



**UNIVERSIDADE ESTADUAL DA PARAÍBA
CAMPUS VII – GOVERNADOR ANTÔNIO MARIZ
CENTRO DE CIÊNCIAS EXATAS E SOCIAIS APLICADAS
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

JOÃO PAULO DE LIMA

**ESTUDO COMPARATIVO ENTRE FRAMEWORKS FRONT-END PARA
DESENVOLVIMENTO DE PÁGINAS WEB**

**PATOS – PB
2019**

JOÃO PAULO DE LIMA

**ESTUDO COMPARATIVO ENTRE FRAMEWORKS FRONT-END PARA
DESENVOLVIMENTO DE PÁGINAS WEB**

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

Área de concentração: Programação Web

Orientador: Prof.Esp. Allyson Jeronimo Dantas

**PATOS - PB
2019**

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.

L732e Lima, Joao Paulo de.
Estudo comparativo entre frameworks front-end para desenvolvimento de páginas web [manuscrito] / Joao Paulo de Lima. - 2019.
49 p. : il. colorido.
Digitado.
Trabalho de Conclusão de Curso (Graduação em Computação) - Universidade Estadual da Paraíba, Centro de Ciências Exatas e Sociais Aplicadas, 2019.
"Orientação : Prof. Esp. Allyson Jeronimo Dantas ,
Coordenação do Curso de Computação - CCEA."
1. Framework. 2. Front-end. 3. Responsividade. 4.
Website. I. Título
21. ed. CDD 005.1

João Paulo de Lima

**ESTUDO COMPARATIVO ENTRE FRAMEWORKS FRONT-END PARA
DESENVOLVIMENTO DE PÁGINAS WEB**

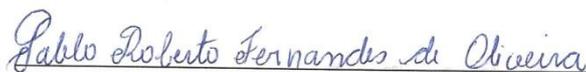
Trabalho de Conclusão de Curso apresentado ao Curso de Bacharelado em Ciências da Computação da Universidade Estadual da Paraíba, em cumprimento à exigência para obtenção do grau de Bacharel em Ciência da Computação.

Aprovado em 27/11/19

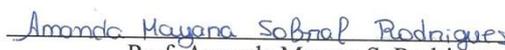
BANCA EXAMINADORA



Prof. Me. Allyson Jerônimo Dantas
(Orientador)



Prof. Me. Pablo Roberto F. de Oliveira
(Examinador)



Prof. Amanda Mayara S. Rodrigues
(Examinadora)

AGRADECIMENTOS

A Deus, por me proporcionar a experiência do conhecimento que me fez chegar onde estou.

A minha mãe, Maria do Bom Sucesso de Lima, por ter proporcionado a melhor educação que ela poderia me propor e que sempre se doou para que meus estudos fosse peça fundamental para nosso futuro.

Ao meu pai, Paulo da Silva Almeida por todo apoio e suporte que foi dado a partir do momento em que nos encontramos.

Ao meu orientador Prof. Allyson Jerônimo, por ter me proporcionado a experiência de concluir este trabalho sempre me guiando da melhor forma possível e sempre resolvendo minhas dúvidas de maneira clara.

A Michelle da Silva Sousa, uma pessoa incrível que marcou minha vida e sempre me deu todo apoio necessário nessa caminhada acadêmica, sempre me motivando e apontando os melhores caminhos para o meu crescimento acadêmico.

Aos amigos, que fiz durante toda essa jornada acadêmica nos quais quero leva-los pra vida, em especial Rafaelly Ferreira e Matheus Paulino, que sempre estiveram do meu lado, proporcionando os melhores dias vividos na universidade. Entre outros grandes amigos, Danielly Gualberto, Kaique Rijkaard, Ricardo Farias, Marinaldo Silva, Pedro Nunes, Luis Henrique, Marquinhos e Kaubir por todo o companheirismo, apoio e suporte que me deram. E um agradecimento especial a Geovane (Joe), amigo que nos deixou ao longo dessa jornada.

Aos professores que sempre entregaram aos alunos toda sua dedicação para formar grandes profissionais futuros.

E a todas as pessoas que integram a Universidade Estadual da Paraíba.

RESUMO

A expansão de sistemas web cresceu junto com avanço da tecnologia, crescimento esse que é um dos grandes contribuintes para a descentralização da informação e inclusão digital, permitindo que muitos meios de comunicação e mídias de exposição de informação migrassem para o ambiente tecnológico atualmente existente. O número de requisições para desenvolvimento de sites é amplamente maior que há anos atrás, gerando as empresas de desenvolvimento uma necessidade de construção e entrega dos seus serviços de maneira mais prática e melhor customizada. Os frameworks front-end foram criados para reduzir o tempo de desenvolvimento de um sistema web e para uma customização de interface mais sofisticada para o usuário final. Este trabalho apresenta uma análise comparativa entre frameworks front-end ricos em oferecer responsividade para páginas web, com o objetivo de avaliar o comportamento dos mesmos em um cenário que possa abstrair o máximo de recursos dos mesmos. Os frameworks Bootstrap e Materialize foram estudados e analisados considerando-se os seguintes critérios de avaliação: responsividade, compatibilidade entre navegadores, esforço de implementação, documentação e linhas de código. Para realização da análise foi desenvolvida uma mesma página web nos dois frameworks estudados, a fim de analisar como foi a implementação do código em ambas as ferramentas. Os resultados apontam uma leve vantagem de customização e organização de código por parte do Bootstrap.

Palavras-Chave: Framework. Front-end. Responsividade.

ABSTRACT

The expansion of web systems has grown along with technology advancement, which is a major contributor to the decentralization of information and digital inclusion, enabling many media and information exposure media to migrate to the existing technology environment. The number of requests for website development is vastly greater than years ago, creating development companies a need to build and deliver their services in a more practical and better tailored manner. Front-end frameworks are designed to reduce development time for a web system and for more sophisticated end-user interface customization. This paper presents a comparative analysis between front-end frameworks rich in providing responsiveness to web pages, with the objective of evaluating their behavior in a scenario that can abstract their maximum resources. Bootstrap and Materialize frameworks were studied and analyzed considering the following evaluation criteria: responsiveness, browser compatibility, implementation effort, documentation and lines of code. To perform the analysis was developed the same web page in both frameworks studied, in order to analyze how was the implementation of the code in both tools. The results point to a slight advantage of Bootstrap customization and code organization.

Keywords: Framework. Front-end. Responsividade.

LISTA DE ILUSTRAÇÕES

Figura 1 – Interface do navegador Mosaic	16
Figura 2 - Representação do modelo cliente/servidor aplicado a Web	18
Figura 3 - Modelo de requisição e resposta do HTTP	19
Figura 4 - Marcações HTML.....	22
Figura 5 - Escopo padrão de uma página HTML.....	23
Figura 6 - Documento HTML sendo estilizado com CSS	25
Figura 7 - programa em JavaScript para calcular e imprimir fatoriais	26
Figura 8 - Estrutura do Bootstrap	30
Figura 9 - Estrutura do Materialize	31
Figura 10 - Cabeçalho do site.	37
Figura 11 - Coleção feminina do site.....	38
Figura 12 - Novos produtos e serviços.....	38
Figura 13 - Rodapé do Athletic Shoes.....	39
Figura 14 - Versão do site para dispositivos móveis.	39
Figura 15 - Erros de renderização de imagem no Internet Explorer.....	41
Figura 16 - Interface para login de usuário criada apenas no Bootstrap.	43

LISTA DE TABELAS

Tabela 1 - Tabela referente às informações de responsividade dos frameworks.....	40
Tabela 2 - Resultado sobre compatibilidade dos navegadores com os frameworks.	41
Tabela 3 - Tabela de implementação CSS.....	42
Tabela 4 - Tabela sobre disponibilidade de documentação.	43
Tabela 5 - Linhas de código escritas por cada Framework.	44

LISTA DE ABREVIATURAS E SIGLAS

CERN	Conseil Européen pour la Recherche Nucléaire
CRUD	Create Read Update Delete
CSS	Cascading Style Sheets
DNS	Domain Name System
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
URL	Uniform Resource Locator

SUMÁRIO

1	INTRODUÇÃO	11
1.1	Problemática	12
1.2	Justificativa	12
1.3	Objetivos	13
1.3.1	<i>Objetivo Geral</i>	13
1.3.2	<i>Objetivos Específicos</i>	13
1.4	Metodologia	13
2	FUNDAMENTAÇÃO TEÓRICA	15
2.1	Word Wide Web	15
2.2	Arquitetura Cliente-Servidor	17
2.2.1	<i>Cliente</i>	18
2.2.2	<i>Servidor</i>	20
2.3	Front-End	21
2.3.1	<i>HTML</i>	22
2.3.2	<i>CSS</i>	24
2.3.3	<i>JavaScript</i>	25
2.4	Back-End	27
2.5	Frameworks	27
2.6	Frameworks Front-End	28
2.6.1	<i>Bootstrap</i>	29
2.6.2	<i>Materialize</i>	30
2.7	Trabalhos Relacionados	31
3	METODOLOGIA	33
3.1	Métricas de Avaliação	33
3.1.1	<i>Responsividade</i>	34
3.1.2	<i>Compatibilidade entre navegadores</i>	34
3.1.3	<i>Esforço de implementação</i>	34
3.1.4	<i>Suporte e documentação</i>	34
3.1.5	<i>Linhas de Código</i>	35
3.2	Estudo Experimental	36

3.2.1	<i>Cenário de Avaliação</i>	36
3.2.2	<i>Desenvolvimento do Site</i>	36
3.2.3	<i>Protótipo</i>	36
4	RESULTADOS E DISCUSSÕES	40
4.1	Responsividade	40
4.2	Compatibilidade entre navegadores	40
4.3	Esforço de implementação	42
4.4	Suporte e documentação	43
4.5	Linhas de código	44
5	CONCLUSÃO	45
	REFERÊNCIAS	46

1 INTRODUÇÃO

O surgimento de novas tecnologias de uma maneira geral facilitou a forma como o ser humano vive hoje. A tecnologia se faz presente em praticamente tudo, ou seja, existe sempre algo tecnológico vinculado a alguém, por exemplo, a ação de realizar um exame de raio x, ou o uso de dispositivos móveis, a facilidade de usar a internet como meio de comunicação, sendo agilidade e praticidade uns dos mais relevantes benefícios que a mesma proporciona. Visto isto, é coerente afirmar que o desenvolvimento de sites e aplicações web cresceu em paralelo com a tecnologia.

