INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA SUL-RIO-GRANDENSE - IFSUL, CÂMPUS PASSO FUNDO
CURSO DE TECNOLOGIA EM SISTEMAS PARA INTERNET

FELIPE GASPARIN GRANDO

ESTUDO DE FRAMEWORKS DE DESIGN RESPONSIVO NO DESENVOLVIMENTO DE APLICAÇÕES WEB

PASSO FUNDO
2016

FELIPE GASPARIN GRANDO

ESTUDO DE FRAMEWORKS DE DESIGN RESPONSIVO NO DESENVOLVIMENTO DE APLICAÇÕES WEB

Monografia submetida como requisito parcial para a obtenção do título de Tecnólogo em Sistemas para Internet no Curso de Tecnologia em Sistemas para Internet do Instituto Federal Sul-Rio-Grandense, Campus Passo Fundo.

Orientador: Prof. Maikon Cismoski dos Santos

PASSO FUNDO

2016

FELIPE GASPARIN GRANDO

ESTUDO DE FRAMEWORKS DE DESIGN RESPONSIVO NO DESENVOLVIMENTO DE APLICAÇÕES WEB

Trabalho de Conclusão de Curso aprovado em 22/06/2016 como requisito parcial para a obtenção do título de Tecnólogo em Sistemas para Internet

Banca Examinadora:

_____

Prof. Me. Maikon Cismoski dos Santos
(Orientador)

_____

Prof. Me. Adilso Nunes de Souza
Avaliador

_____

Prof. Dr. Josue Toebe
Avaliador

_____

Prof. Me. Adilso Nunes de Souza
Coordenador do Curso

PASSO FUNDO
2016

RESUMO

O avanço computacional possibilitou o surgimento de diferentes dispositivos com acesso à web, exigindo o desenvolvimento de aplicações que sejam flexíveis de modo que, independentemente da plataforma utilizada pelo usuário, sempre mantenha o layout organizado e assim, forneça uma maior facilidade de navegação. Para atender a demanda de aplicações com essas características, surgem os frameworks de design responsivo, reduzindo o tempo de implementação através da reutilização de códigos. O presente trabalho traz um estudo, analisando os frameworks *Bootstrap, Foundation e Skeleton.* Um website foi desenvolvido como ambiente de testes através de três versões, uma para cada framework, permitindo compará-los. Assim, vantagens e limitações foram identificadas através diferentes aspectos analisados, como a construção do layout, a implementação, os recursos de interface fornecidos, o emprego do design responsivo e customização de componentes através de cada framework.

Palavras-chave: Design Responsivo, Frameworks, Bootstrap, Foundation, Skeleton, Convenções Web.

ABSTRACT

The computational advances made possible the emergence of different devices with web access, requiring the development of applications that are flexible so that regardless of platform used by the user, always keep organized layout and thus provide an easier navigation. To meet the demand for applications with these features, responsive design frameworks emerge, reducing deployment time through code reuse. This paper presents a study analyzing frameworks *Bootstrap*, *Foundation* and *Skeleton*. A website was developed as a test environment through three versions, one for each framework, allowing their comparison. In this way, advantages and limitations were identified through different aspects analyzed, such as building layout, implementation, provided the interface features, the use of responsive design and customization components.

Keywords: Responsive Design, Frameworks, Bootstrap, Foundation, Skeleton, Web Conventions.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

Com a evolução da tecnologia, a diversidade e quantidade de dispositivos computacionais com capacidade de conexão à internet está em constante crescimento. Computadores desktop, notebooks, tablets, smartphones, consoles de videogame e smart TVs são exemplos que permitem aos usuários navegar no *World Wide Web*.

O acesso à websites que não apresentam layouts adequados à variedade de resoluções de tela desses aparelhos podem trazer problemas, como a visualização de uma interface desorganizada e dificuldades de navegação. A técnica de Design Responsivo surge parar suprir essa necessidade, permitindo a implementação de layouts capazes de adaptar o seu conteúdo e manter uma boa navegação, independentemente do dispositivo utilizado pelo usuário (Allsopp, 2000).

As linguagens CSS e HTML, bases do desenvolvimento de páginas web, possibilitam a implementação de websites com características responsivas. Há elementos no CSS, conhecidos como *medias queries*, que conseguem detectar a resolução de tela do aparelho. A partir dessa característica, é possível desenvolver um layout adequado a cada situação. Isso pode ser uma tarefa complexa, pois é necessário a implementação de um novo design para cada nova resolução (Marcotte, 2010).

Em contraste, surge uma variedade de frameworks capazes de simplificar e facilitar a implementação de aplicações web responsivas. Os frameworks *Bootstrap* (Bootstrap, 2015)*, Foundation* (Foundation, 2015) *e Skeleton* (Skeleton, 2015) são exemplos que se destacam no desenvolvimento web, permitindo a reutilização de códigos e reduzindo o tempo de implementação.

