-is-ecommerce>> Acesso em: 02 fev. 2022.

What is Nuxt? **Nuxt3**. Disponível em: <<https://v3.nuxtjs.org/concepts/introduction>>
Acesso em: 02 fev. 2022.

What is web performance? **MDN web docs**. Disponível em:
<https://developer.mozilla.org/en-US/docs/Learn/Performance/What_is_web_performance> Acesso em: 26 jul. 2022.

Why Web Performance Matters:
YSlow. Disponível em: <<http://developer.yahoo.com/yslow>> Acesso em: 26 jul. 2022.