A necessidade de aliar tempo e qualidade fez com que inúmeras empresas do ramo tecnológico desenvolvessem frameworks (ferramentas) que contribuíssem para que o desenvolvedor fizesse uso de uma alta gama de recursos e praticidades, permitindo ao mesmo uma redução considerável no tempo de desenvolvimento de uma aplicação Web, e na vasta oportunidade de otimização da mesma.

Um framework é uma coleção integrada de elementos de software reusáveis e extensíveis para um domínio específico da aplicação e que podem ser habilitados para atender a uma aplicação em particular. Os frameworks são antigos, datando do final da década de 80, mas apenas recentemente vêm sendo amplamente empregados devido à sua integração com os padrões de projeto (SANCHES, 2005).

No desenvolvimento de aplicações para web é fundamental a compreensão da arquitetura cliente-servidor, modelo que consiste na separação da aplicação em máquinas diferentes e que se comunicam para troca de informações por meio do protocolo HTTP. Na arquitetura cliente-servidor o processamento de informação é dividido em módulos ou processos diferentes, um dos processos terá a responsabilidade de executar a manutenção da informação, processo esse chamado de servidor, enquanto o processo cliente será responsável pela obtenção dos dados (BATISTI, 2001).

A utilização de frameworks no desenvolvimento front-end, tem um papel de suma importância, haja vista, entregar ao desenvolvedor web inúmeras bibliotecas e evitar a prática de repetição de código. Outro fator importante é a quantidade de frameworks existentes no cenário atual, possibilitando ao mesmo o conhecimento de uma vasta quantidade de opções que irão se adequar de acordo com a necessidade de criação e qual aplicabilidade o framework trará para a página.

1.1 Problemática

O cenário atual de desenvolvimento Web exige muito tempo ao desenvolvedor, muitas das vezes o mesmo precisa atuar tanto no desenvolvimento front-end quanto no back-end, o tornando um desenvolvedor full stack.¹

Faz-se presente a necessidade de tecnologias que contribuam no processo de desenvolvimento e que permitam práticas para uma boa estruturação do front-end, frameworks que utilizem os mais aplicáveis padrões de projeto e que entreguem portabilidade necessária para um desenvolvimento ágil e coeso.

Esse trabalho visa conhecer os frameworks front-end mais utilizados atualmente e investigar sobre a relevância de três dos mais populares frameworks de desenvolvimento Web. Dadas situações anteriores, se torna importante entrar na problemática do quanto o framework front-end influencia num desenvolvimento de uma aplicação web, quais frameworks mais aconselháveis a se utilizar e quais conseguem suprir as necessidades do desenvolvedor.

1.2 Justificativa

O mercado de desenvolvimento web avança a passos largos, devido à necessidade de expandir o uso da tecnologia em empresas, nas mais diversas áreas de atuação das mesmas. Tal crescimento aumentou exponencialmente a busca por profissionais capacitados de TI.

De acordo com a pesquisa realizada por Umblar (2017), entre as 1224 vagas ofertadas pela plataforma trampos.co, 1023 eram destinadas ao desenvolvimento web e o front-end dito como um dos conhecimentos mais requisitados por empresas de desenvolvimento Web.

Outro fator importante é que cenário atual exige que os sites tenham o mais alto nível interação entre o cliente e usuário, se fazendo importante que as aplicações web ofereçam interfaces amigáveis, robustas, visualmente agradáveis para o usuário, e proporcionar essas características aos usuários usando diretamente as tecnologias nativas (HTML, CSS e JavaScript) , leva muito mais

¹ Desenvolvedor full stack é aquele que trabalha tanto no front-end como no back-end de uma aplicação.

tempo e trabalho ao desenvolvedor, além de requerer do desenvolvedor um conhecimento muito avançado na área de design.

Os frameworks irão contribuir consideravelmente facilitando o desenvolvimento de componentes de interfaces, amigáveis e robustas, porém, de maneira mais rápida e prática.

De modo geral, os objetivos principais de se utilizar um framework advêm da modularidade, reusabilidade, extensibilidade e inversão de controle que ele fornece aos desenvolvedores (FAYAD & SCHMIT, 1997, *apud* PRATI, 2003).

1.3 Objetivos

Será posto em pauta os objetivos gerais e específicos deste trabalho, com a finalidade de demonstrar o seguimento no qual as tarefas serão executadas.

1.3.1 Objetivo Geral

Realizar um estudo comparativo entre os mais utilizados frameworks front-end voltado a desenvolvimento web.

1.3.2 Objetivos Específicos

- Realizar um estudo a respeito dos frameworks voltados ao desenvolvimento Web;
- Desenvolver um cenário no qual será especificado um projeto para ser implementado por cada um dos frameworks;
- Realizar uma análise comparativa entre os frameworks estudados.

1.4 Metodologia

A fase inicial deste trabalho teve foco na realização da pesquisa bibliográfica para fundamentação teórica das metas de estudo, será produzida uma revisão literária a respeito do desenvolvimento Web, O funcionamento da arquitetura cliente-servidor, bem como um levantamento dos principais frameworks front-end tendo

como base dados do GitHub².

Subsequentemente será realizado um comparativo entre os frameworks front-end, elencados na seguinte pesquisa.

² Plataforma de hospedagem de código-fonte com controle de versão usando o **Git**.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são descritos conceitos essenciais que são usados para o desenvolvimento dessa pesquisa, bem como são postos conceitos importantes para melhor domínio sobre o estudo.

2.1 Word Wide Web

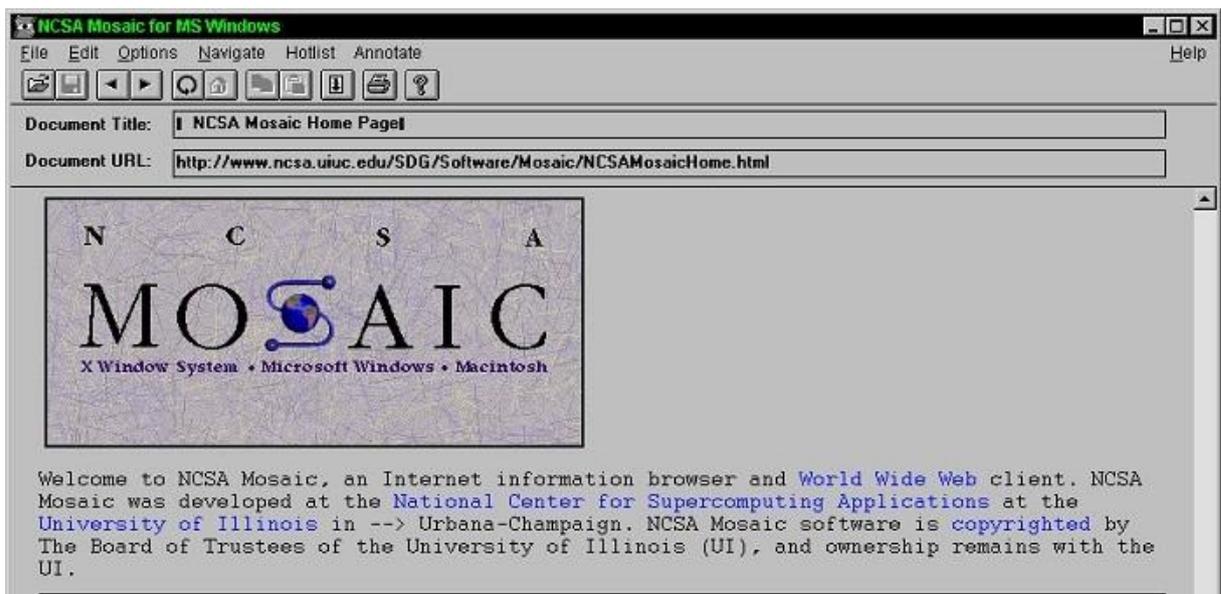
A descentralização da informação e o constante crescimento de novas tecnologias estão muito associados ao quanto à internet deixa seus usuários mais próximos e acessíveis, quebrando muita das vezes a barreira da distância e levando o termo comunicação a um novo patamar. Toda essa comodidade faz com que veículos de comunicação e informação que antes atuavam de maneira física, se colocassem na necessidade de se adequar a essa nova perspectiva na qual o mundo se encontra. Com isso, é bastante comum, hoje, o ser humano não comprar mais jornal ou acompanhar as notícias do dia através do antigo rádio convencional.

Outro exemplo notável é o quanto mais se substitui a TV por plataformas online de entretenimento. Visto isso, é fácil notar que a requisição e a necessidade de desenvolvimento de sites robustos e otimizados crescem a cada instante, tornando o desenvolvimento web uma das áreas da computação que mais necessita de praticidade e ferramentas que facilitem tais necessidades.

Tanenbaum (2003, p.462). define a World Wide Web como “uma estrutura arquitetônica que permite o acesso a documentos vinculados espalhados por milhões de máquinas na Internet”. A Web a princípio atuava como um meio de distribuição de dados sobre física, sendo a distância um fator que influenciava na comunicação dos cientistas que trabalhavam em um mesmo projeto, mas de lugares distintos, a Web permitia que os cientistas pudessem compartilhar informações e suas colaborações em projetos. Um grande contribuinte para a popularização da mesma foi a forma que uma interface bem trabalhada somada a distribuição de informação de maneira mais rápida acontecia. Atualmente é possível obter informações sobre praticamente tudo que se deseja saber através dos mais variados sites, tornando a internet um ambiente didático, substituindo muita das vezes a necessidade de um indivíduo se deslocar de sua residência para obter informações em bibliotecas ou escolas (TANENBAUM, 2003).

É importante citar alguns acontecimentos que contribuíram para o que se entende como navegar na internet. Para navegar na internet e obter todos os recursos possíveis se faz necessário o uso de um software que interprete as linguagens necessárias e que mostre para o usuário o resultado final, esteticamente interessante e que seja de fácil compreensão, esse software é chamado de Browser, do português Navegador. O primeiro Browser foi criado pelo físico do CERN Tim Berners-Lee onde se fazia o uso de Hipertexto para compartilhar informações. O primeiro navegador foi chamado então de WorldWideWeb, onde era basicamente um navegador em que eram exibidos textos. Porém, o primeiro navegador despertou em alguns cientistas o interesse em intensificar pesquisas a respeito dos navegadores e a busca por o desenvolvimento de um software melhor trabalhado, daí, um pesquisador chamado Marc Andreessen, da University of Illinois, desenvolveu o primeiro navegador gráfico, o Mosaic, que se tornou tão popular que, logo, Marc Andreessen criou sua própria empresa, uma das mais ascendentes na época, a Netscape Communications Corp (TANENBAUM. 2003). A imagem a seguir apresenta a interface do navegador Mosaic.