1.1 JUSTIFICATIVA

As diferentes características presentes nos frameworks de design responsivo permitem a elaboração de um estudo através de uma análise comparativa, verificando vantagens e limitações que estes podem apresentar ao implementar websites responsivos.

Os aspectos negativos podem influenciar ao definir qual framework pode ser utilizado pelo desenvolvedor e os aspectos positivos, quando unificados, podem ser

empregados no desenvolvimento de uma aplicação que contemple os recursos mais vantajosos identificados em cada framework.

1.2 OBJETIVOS

O objetivo desse trabalho é realizar um estudo sobre frameworks para o desenvolvimento de interfaces web responsivas.

A partir desse objetivo geral, alguns objetivos específicos foram definidos para o desenvolvimento do trabalho:

- Projetar um website como um ambiente de testes para o estudo.
- Criar o website projetado e implementá-lo através dos recursos de cada um dos seguintes frameworks: *Bootstrap, Foundation e Skeleton*.
- Analisar e comparar os frameworks em relação às suas diferentes características.
- Identificar vantagens e desvantagens em cada framework.

1.3 ESTRUTURA DA MONOGRAFIA

Essa monografia foi dividida nos seguintes capítulos, os quais visam uma melhor organização e estruturação para o entendimento da mesma:

- Capítulo 2 (Referencial Teórico): apresenta os conceitos e embasamentos necessários para a construção do conhecimento empregado no desenvolvimento do website projetado como ambiente de testes para o estudo.
- Capítulo 3 (Metodologia): descreve os métodos e técnicas que serão empregados no desenvolvimento prático do trabalho, desde o projeto do website até os aspectos que serão avaliados na análise comparativa.
- Capítulo 4 (Resultados): por fim, este capítulo apresenta os resultados obtidos através da aplicação da metodologia proposta.

2 REFERENCIAL TEÓRICO

Neste capítulo são apresentados os conceitos que fornecem o embasamento teórico para a construção do conhecimento exigido no presente trabalho. Dentre os conceitos estudados estão as linguagens web, convenções web, design responsivo, ferramentas para teste de design responsivo e frameworks de design responsivo, dentre outros.

2.1 LINGUAGENS WEB

Segundo Mazza (2012), o desenvolvimento web se encontra em uma época revolucionária. Diversas empresas impulsionam a evolução das tecnologias que usamos para criar uma web melhor, trabalham exclusivamente em melhorias para os navegadores mais utilizados, participam na definição de novos padrões e também disseminam o conhecimento juntamente à comunidade de desenvolvedores.

Mazza (2012), destaca a importância de aprender algumas linguagens do desenvolvimento web, como HTML e CSS, no seguinte trecho:

> [...] importância de se aprender HTML e CSS. Se você estiver trabalhando ou pretende trabalhar com tecnologia, acredito que boa parte – ou tudo – dos seus projetos será utilizado através de um navegador. Seja um sistema interno de um banco, uma rede social, um grande portal de notícias ou sites para campanhas de publicidade, o meio comum hoje em dia é a web, e é bastante interessante ter uma ótima base de conhecimento [..] (Mazza, 2012. p. 2)

O autor ainda afirma que no começo do desenvolvimento web, desde o surgimento dos primeiros navegadores como o Netscape e as primeiras versões do Internet Explorer em 1995, sempre houve uma guerra de incompatibilidade, dificultando o funcionamento da web em clientes diferentes. Essa briga impulsionou a busca por padrões, bem como a evolução das tecnologias que fazem os navegadores funcionar (Mazza, 2012).

Assim, com o intuito de aprender as tecnologias que são tradicionalmente utilizadas como padrões nos navegadores, nesta seção são apresentadas as linguagens que formam a base da implementação de páginas web: HTML, CSS e JavaScript. Estas trabalham em conjunto sendo empregadas na criação de websites e também servem como base para a implementação de frameworks de design responsivo.

2.1.1 HTML

HyperText Markup Language (HTML), traduzindo para o português, Linguagem de Marcação de Hipertexto, é uma linguagem que surgiu em 1990 através de pesquisas realizadas por Tim Berners-Lee ao buscar um meio de compartilhamento eletrônico de textos e pesquisas entre os cientistas do mundo inteiro (Maurício Silva[1], 2008). Sua marcação pode ser definida pelo uso de *tags,* as quais são responsáveis por organizar e estruturar o conteúdo e os elementos de interface que compõem a página.