Figura 1 – Interface do navegador Mosaic



Fonte: Brito (2014).

2.2 Arquitetura Cliente-Servidor

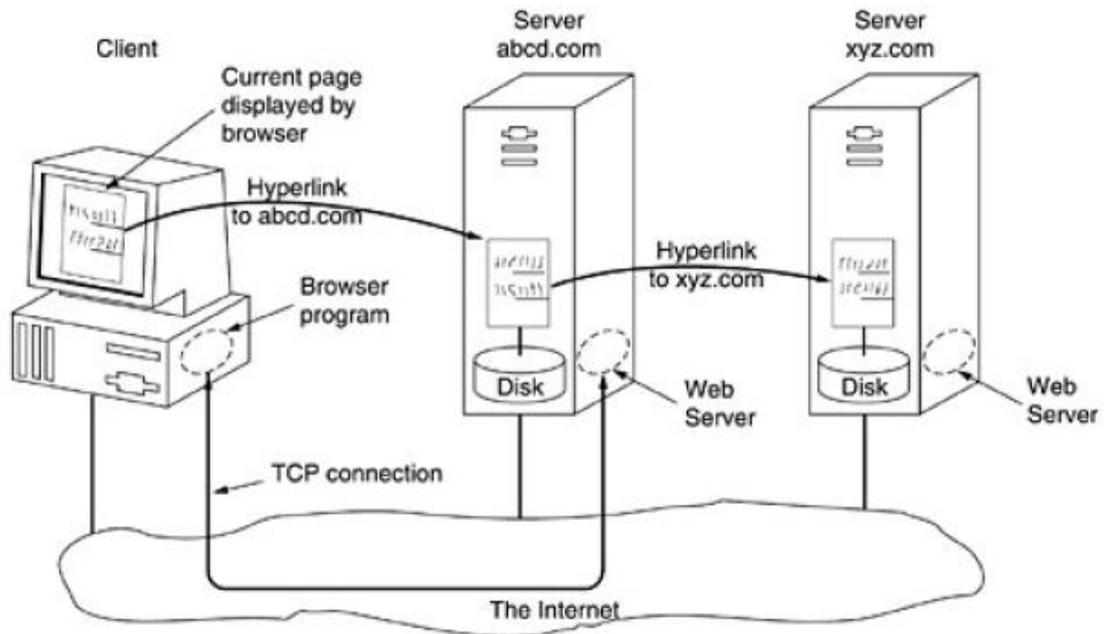
Para falar a respeito de como a web se comporta é importante entender como funciona o modelo Cliente/Servidor e como um site conversa com o usuário que está acessando o mesmo.

A arquitetura Cliente/Servidor é uma arquitetura onde o processamento da informação é dividido em módulos ou processos distintos, um processo responsável pela manutenção da informação (Servidor), enquanto que outro processo é responsável pela obtenção dos dados (Cliente). BATTISTI, (2001, p.38).

Sempre que o usuário executa um processo em um determinado site, esse processo se comunica com um servidor, o servidor atende às solicitações do usuário e as entrega como resposta. Essa constante requisição e entrega de informações compõe a arquitetura cliente/servidor.

O modelo Cliente/Servidor é uma abordagem que tem foco na divisão dos processos em plataformas independentes que se comunicam, possibilitando que os recursos sejam compartilhados enquanto se ganha o máximo de benefício de cada dispositivo diferente, logo, a arquitetura Cliente/Servidor é um modelo lógico (VASKEVITCH, 1995). A Figura 2 mostra a representação do modelo cliente-servidor.

Figura 2 - Representação do modelo cliente/servidor aplicado a Web



Fonte: Tanenbaum (2003).

A **Figura 2** apresenta todo o processo de requisição e resposta que ocorre na arquitetura cliente/servidor, onde toda parte visual, no tocante a interface que o navegador disponibiliza ao usuário é dita como cliente, e toda solicitação e gerenciamento de informação fica a responsabilidade do servidor.

2.2.1 Cliente

É possível entender como uma página web funciona a partir da compreensão do lado do cliente. De um modo geral, qualquer usuário que utilize um computador que esteja conectado a internet é considerado como um cliente, logo, o navegador que esteja hospedado em uma determinada máquina, consiste em um programa cliente. O navegador é o responsável por requisitar a informação ao servidor que responde entregando os dados na tela do usuário.

O protocolo HTTP (HyperText Transfer Protocol) compõe um conjunto de protocolos de comunicações TCP/IP (Transmission Control Protocol/Internet Protocol), este protocolo é responsável por toda troca de informações que o cliente terá com um servidor web.

O HTTP — Protocolo de Transferência de Hipertexto (HyperText Transfer Protocol) —, o protocolo da camada de aplicação da Web, está no coração da Web e é definido no [RFC 1945] e no [RFC 2616]. O HTTP é executado em dois programas: um cliente e outro servidor. Os dois, executados em sistemas finais diferentes, conversam entre si por meio da troca de mensagens HTTP. O HTTP define a estrutura dessas mensagens e o modo como o cliente e o servidor as trocam KUROSE (2007, p. 72).

É possível ver na figura seguinte como o HTTP funciona dentro do modelo cliente-servidor.

Figura 3 - Modelo de requisição e resposta do HTTP



Fonte: Kurose (2007).

A **Figura 3** mostra a função do HTTP dentro de uma arquitetura cliente/servidor onde o HTTP permite a comunicação e troca de informações entre o cliente e o servidor da arquitetura.

Uma página web permite inúmeras formas de interação com o usuário, uma dessas interações é a permissão do mesmo ter a possibilidade de com um mouse clicar em diferentes seções de um site e ser redirecionada a outra página correspondente. De modo mais simples, imaginando um site de notícias, é comum que todo o noticiário do site não esteja presente em uma única página, onde o normal é que apenas os destaques das notícias estejam presentes na página principal do site, desse modo, o usuário escolhe a informação que deseja visualizar e ao clicar na informação uma nova página será aberta, seja na mesma aba de navegação, seja em uma aba seguinte. Para tais ações serem possíveis é preciso que o hiperlink vinculado a página de algum modo possa nomear essa nova página que será exibida ao usuário final.

As URLs (Uniform Resource Locators) são responsáveis por nomear páginas web. Em suma, a URL é composta por: nome do protocolo (HTTP), DNS da máquina

em que a página está situada (www.google.com) e o nome do arquivo na qual a página está contida (nome.html). Sempre que o usuário clicar em um hiperlink uma série de ações será executadas para que o link selecionado possa ser redirecionado para uma nova página (TANEBAUM, 2003). Considerando que o usuário optou por fazer uma pesquisa em um determinado site de busca a respeito de telefonia na web e foi indicado a Home Page da ITU, <http://www.itu.org/home/index.html>. Conforme Tanenbaum (2003, p.465) as seguintes etapas irão acontecer:

1. O navegador determina o URL (verificando o que foi selecionado).
2. O navegador pergunta ao DNS qual é o endereço IP de www.itu.org.
3. O DNS responde com 156.106.192.32.
4. O navegador estabelece uma conexão TCP com a porta 80 em 156.106.192.32.
5. Em seguida, o navegador envia um comando solicitando o arquivo /home/index.html.
6. O servidor www.itu.org envia o arquivo /home/index.html.
7. A conexão TCP é encerrada.
8. O navegador exibe todo o texto de /home/index.html.
9. O navegador busca e exibe todas as imagens que o arquivo contém.

Para finalizar a compreensão por parte do cliente, ao encerrar todos os processos anteriores, para que a página web possa ser visualizada pelo usuário, o navegador precisa interpretar o formato da página que será exibida. Levando em consideração que existem inúmeros navegadores no cenário atual, e para permitir que todos tenham a mesma compreensão de uma página, as páginas web são escritas e uma linguagem que permita que qualquer página possa ser lida independente do navegador que está interpretando a mesma. A linguagem HTML (Hypertext Markup Language) é a responsável escrever as páginas e permitir que o navegador possa mostrá-las para o usuário final (TANENBAUM). A linguagem HTML será abordada posteriormente neste trabalho.

2.2.2 Servidor

Compreendido o comportamento do cliente e a forma como ele comunica-se com o servidor, todo o processo de comunicação entre essa arquitetura se encerra

com o servidor atendendo às solicitações do cliente. Em um contexto geral, na arquitetura cliente/servidor há sempre um hospedeiro em funcionamento, denominado como servidor, o mesmo é responsável por atender as requisições de muitos outros hospedeiros, ditos como clientes (KUROSE, 2007). Em uma aplicação Web um servidor Web sempre estará em funcionamento atendendo tudo que é requisitado pelo navegador, tal situação torna um único servidor hospedeiro muito das vezes incapaz de atender todas as solicitações do navegador, onde, se faz necessário que seja arquitetado datacenters onde os mesmos teriam a capacidade de alojar uma quantidade considerável de servidores hospedeiros, possibilitando que o servidor Web torne se mais robusto e possa atender, sites que cobrem muito do servidor, como um site de stream de vídeos ou músicas (KUROSE, 2007).

Anteriormente foram mostradas as etapas que ocorrem quando o usuário por meio do navegador clica em um determinado hiperlink, tais ações que são de responsabilidade do cliente são respondidas pelo servidor segundo Tanenbaum (2003, p.468) da seguinte maneira.

1. Aceitar uma conexão TCP de um cliente (um navegador).
2. Obter o nome do arquivo solicitado.
3. Obter o arquivo (do disco).
4. Retornar o arquivo ao cliente.
5. Encerrar a conexão TCP.

Em um contexto geral essas são as características mais importantes a respeito de um servidor web, sendo o mesmo ponto chave do modelo cliente/servidor por ser o responsável por todo o trabalho de entrega de requisições ao cliente.

2.3 Front-End

Será descrito a partir deste ponto, as tecnologias que tornam possíveis a interação de interface entre o usuário e o site, de que forma uma página web é exibida, como e quais linguagens permitem tais circunstâncias.

No campo do desenvolvimento web é comum proferir a respeito de programação Front-End, de maneira compreensível, falar de Front-End é

caracterizar quais ferramentas ou linguagem promovem a visualização de um site ao usuário. A seção seguinte irá abordar sobre a Linguagem HTML, CSS e JavaScript.