Tim Berners-Lee desenvolveu um software e um protocolo para recuperação de hipertextos denominado *Hipertext Transfer Protocol* (HTTP) e o formato de hipertexto utilizado por ele foi denominado HTML. Assim, em 1990, Tim conseguiu recuperar hipertextos em diferentes computadores na sua estação de trabalho, criando o HTML e dando origem ao surgimento da Web (Maurício Silva[1], 2008).

Entre os anos de 1991 a 1998 o HTML evoluiu de sua primeira versão até a quarta. Somente em 1997 a versão 3.2 do HTML passou a ser recomentada pelo *Word Web Consortium*[1] (W3C). Já a sua atual versão, o HTML5, foi recomendada depois de 17 anos, em 2014 (W3C, 2016).

Pode-se dizer que hipertexto é todo o conteúdo inserido e estruturado em um documento e que tem como principal característica a possibilidade de interligação a outros documentos na web. A linguagem HTML tem como objetivo criar documentos que podem ser lidos praticamente em qualquer tipo de computador e que através de navegadores, possam ser transmitidos pela internet (Maurício Silva[1], 2008).

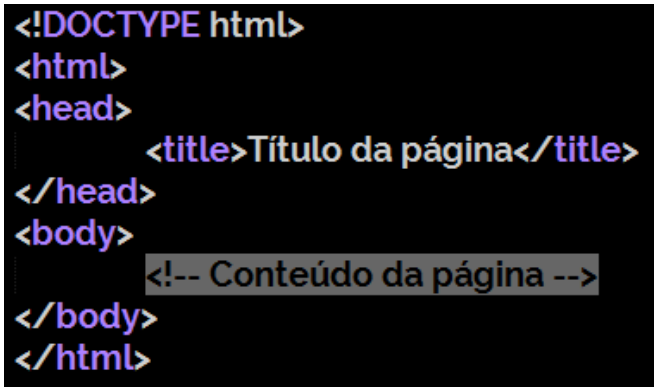
O HTML é o responsável por dizer aos navegadores como será apresentada a página através de suas tags, passando a estrutura e o significado dos seus textos. Ele mostra ao *browser* a estrutura de seu documento: onde estão os títulos, onde estão os parágrafos, o que precisa de destaque e assim por diante. Dadas essas informações, os navegadores possuem regras padrões internas para a exibição de cada um desses elementos (Freeman, 2008).

[1] O Consórcio *World Wide Web* (W3C) é um consórcio internacional no qual organizações filiadas, uma equipe em tempo integral e o público trabalham juntos para desenvolver padrões para a Web.

Segundo Harold (2010), chegará um ponto no qual uma aplicação web necessitará ser expressa em HTML, que é a linguagem universal de descrição de páginas web. Apesar de ser uma linguagem limitada, ela é bastante especializada.

A estrutura básica de um documento HTML pode ser visualizada através da Figura 1, apresentando as principais tags que a compõe.

Figura 1: Estrutura básica de um documento HTML



Fonte: DO AUTOR (2016)

A tag <*html*> ao navegador que o documento é realmente um arquivo HTML. A tag <*head*> é responsável por conter informações como importações de outros arquivos, definições de estilos, outras referências e o título da página, como exemplificado na figura através da tag <*title*>. Já na tag <*body*>, pode ser inserido o conteúdo da página, o quel será exibido aos usuários.

Atualmente na sua quinta versão, conhecida como HTML5, a linguagem trouxe novos elementos e funcionalidades, possibilitando melhores experiências e integrações aos desenvolvedores. Escrever HTML tornou-se mais simples (Mazza, 2012).

Segundo Castro e Hylsop (2013), saber um pouco do básico da origem do HTML ajuda a entender o HTML5. O número de versões do HTML aumentou conforme a evolução da linguagem, que introduziu outros elementos e ajustes às suas regras. Assim, o HTML5 nada mais é do que uma evolução natural de seus antecessores, buscando refletir as necessidades dos sites atuais quanto os do futuro. Ele adiciona novos recursos como os elementos adicionais de artigo, seção e figura, que são utilizados para descrever os conteúdos.

A evolução e simplificação da linguagem através de suas versões pode ser observada através dos exemplos ilustrados na Figura 2 e 3, apresentando as mudança significativa ocorridas na atual tag *<!DOCTYPE>*.

Figura 2: Tag *<!DOCTYPE>* na quarta versão da linguagem HTML

```
<!-- DTD do HTML 4, em modo "strict". -->
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
```

Fonte: MAZZA (2012)

Figura 3: Tag *<!DOCTYPE>* na atual versão do HTML

```
<!-- Doctype atual -->
<!DOCTYPE html>
```

Fonte: MAZZA (2012)

2.1.2 CSS

Em meados dos anos 90, com a popularização dos documentos HTML, começaram a surgir atributos de estilização de fontes e cores. Assim, as *Cascading Style Sheet* (CSS), traduzidas como Folhas de Estilos em Cascata, surgiram da necessidade de dividir as tarefas, deixando a linguagem HTML responsável somente pela organização e estruturação do conteúdo, enquanto a linguagem CSS ficou responsável pela estilização (Maurício Silva[2], 2008).