2.3.1 HTML

“HTML é uma linguagem de marcação criada para a publicação de hipertexto na *web*. Seus marcadores ou *tags* apoiam a estruturação de documentos em títulos, subtítulos, parágrafos, listas, tabelas, *hiperlinks*, etc.” (MELO, 2007, p. 26). A linguagem HTML (HyperText Markup Language) é dita como uma linguagem de marcação de texto. Brooks (2007) trata a linguagem como uma linguagem de marcação que tem como finalidade especificar a estrutura de um documento. A vantagem do HTML ser uma linguagem de marcação em comparativo com linguagens que não tenham essa característica é a maior facilidade para criar um navegador destinado à linguagem de marcação: o navegador só precisa entender os comandos de marcação (Tanenbaum, 2003). A Figura 4 ilustra como funciona um escopo básico de marcações HTML para criação de uma página Web.

Figura 4 - Marcações HTML

```
1
2 <!DOCTYPE HTML>
3 <html lang="pt-br">
4   <head>
5     <title>exemplo</Title>
6   </head>
7   <table border>
8     <tr>
9       <td>Itens/Mês</td>
10      <th>Janeiro</th><th>Fevereiro</th><th>Março</th>
11    </tr>
12    <tr>
13  </tr>
14  <tr>
15    <th>Linhas</th><td>3</td><td>3</td><td>5</td>
16  </tr>
17 </table>
18
```

A estrutura de um documento HTML consiste no agrupamento de comandos (tags) que são essenciais para a escrita da linguagem em um editor de texto. Logo, é preciso definir o escopo da estrutura que é determinado através das tags <html> e </html>, as tags anteriores são responsáveis por definir o tamanho do documento, onde, a tag de fechamento </html> delimita o fim da estrutura. Em seguida é definido o cabeçalho da página, para isso, são usadas as tags <head> e </head>, por fim, o corpo da estrutura é definido, responsável por conter o conteúdo que é visualizado pelo usuário. Dentre esta estrutura, várias outras tags são utilizadas. Na Figura 5 é possível ver a estrutura base para utilização do HTML.

Figura 5 - Escopo padrão de uma página HTML

```
01 <!DOCTYPE html>
02 <html>
03 <head>
04   <meta charset="UTF-8"/>
05   <title>Document</title>
06 </head>
07 <body>
08   <!-- Conteúdo -->
09 </body>
10 </html>
```

Fonte: DEVMEDIA (2010).

Segundo Korpela (2005) a diferença entre a linguagem HTML e as linguagens de programação é que, estas contêm instruções que transformam a informação, em contra partida, a linguagem HTML e demais linguagens de marcação apenas estruturam a informação.

O navegador Web é o responsável por decifrar a escrita HTML e transformar toda a marcação em um ambiente operacional para o usuário, propondo a interação entre o usuário e a linguagem, transformando todo o código escrito em HTML em uma interface que seja de simples compreensão para o mesmo. Visto isso, o HTML permite ao usuário visualização de vídeos online, acesso a redes sociais, e sites em geral. Claro, somado ao HTML as linguagens de estilo CSS e o JavaScript considerada como uma linguagem dinâmica, compõem o Front-End de um site e proporcionam todas as possibilidades já citadas.

Uma das praticidades pra quem escreve em HTML é que por mais que existam editores focados nas linguagens de marcação é possível desenvolver uma

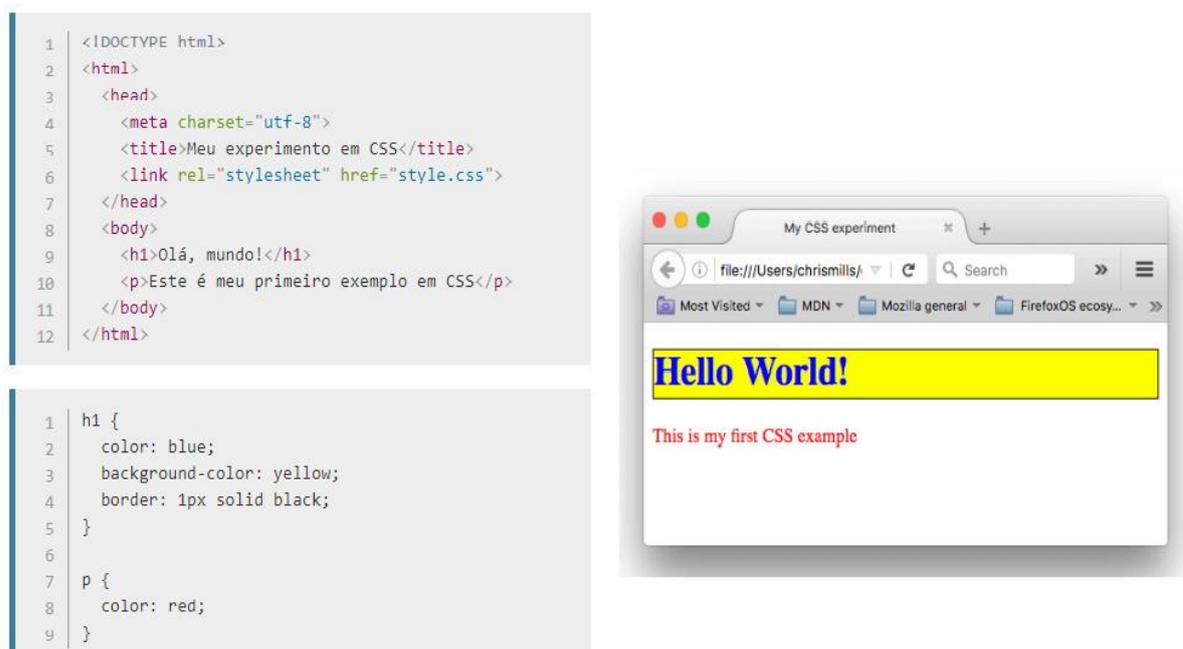
página Web com um editor padrão de texto, como o bloco de notas, presente em qualquer navegador. De todo modo, os editores de texto dedicados ao desenvolvimento web como Sublime Text, Visual Studio Code e Notepad++ são comumente utilizados.

2.3.2 CSS

O CSS (Cascading Style Sheets) do português Folha de Estilo em Cascata é mais uma linguagem que compõem o Front-End de uma página Web. O CSS tem como principal característica dar estilo a páginas HTML ou outros tipos de documentos voltados a Web, transformando toda a formatação crua em HTML numa página bem elaborada. Para Grannel (2007) é uma linguagem de estilo que se caracteriza por especificar os aspectos de vários elementos que foram definidos em uma linguagem de marcação, de forma isolada. O CSS vai definir como o documento escrito em HTML será representado para o usuário, permitindo uma gama de possibilidades de estilização como textos, imagens, templates, etc.

Para maior clareza, supondo que determinada pessoa está escrevendo algum documento em um editor de texto comum, e o mesmo permite ao usuário mudar, fonte do texto, cor, aparência do documento escrito, etc. Tais ações ocorrem da mesma maneira quando um site esta sendo desenvolvido em HTML, onde o CSS vai permitir que o desenvolvedor transforme o documento em desenvolvimento de acordo com as características visuais que o site precisa ter. A Figura 6 mostra uma página que foi estilizada usando a linguagem CSS.

Figura 6 - Documento HTML sendo estilizado com CSS



Fonte: Mozilla (2019)

2.3.3 JavaScript

Dentre as linguagens que permitem o desenvolvimento de uma página Web o JavaScript é uma linguagem de programação que possibilita que o usuário tenha maior interação com o site. Segundo Goodman *apud* Machado & Fernandes (2018) esta linguagem permite dinamicidade entre navegador e usuário, propondo oportunidades como, por exemplo, validar os valores inseridos pelo usuário no navegador do cliente, não necessitando que seja efetuada uma requisição ao servidor. Bortolossi (2012) aborda o JavaScript como uma linguagem que assume alterar e comunicar-se, de forma dinâmica e interativa, o conteúdo e o aspecto dos vários elementos que compõem uma aplicação elaborada com HTML e CSS. É uma linguagem de script que trás muito da lógica proposta por outras linguagens de programação, sendo uma linguagem inspirado na linguagem de programação Java, considerando-se uma linguagem de alto nível.

O JavaScript tem como principais características ser um linguagem dinâmica, seu código não precisa ser traduzido para código de máquina, porque navegadores atuais interpretam a mesma onde tudo que é desenvolvido pela linguagem é

executado diretamente pelo navegador. Flanagan (2011, *apud* Machado & Fernandes, 2018, p. 32). Ressalta a versatilidade do JavaScript:

“A ampla maioria dos sites modernos usa JavaScript e todos os navegadores modernos – em computadores de mesa, consoles de jogos, tablets e smartphones – incluem interpretadores JavaScript, tornando-a a linguagem de programação mais onipresente da história. JavaScript faz parte da tríade de tecnologias que todos os desenvolvedores Web devem conhecer: HTML, para especificar o conteúdo de páginas Web; CSS, para especificar a apresentação dessas páginas e JavaScript, para determinar o comportamento delas. p,1”

A Figura 7 ilustra a utilização do JavaScript por meio da construção de um sistema que calcula e imprime fatoriais de um número.

Figura 7 - programa em JavaScript para calcular e imprimir fatoriais

```
<html>
<head>
<script language="javascript" type="text/javascript">
function response(test_form) {
  function factorial(n) {if (n == 0) return 1; else return n * factorial(n - 1);}
  var r = eval(test_form.number.value);    // r = typed in argument
  document.myform.mytext.value = "Here are the results.\n";
  for (var i = 1; i <= r; i++)              // print one line from 1 to r
    document.myform.mytext.value += (i + "! = " + factorial(i) + "\n");
}
</script>
</head>
<body>
<form name="myform">
Please enter a number: <input type="text" name="number">
<input type="button" value="compute table of factorials" onclick="response(this.form)">
<p>
<textarea name="mytext" rows=25 cols=50> </textarea>
</form>
</body>
</html>
```

Fonte: Tanenbaum (2003).

A **Figura 7** mostra o uso da JavaScript na prática, onde as várias funções permitidas pela linguagem permitem o desenvolvimento de sistemas como a calculadora apresentada na figura citada.

2.4 Back-End

Dentre as engrenagens que impulsionam o desenvolvimento Web cabe entender que forma o back-end contribui para o mesmo. De modo geral, o back-end corresponde a tudo que o usuário não vê em um site, em contra partida, ele é responsável por processar os dados que são concebidos pelo front-end. Segundo Morrison (2008) back-end é capaz de cumprir com as normas de negócio da aplicação e tem tarefa de autenticar e consistir os dados vindos do front-end (apud AMARAL & NERIS). Sempre que o usuário fazer alguma requisição será o back-end que irá tratar essa requisição e enviar os dados para que o front-end exiba as informações, com isso, é possível dizer que o back-end atua junto com os servidores, logo, várias tecnologias estão associadas ao desenvolvimento back-end entre elas:

- PHP
- Ruby on Rails
- Node.js
- Python

Esse nível de desenvolvimento está também muito associado ao processamento final dos dados trabalhando com as principais ferramentas para banco de dados como:

- MySQL
- MS-SQL
- Oracle
- SAP

2.5 Frameworks

Um framework é uma coleção integrada de elementos de software reusáveis e extensíveis para um domínio específico da aplicação e que podem ser habilitados para atender a uma aplicação em particular. Os frameworks são antigos, datando do final da década de 80, mas apenas recentemente vêm sendo amplamente empregados devido à sua integração com os padrões de projeto (SANCHES, 2005).

Outra definição importante dada por Wirfs-Brock (1991) *apud* Silva (2000) relata o seguinte:

Um framework é um esqueleto de implementação de uma aplicação ou de um subsistema de aplicação, em um domínio de problema particular. É composto de classes abstratas, concretas e provê um modelo de interação ou colaboração entre as instâncias de classes definidas pelo framework. Um framework é utilizado através de configuração ou conexão de classes concretas e derivação de novas classes concretas a partir das classes abstratas do framework.

De modo geral, os objetivos principais de se utilizar um framework advêm da modularidade, reusabilidade, extensibilidade e inversão de controle que ele fornece aos desenvolvedores (FAYAD & SCHMIT, 1997, *apud* PRATI, 2003).

2.6 Frameworks Front-End

Dadas definições a respeito de frameworks no geral, direcionados ao desenvolvimento front-end, os frameworks impulsionam o desenvolvimento de uma aplicação Web, contribuindo na criação de interfaces e na maneira em que os dados são exibidos. Uma empresa suíça chamada KeyCDN, fez um levantamento e segundo dados adquiridos pela empresa através do site GitHub e o sistema de estrelas que o GitHub utiliza, descobriu quais frameworks front-end mais utilizados atualmente (SCUDERO, 2018). Em suma, o sistema de estrelas do GitHub refere-se a um dos meios de classificação mais importantes relacionado a código, logo, quanto mais estrelas um repositório possui, mais popular e importante ele é. Segundo a pesquisa, partindo do mais popular para o menos, o ranking apresenta os seguintes frameworks:

- Bootstrap
- Foundation
- Materialize
- Semantic UI
- Material UI
- Pure

O seguinte trabalho visa fazer um comparativo entre dois dos frameworks citados e desenvolver protótipos para testes e comparações, foram escolhidos os frameworks Bootstrap por ser o framework mais utilizado segundo a pesquisa acima

e o framework Materialize por estar bem colocado no ranking acima e se tratar de um framework criado pelo Google. O framework Foundation que se encontra no segundo lugar do ranking acima não fará parte do estudo por apresentar uma implementação e uso da ferramenta que fogem da proposta da pesquisa. No presente momento se faz necessário compreender um pouco mais sobre alguns dos frameworks acima.

2.6.1 Bootstrap

O Bootstrap é o mais utilizado framework JavaScript, HTML e CSS para o desenvolvimento front-end de sites e aplicações Web responsivas, tornando o desenvolvimento de site muito mais rápido e fácil. O site oficial do Bootstrap o define como um poderoso, elegante e intuitivo framework front-end que permite o desenvolvimento web de modo prático e fácil (SILVA, 2015, *apud* ELEUTERIO, 2018). O Bootstrap é usado para ilustrar telas em HTML, tornando primordial o conhecimento de HTML para uso da ferramenta. O framework irá permitir a criação de sites completos e estruturas complexas, que podem ser acessadas de maneira simples em dispositivos diversos (SCHMITZ, 2014, *apud* ALBINO ET. AL., 2015).

O Bootstrap foi criado em 2011, onde era um projeto interno da empresa *Twitter, Inc.*, e tinha como iniciativa alinhar o modelo de arquitetura da informação com o padrão de código para soluções internas do *Twitter*. O projeto rendeu tanto sucesso que os desenvolvedores resolveram lançar o Bootstrap como um projeto livre e essa iniciativa gerou milhares de contribuições da comunidade de desenvolvedores front-end. Ele é conhecido como o framework mais completo por atender todas as linguagens que compõe o front-end: HTML, CSS e JavaScript.

Define-se como um grupo de elementos comuns em sites e páginas Web que detém soluções elegantes de menus, rodapés, colunagens, formulários, dentre inúmeros outros elementos comuns na arquitetura da informação em sites que seguem padrões globais de usabilidade e acessibilidade (CEREJA, 2016). A Figura 8 apresenta o modelo de configuração para utilização do Bootstrap, esse modelo é oferecido no próprio site do Bootstrap.

Figura 8 - Estrutura do Bootstrap

```

<!DOCTYPE html>
<html lang="pt-br">
  <head>
    <!-- Meta tags Obrigatórias -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css">

    <title>Olá, mundo!</title>
  </head>
  <body>
    <h1>Olá, mundo!</h1>

    <!-- JavaScript (Opcional) -->
    <!-- jQuery primeiro, depois Popper.js, depois Bootstrap JS -->
    <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965Dz00rT7al" ></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js" integrity="sha384-wTl" ></script>
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js" integrity="sha384" ></script>
  </body>
</html>

```

Fonte: Cereja (2016).

A **Figura 8** apresenta a estrutura inicial pra quem começa a desenvolver uma página web utilizando o framework Bootstrap, onde essa é a configuração necessária que deve ser feita para uso do mesmo.

2.6.2 Materialize

O Materialize é um framework front-end que faz uso do Material Design como inspiração para cores, ícones e formatos. O framework é inspirado no Material Design por este ser uma linguagem-conceito visual desenvolvida pela Google que tem como objetivo maior reduzir conceitos clássicos de linguagem de design e práticas que rendam melhor experiência de uso do usuário (MATERIAL DESIGN, 2018, *apud* HRUBA, 2018).

O framework assim como outros já citados foi desenvolvido com a iniciativa de acelerar o desenvolvimento de web sites e aplicações Web, oferecendo componentes estilizados personalizáveis, com recursos que permitem o design responsivo de forma ágil e eficiente, visando também proporcionar uma experiência de qualidade aos usuários (MATERIALIZE, 2018, *apud* HRUBA, 2018).

Uma das grandes vantagens do framework é que o mesmo faz bom uso do JQuery, o tornando uma ferramenta com uma alta gama de recursos visuais, como

Captions, Modais e Ligthbox (Becode, 2019). A Figura 9 apresenta o modelo de configuração para uso do Materialize dado pelo próprio site do Materialize.

Figura 9 - Estrutura do Materialize

```

<!DOCTYPE html>
<html>
  <head>
    <!--Import Google Icon Font-->
    <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="st
    <!--Import materialize.css-->
    <link type="text/css" rel="stylesheet" href="css/materialize.min.css" media

    <!--Let browser know website is optimized for mobile-->
    <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
  </head>

  <body>

    <!--JavaScript at end of body for optimized loading-->
    <script type="text/javascript" src="js/materialize.min.js"></script>
  </body>
</html>

```

Fonte: Materialize (2019).

A **Figura 9** apresenta a estrutura inicial pra quem começa a desenvolver uma página web utilizando o framework Materialize, onde essa é a configuração necessária que deve ser feita para uso do mesmo.

2.7 Trabalhos Relacionados

Este tópico apresenta alguns trabalhos relacionados ao tema da pesquisa.

O trabalho de Do Amaral (2016) apresenta uma análise comparativa entre bibliotecas Javascript para o desenvolvimento do front-end de interface web ricas e responsivas com a finalidade de distinguir o desempenho das mesmas no apoio ao reaproveitamento do back-end. Amaral (2016) efetuou um estudo com as bibliotecas JQuery Mobile e Bootstrap e para a realização das análises foram considerados os seguintes critérios de avaliação: responsividade, compatibilidade entre os navegadores, esforço de implementação e documentação. Para o estudo, foi criada uma arquitetura que serve de apoio para que o desenvolvedor possa reaproveitar o

back-end da aplicação a ser desenvolvida. Segundo os resultados, ambas as bibliotecas apoiam o reaproveitamento do back-end, mas concentram-se em características diferentes.

O trabalho de De Andrade (2013) busca realizar uma análise comparativa entre diferentes ferramentas para o desenvolvimento ágil para web que permitem o desenvolvimento de software de maneira mais rápida e eficiente, utilizando-se de recursos como geração de código e convenções sobre configurações, entre outros. Dentre o desenvolvimento do trabalho de Andrade (2013) a princípio foram realizadas pesquisas quem objetivavam definir quais ferramentas seriam utilizadas, em seguida se definiu os critérios de avaliação para alcançar os resultados onde o cenário escolhido trata-se de um conjunto de CRUDs, simulando um sistema de gerenciamento de cadastros de alunos, cursos e períodos em uma instituição, além da criação de semestres para cada curso e matriculas dos alunos nesses semestres.

3 METODOLOGIA

Esse trabalho tem como objetivo analisar dois frameworks front-end e relatar como cada framework contribui no auxílio do desenvolvimento front-end, apontando porque uma ferramenta de auxílio no desenvolvimento web reduz o tempo criação de uma página web, além da importância da gama de customização que um framework permite seus sistemas de grids personalizados e a facilidade de adaptar um site de acordo com o dispositivo usado.

A metodologia do trabalho se concentra em algumas etapas importantes:

Em primeira instância, foram realizadas pesquisas bibliográficas a respeito do assunto central e temas em comum, com o objetivo de adquirir um conhecimento maior sobre o desenvolvimento web, além de um estudo sobre frameworks buscando identificar quais os frameworks front-end mais utilizado atualmente pela comunidade de desenvolvimento web.

Após a identificação das ferramentas mais populares segundo o GitHub, foi feito um estudo concentrado em dois frameworks selecionados para a pesquisa, o Bootstrap e o Materialize.

A etapa seguinte se fundamenta em um levantamento de métricas de avaliação dos frameworks que serão estudados.

Em seguida será prototipado uma página web que será desenvolvida em cada framework e servirá como um dos critérios de comparação entre os mesmos.

Por fim, será formulado um quadro comparativo com resultado das experiências efetuadas de acordo com os critérios de avaliação anteriormente acordados.

3.1 Métricas de Avaliação

A seguinte pesquisa baseia-se em algumas métricas de comparação de frameworks já realizadas por outros pesquisadores, como a pesquisa de Do Amaral (2016), que comparou frameworks de front-end para aplicações web ricas visando o reaproveitamento do back-end e a pesquisa do site FreeCodeCamp (2019,) que realizou uma comparação de frameworks utilizando o RealWod com benchmarks.

Os critérios de avaliação escolhidos para a realização da pesquisa foram responsividade, compatibilidade entre navegadores, suporte/documentação, esforço

de implementação e linhas de código. Dito isto um mesmo site foi desenvolvido de maneira similar utilizando os dois frameworks para que suas eficácias e insuficiências fossem detectadas e analisadas.

3.1.1 Responsividade

Para avaliar a responsividade das ferramentas será levada em consideração a necessidade de adição/alteração de classes estilo utilizadas em um elemento HTML para que o site torne-se responsivo. Logo, é possível concluir que quanto menos classes de CSS utilizadas e mais classes usadas oferecidas pelo framework, melhor será a avaliação da ferramenta.

3.1.2 Compatibilidade entre navegadores

Para analisar a compatibilidade entre navegadores o site será aberto no Google Chrome, Internet Explorer e Mozilla Firefox. Os testes que serão feitos relacionados a esse critério são testes de renderização de imagem, além de que se em algum dos navegadores houve algum problema na exibição do site ou erro na formatação do site. Será analisado também se os navegadores tem alguma incompatibilidade com as classes oferecidas pelos frameworks estudados.

3.1.3 Esforço de implementação

O principal objetivo deste critério é avaliar a complexidade de desenvolver um site utilizando os frameworks estudados, analisando que os frameworks não dão cem por cento da capacidade de customização de um site, sendo necessária a criação de algumas classes no documento estilo CSS. Com isso, será avaliado o quão foi necessário a criação de classes no documento estilo.css para auxiliarem os frameworks no desenvolvimento da página Web.

3.1.4 Suporte e documentação

É de suma importância para quem quer ingressar no uso de uma nova ferramenta de desenvolvimento que a plataforma ofereça uma documentação de

suporte ao uso da ferramenta, dito isso será realizada uma análise sobre o quão as ferramentas contribuem nesses critérios para seus usuários.

3.1.5 Linhas de Código

Baseado na ideia de desenvolvimento do protótipo da página web, quantas linhas de código precisaram ser escritas para criar o site, esse critério ressalva que alguns frameworks exigem menos customização que outros, reduzindo as linhas de desenvolvimento.

3.2 Estudo Experimental

Esta seção descreve o cenário de estudo e as análises que foram feitas a fim dos critérios estabelecidos de acordo com os frameworks selecionados.

3.2.1 Cenário de Avaliação

Para realizar a análise comparativa dos frameworks foi determinado o desenvolvimento do front-end de uma página web, buscando ser implementado na construção da página o máximo de recursos possíveis oferecidos pelas ferramentas.

Dentro do contexto citado, será desenvolvido um E-commerce ³de tênis esportivo, a partir deste momento chamado de Athletic Shoes. A escolha pelo desenvolvimento do E-commerce se dá devido ser um site que exija uma maior customização, sendo possível a utilização dos vários recursos ofertados pelas ferramentas de estudo.

3.2.2 Desenvolvimento do Site

No desenvolvimento do E-commerce ambos componentes de desenvolvimento foram estabelecidos para os frameworks, com isso, na construção do site foram utilizados o editor de textos Sublime Text 3, as linguagens HTML, CSS e JavaScript e a cada um desses componentes foi integrado a biblioteca do framework, logo, dentro dos componentes citados dois sites foram construídos, um para cada framework, com a proposta de ter o design final igual, mas construídos de acordo com as possibilidades e funções permitidas por cada framework.

3.2.3 Protótipo

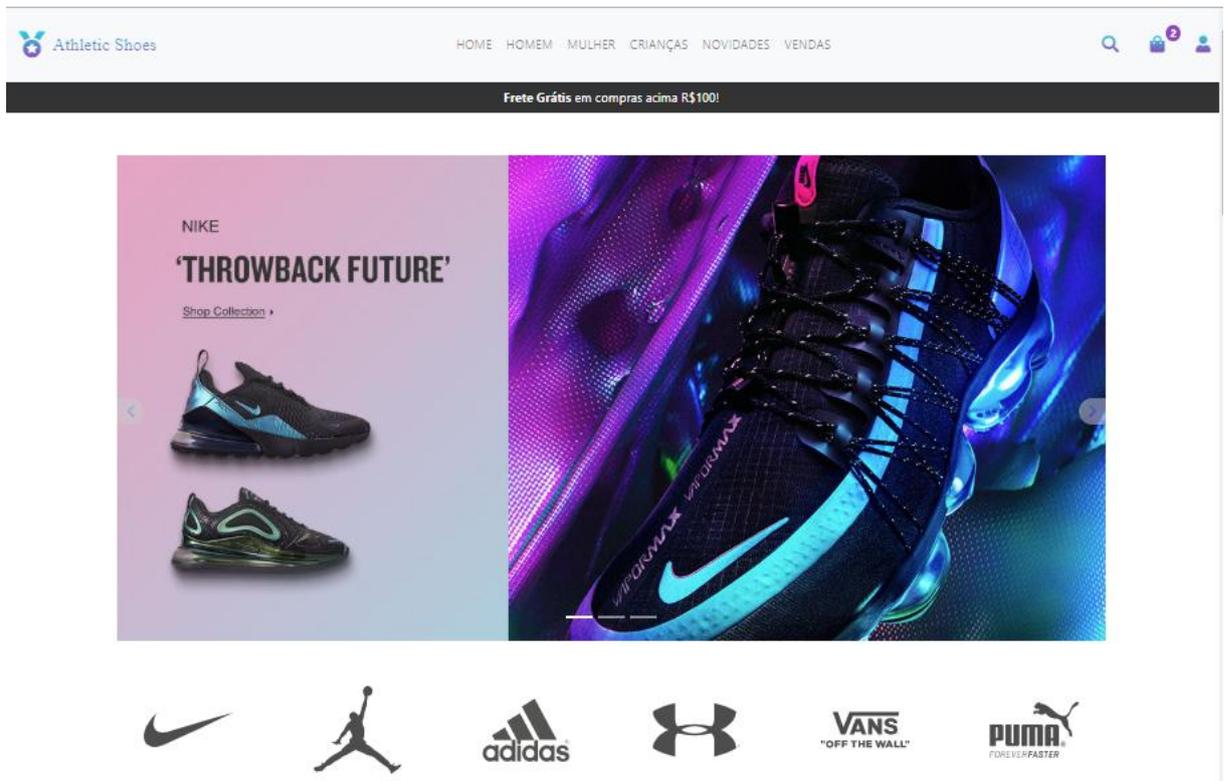
Essa seção relata o resultado final do desenvolvimento do front-end do Athletic Shoes em cada framework em versões para telas desktop e telas mobile.

As imagens a seguir mostram o protótipo do site que deve ser desenvolvido no Bootstrap e no Materialize, expondo as principais construções do site, como barra de navegação, slides de exibição de imagens, imagens dos produtos, etc.

³ E-commerce refere-se a comércio eletrônico de produtos.

A Figura 10 exibe o cabeçalho do Athletic Shoes que contém elementos que necessitam de muita customização de estilo CSS, como exemplo da barra de navegação, slide de imagens e grid para estilização da página. Esta seção exibe a ideia principal do site, expondo os principais produtos da loja no slide de imagens e as marcas nas quais o site trabalha.

Figura 10 - Cabeçalho do site.



Fonte: autor (2019).

A Figura 11 segue a seção de tênis femininos do site, exibindo imagens dos produtos onde as mesmas foram tratadas com responsividade para que se adaptem ao grid do site e deixam todo corpo do site simétrico.

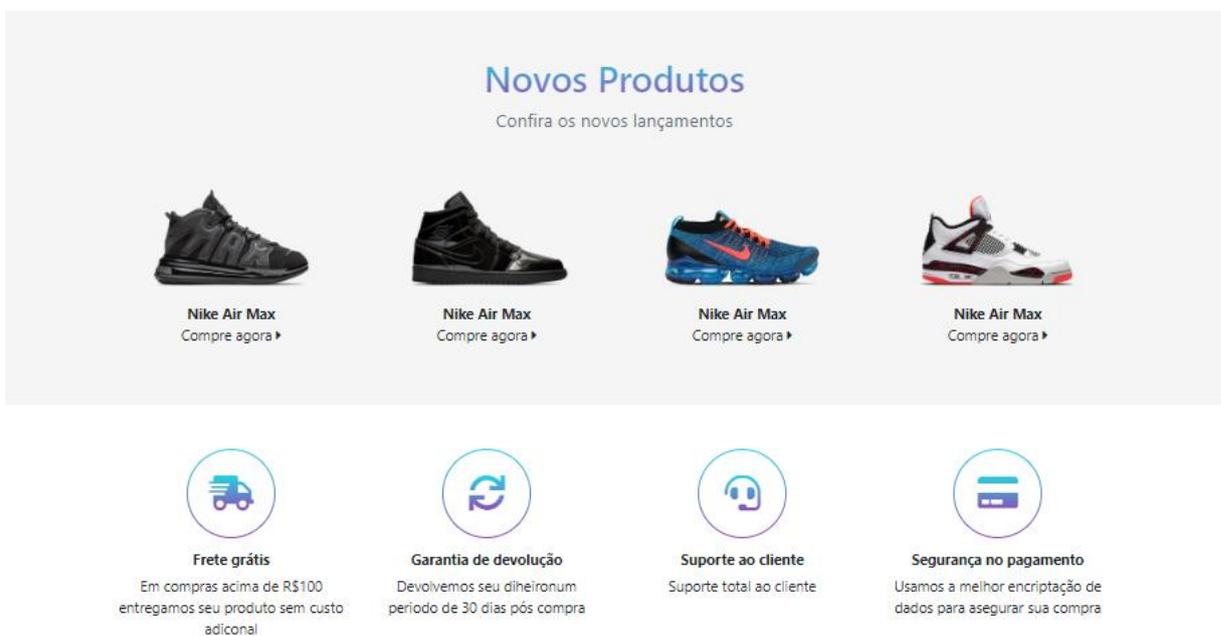
Figura 11 - Coleção feminina do site.



Fonte: Autor (2019)

No mesmo corpo do site logo após a seção anterior a página web apresenta os novos produtos da loja além dos serviços que a Athletic Shoes oferece aos seus clientes. A Figura 12 apresenta esta seção.

Figura 12 - Novos produtos e serviços.



Fonte: Autor (2019).

Por fim, a Figura 13 apresenta o rodapé do site com algumas informações sobre o mesmo.

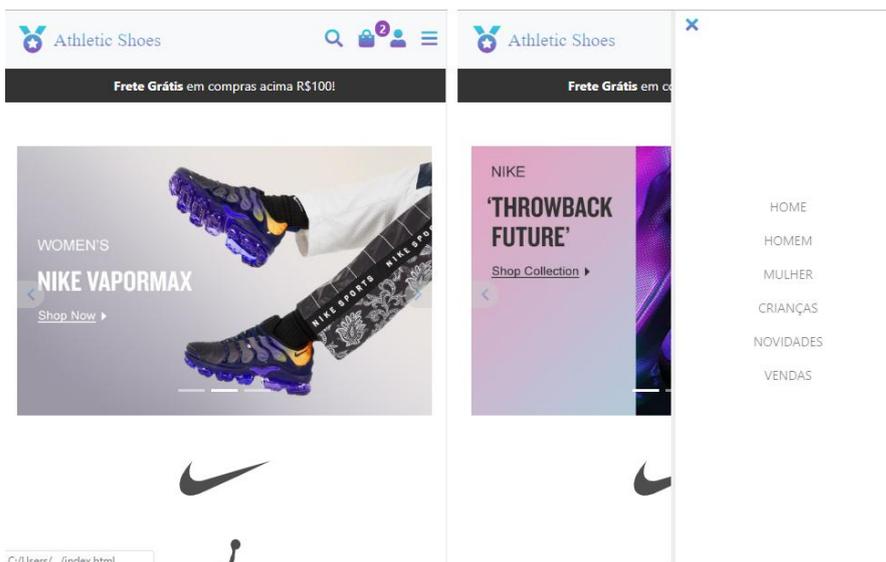
Figura 13 - Rodapé do Athletic Shoes.



Fonte: Autor (2019).

Por fim, a seção seguinte irá mostrar algumas diferenças que ocorrem quando o usuário usa um site em telas de celulares e tablets. Essas mudanças são resultado da responsividade que os frameworks permitem ao desenvolvimento web e são mudanças úteis para melhor experiência de visualização do mesmo. A Figura apresenta mudanças da barra de navegação e na exibição dos itens e imagens do site.

Figura 14 - Versão do site para dispositivos móveis.



Fonte: Autor (2019).

4 RESULTADOS E DISCUSSÕES

Depois de estabelecidas as métricas de avaliação e uma familiarização com o protótipo desenvolvido, serão descritos em seguida a análise final dos resultados apurados.

4.1 Responsividade

A Tabela 1 mostra quantas classes foram utilizadas no HTML do site Athletic Shoes além de quanta customização de grid foi utilizada para dar responsividade aos componentes utilizados.

Tabela 1 - Tabela referente às informações de responsividade dos frameworks.

Frameworks	Classes do Framework	Classes CSS usadas	Grids utilizados	Suporte a Slides
Bootstrap	206	77	5	Ok
Materialize	125	62	5	Ok

Fonte: Autor (2019).

A tabela mostra que o framework Bootstrap ofereceu ao desenvolvimento do site uma alta gama de customização, sendo necessário um numero bem reduzido de classes estilo CSS se for comparado que o framework ofertou 206 classes ao corpo HTML sendo necessária um numero de classes CSS menor. Em contra partida, o Materialize ofereceu ao site um número menor de customização sendo necessária uma utilização considerável de casses estilo CSS, visto que, foram ofertadas 125 classes do framework Materialize e necessária um número de classes CSS próximo ao que foi feito no Bootstrap. Ou seja, para uma customização com menos recursos oferecidos pelo Materialize foi preciso construir um número de classes CSS similar ao Bootstrap que ofereceu mais recursos de customização.

4.2 Compatibilidade entre navegadores

Na problemática sobre compatibilidade será avaliado o comportamento do site nos navegadores escolhidos. A Tabela 2 apresenta se existiu ou não alguma falha

na exibição do site em determinado navegador ou incompatibilidade com algum componente fornecido pelas ferramentas em estudo. Para cada renderização correta o navegador receberá o valor OK como validação.

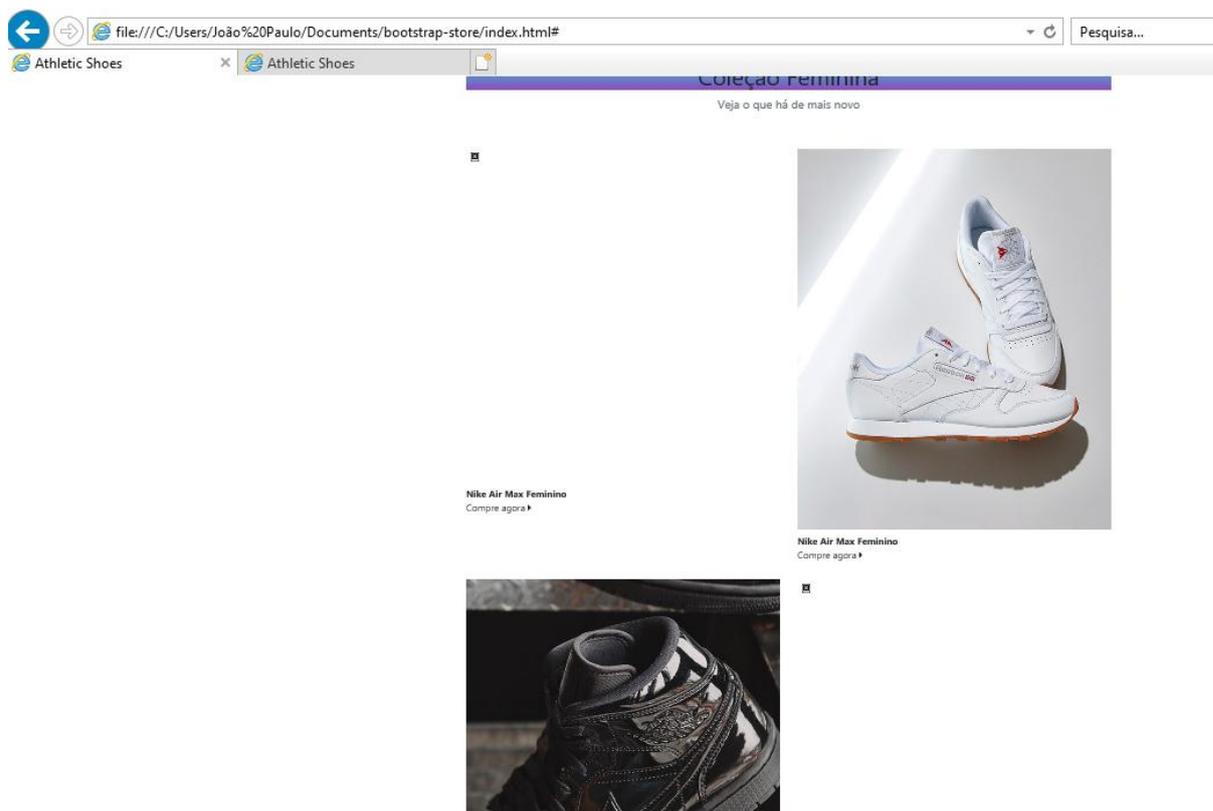
Tabela 2 - Resultado sobre compatibilidade dos navegadores com os frameworks.

Frameworks	Google Chrome	Internet Explorer	Mozilla Firefox
Bootstrap	OK	Falha	OK
Materialize	OK	Falha	OK

Fonte: Autor (2019).

A tabela mostrou que os dois frameworks demonstraram certa incompatibilidade com o navegador Internet Explorer. A Figura 15 mostra alguns erros de renderização apresentados no navegador em questão.

Figura 15 - Erros de renderização de imagem no Internet Explorer.



Fonte: Autor (2019).

4.3 Esforço de implementação

Nesse critério serão avaliadas quantas classes CSS foram criadas no documento estilo CSS para complementar o que os frameworks não podiam oferecer ao documento HTML. Diferente da avaliação de responsividade, serão analisadas quantas classes foram criadas ou quantas modificações de estilo foram feitas no documento CSS do site, não será levado em questão quantas vezes as classes se apresentam dentro do documento HTML por já ter sido analisado anteriormente. A Tabela 3 aponta o resultado dessa análise.

Tabela 3 - Tabela de implementação CSS.

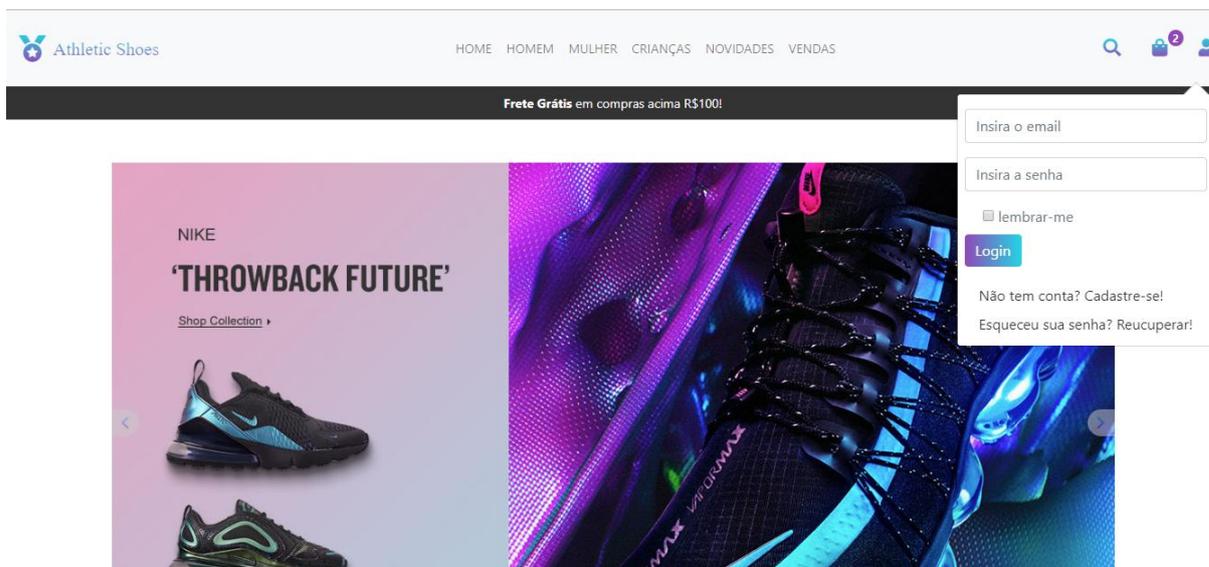
Frameworks	Nº de classes CSS criadas e Modificação de estilo feitas
Bootstrap	53
Materialize	46

Fonte: Autor (2019).

A tabela mostra que foi necessária maior criação de classes estilo CSS e suas respectivas modificações de estilos que são realizadas no mesmo documento ao site que foi desenvolvido utilizando o Bootstrap

Um ponto que tem que ser levado em consideração é que foi possível a criação de uma interface para login de usuário utilizando os recursos do framework Bootstrap que não foi possível ser feita no Materialize, isso leva a necessidade de criação de mais classes estilo CSS ao site feito com o Bootstrap. Logo, os valores da tabela não apontam uma desvantagem do Bootstrap sobre o Materialize, visto que dada uma maior customização oferecida pelo Bootstrap foi preciso criar mais classes de estilo. A Figura 16 apresenta a interface citada.

Figura 16 - Interface para login de usuário criada apenas no Bootstrap.



Fonte: Autor (2019).

4.4 Suporte e documentação

Será avaliado nesta seção o quanto a empresa se preocupa com os desenvolvedores e se o possível conteúdo que a mesma entrega é de fácil compreensão para seus usuários. Para isso é dado OK como valor de avaliação se cada empresa disponibiliza documentação em seu próprio site e se é fácil conseguir documentação de sites terceiros, como fóruns, etc. a Tabela 4 apresenta os resultados.

Tabela 4 - Tabela sobre disponibilidade de documentação.

Frameworks	Documentação	Fóruns	Exemplos de Implementação
Bootstrap	OK	OK	OK
Materialize	OK	OK	OK

Fonte: Autor (2019).

Os resultados da tabela anterior sugere que os sites dos respectivos frameworks oferecem informação suficiente para que um desenvolvedor que ainda não conheça a ferramenta possa ter material de estudo disponível para aperfeiçoamento da mesma. Os sites dos frameworks apresentam documentação e

exemplos de utilização dos componentes das ferramentas, além de diversos fóruns e sites oferecerem o mesmo nível de informação.

4.5 Linhas de código

A tabela 5 expõe quantas linhas de código foram necessárias para todo o desenvolvimento do site em cada framework estudado.

Tabela 5 - Linhas de código escritas por cada Framework.

Frameworks	Código HTML	Código CSS	Código JavaScript	Código total do documento
Bootstrap	404	256	6	666
Materialize	352	232	11	595

Fonte: Autor (2019).

Dados os resultados da tabela anterior é notável que o framework Bootstrap apresente um numero de linhas de códigos consideravelmente similar ao framework Materialize, frisando novamente que no Bootstrap foi possível introduzir ao seu código uma interface login de usuário, fato que fez com que fosse preciso a escrita de mais código no documento do Bootstrap, o mesmo não foi feito no Materialize devido à documentação do mesmo não mostrar recursos que permitissem a mesma customização feita no Bootstrap. Com isso, pode se concluir, que com maior número de recursos oferecidos pelo Bootstrap foi possível fazer um site com uma quantidade de código quase similar ao Materialize onde no mesmo algumas funcionalidades do site não obtiveram a mesma exatidão de funcionalidade como o Bootstrap.

5 CONCLUSÃO

Este trabalho teve como principal foco uma análise sobre frameworks para o desenvolvimento web, buscando propor aos desenvolvedores uma melhor experiência do desenvolvimento front-end através da utilização de ferramentas que permitem uma customização web mais sofisticada e dinâmica.

A existência de diversos frameworks front-end para o desenvolvimento web dificulta a escolha do desenvolvedor sobre qual ferramenta ideal para uso. Torna-se fundamental uma análise que busque comparar as ferramentas mais utilizadas atualmente a fim de esclarecer a quantidade de recursos que os frameworks podem proporcionar ao desenvolvedor, além de qual framework se aplica melhor a proposta de desenvolvimento dos projetos exigidos, gerando para quem desenvolve uma diminuição de tempo de criação do site e ao usuário final uma melhor experiência de uso do mesmo, tanto visual, como interativa.

Os critérios de avaliação escolhidos para a análise dos resultados foram escolhidos de acordo com a importância de adaptação de uso de uma nova ferramenta de trabalho. Logo, com as métricas de análises já estabelecidas, as mesmas foram testadas e analisadas em um cenário de avaliação criado.

Baseado nos resultados apurados dentro do cenário de testes estabelecido é plausível afirmar que os frameworks examinados possuem uma semelhança de utilização. Levando em conta que os quadros de análises criados mostram um resultado próximo entre as ferramentas, com a ressalva de que o framework Bootstrap aponta uma pequena vantagem sobre o Materialize oferecendo um nível de customização maior e um nível organizacional da estrutura do documento mais sofisticada, exigindo menos utilização de recursos CSS que o Materialize.

Como recomendações para trabalhos futuros é sugerido um estudo comparativo entre mais frameworks front-end web, além de um estudo sobre frameworks front-end num contexto geral, explorando como, por exemplo, frameworks front-end para desenvolvimento móbil.

REFERÊNCIAS

ALBINO, João Pedro et al. **Design de interfaces para web baseados no sistema de grade do bootstrap 3 interface design for web based on grid system bootstrap 3**. 2015.

BATTISTI, Júlio. **SQL Server 2000: Administração e Desenvolvimento – Curso Completo**. 2. ed. Rio de Janeiro: Axcell Books, 2001.

BOOTSTRAP. 2019. Disponível em <https://getbootstrap.com.br/docs/4.1/getting-started/introduction/>. Acesso em: 02 jun. 2019.

BRITO, Edivaldo. World Wide Web completa 25 anos em meio à crise de espionagem global. **TechTudo**, [S. l.], p. sn, 12 mar. 2014. Disponível em: <https://www.techtudo.com.br/noticias/noticia/2014/03/world-wide-web-completa-25-anos-em-meio-crise-de-espionagem-global.html>. Acesso em: 3 nov. 2019.

BROOKS, David R. **an Introduction to HTML and JavaScript: for Scientists and Engineers**. Springer Science & Business Media, 2007.

BORTOLOSSI, H. J. **Criando conteúdos educacionais digitais interativos em matemática e estatística com o uso integrado de tecnologias: GeoGebra, JavaView, HTML, CSS, MathML e JavaScript**. Revista do Instituto GeoGebra Internacional de São Paulo. ISSN 2237-9657, v. 1, n. 1, 2012.

DA SILVA CEREJA, José Ricardo. **Lâmpada Mágica-Framework para criação de ambientes de front-end**. 2016.

DE ANDRADE, Gernan Soares. **UMA ANÁLISE COMPARATIVA ENTRE AS DIFERENTES FERRAMENTAS PARA DESENVOLVIMENTO ÁGIL PARA WEB**. Orientador: Pablo Ribeiro Suárez. 2013. Dissertação (Bacharelado em Sistemas de Informação) - UNIFIP, Patos PB, 2013

HTML básico - códigos HTML. **DEV MEDIA**, 2010. Disponível em: <<https://www.devmedia.com.br/html-basico-codigos-html/16596>>. Acesso em: 02, Jun 2019.

DO AMARAL, Rodrigo Augusto; DE ALMEIDA NERIS, Vânia Paula. **Análise comparativa entre frameworks de frontend para aplicações web ricas visando reaproveitamento do back-end**. Revista TIS, v. 4, n. 1, 2016.

ELEUTERIO, Andreíza de Paula da Silva. **Moockinho**: um sistema de cursos online. 2018.

E SILVA, Ricardo Pereira. **Suporte ao desenvolvimento e uso de frameworks e componentes**. 2000.

FOUNDATION. 2019. Disponível em: <https://foundation.zurb.com/>. Acesso em: 02 jun. 2019.

FreeCodeCamp. 2019. Disponível em: <https://www.freecodecamp.org/news/a-realworld-comparison-of-front-end-frameworks-with-benchmarks-2019-update-4be0d3c78075/>. Acesso em: 08 nov. 2019.

GRANNELL, C. **The essential guide to CSS and HTML web design**. New York: APress, 2007.

HRUBA, Filipe Franchini. **Desenvolvimento de interfaces web adaptado para portadores de daltonismo**. 2018. Trabalho de Conclusão de Curso. Universidade Tecnológica Federal do Paraná.

KORPELA, J. **Programs vs. markup**. IT and communication. Tampere University of Technology, 2005.

KUROSE, James F.; ROSS, Keith W.; ZUCCHI, Wagner Luiz. **Redes de Computadores ea Internet: uma abordagem top-down**. Pearson Addison Wesley, 2007.

LUJÁN-MORA, Sergio. **Programación en Internet: clientes web**. Editorial Club Universitario, 2001.

MACHADO, ADRIANO DA CONCEIÇÃO; FERNANDES, WYLLER SILVEIRA. **DESENVOLVIMENTO DE FERRAMENTA ON-LINE EDUCACIONAL PARA CÁLCULO DE ESTRUTURAS RETICULADAS VIA MÉTODO DOS DESLOCAMENTOS EM FORMULAÇÃO MATRICIAL IMPLEMENTADA EM LINGUAGEM JAVASCRIPT (JAWS)**, 2018.

MELO, Amanda Meincke et al. **Design inclusivo de sistemas de informação na web**. 2007.

MATERIALIZE. 2019. Disponível em: <https://materializecss.com/>. Acesso em: 02 jun. 2019.

MOZILLA. Developer Network. 2017. Disponível em: <<https://developer.mozilla.org>>. Acesso em Maio 2019.

SANCHES, Mauricio Gruhn et al. **Um estudo sobre os riscos inerentes a implantação do reuso de componentes no processo de desenvolvimento de software**. 2005.

SCUDERO, Erick. **Os 6 frameworks front-end mais amados no mundo (segundo o GitHub)**. [S. l.], 19 fev. 2018. Disponível em: <https://becode.com.br/frameworks-front-end-mais-amados-segundo-github/>. Acesso em: 2 jun. 2019.

TANENBAUM, Andrew S. et al. **Computer networks**, 4-th edition. ed: Prentice Hall, 2003.

TECHIO, Gabriel Bressan; CHICON, Patrícia Mariotto Mozzaquatro. Objeto Virtual de Aprendizagem LASE: Aprendendo Sobre a Engenharia de Software Adaptação com Frameworks Bootstrap e Foundation. **I Seminário de Pesquisa Científica e Tecnológica**, v. 1, n. 1, 2017.

TEIXEIRA, Mário Meireles; SANTANA, Marcos José; SANTANA, Regina HC. **Servidor Web com Diferenciação de Serviços: Fornecendo QoS para os Serviços da Internet**. 2005.

Umblor. 2017. **Mercado de desenvolvimento web: panorama 2016 e expectativas 2017**. Disponível em: <<https://blog.umblor.com/br/mercado-de-desenvolvimento-web-panorama-2016-e-expectativas-2017/>>. Acesso em: 04 jun. 2019.

VASKEVITCH, David. **Estratégia Cliente/Servidor**: um guia para a reengenharia da empresa. São Paulo: Berkeley, 1995.