A finalidade de tornar o CSS responsável pela estilização dos documentos HTML pode ser citado por Maurício Silva[2] (2008), no seguinte trecho:

> [...] não cabe à HTML fornecer informações ao agente de usuário sobre a apresentação dos elementos. Por exemplo: cores de fontes, tamanhos de textos, posicionamentos e todo o aspecto visual de um documento [...].
>
> Cabem à CSS todas as funções de apresentação de um documento, e esta é a sua finalidade maior. (Lopes, 2008. p.50)

No mês de dezembro de 1996, a primeira versão do CSS foi lançada como uma recomendação oficial do W3C. Segundo o W3C, o CSS descreve como os elementos do HTML são exibidos na tela, em papel ou outros meios de

comunicação. O CSS também permite a reutilização de códigos, já que permite controla a aparência de várias páginas da Web de uma só vez.

A sua atual terceira versão, conhecida por CSS3, é mais poderosa do que as versões anteriores e introduz inúmeros efeitos visuais, como sombras de texto, cantos arredondados e gradientes (Castro; Hylsop, 2013).

Segundo Mazza (2012), todos os navegadores possuem estilos padrões que são aplicados em todas as páginas como o negrito da tag <*strong*>, a borda do <*legend*> e os tamanhos de fonte das tags <*h1*> ao <*h6*>. Esses padrões se diferenciam entre os próprios navegadores e assim, podem ser visualizados diferentemente quando acessados pelos usuários. Assim, o CSS permite padronizar a estilização proposta através dos diferentes navegadores que temos na web.

A Figura 4 ilustra a estrutura de um documento CSS. Percebe-se que os componentes a serem estilizados podem ser selecionados através do nome, da classe ou identificador. Dentro das chaves de cada elemento, os atributos de estilização podem ser definidos.

Figura 4: Estilizações de componentes na linguagem CSS

```
body {
        background-color: white;
        margin: 0;
        padding: 0;
        font-family: 'Ubuntu Mono';
        font-size: 20px;
}
.classe {
        display: table;
        margin-top: 15px;
        height: 40px;
        transition: margin-left .5s;
}
#id {
        float: left;
        display: block;
        height: 50px;
        transition: width .5s, height .5s, margin-right .5s, margin-top  .5s;
}
```

Fonte: DO AUTOR (2016)

Percebe-se que o seletor de classes exige a utilização de um ponto anteriormente ao nome, enquanto no seletor através do identificador utiliza-se o sustenido. Segundo Freeman (2008), cada declaração em CSS consiste de um local, uma propriedade nesse local e um estilo a ser aplicado àquela propriedade.

2.1.3 JavaScript

O JavaScript é uma linguagem de programação orientada a objetos e tradicionalmente utilizada no lado do cliente. Os scripts da linguagem podem ser incorporados diretamente nas páginas web que usam a linguagem HTML, permitindo a criação de conteúdos dinâmicos.

Como exemplos práticos de sua utilização, podem ser tratadas tarefas comuns como, por exemplo, a validação de dados em formulários, manipulação de eventos ocorridos na página, o trabalho com cookies, a criação de animações portáveis através de HTML dinâmico e principalmente a manipulação de HTML e de CSS (Flanagan, 2004).

Criada por Brendan Eich em 1995, durante o período inicial da web, atualmente é mantida pela *European Computer Manufacturer's* (ECM). Seu uso geral foi incorporado primeiramente no Netscape, Internet Explorer e em outros navegadores conforme estes foram sendo aprimorados (Flanagan, 2004).

Segundo Flanagan (2013), o JavaScript é uma linguagem importante por estar presente na maior parte das aplicações existentes. No seguinte trecho, o autor cita a sua importância:

> 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 [...].
>
> 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 especificar o comportamento delas. (Flanagan, 2013. p. 1)

2.2 CONVENÇÕES WEB

Convenções são padrões que aprimoram o sentido de domínio dos usuários em relação a um site, ajudando-os a realizar suas tarefas proporcionando-os satisfação (Nielsen e Loranger, 2007).

O uso de convenções possibilita uma maior facilidade de compreensão e menor tempo de raciocínio aos usuários ao identificar e utilizar os componentes de um website. Como exemplo, estudos comprovam que a localização do logotipo no topo da página auxilia o usuário a identificar em qual website ele está navegando. Caso essa convenção não seja levada em consideração, pode-se gerar confusão ao usuário, exigindo maior raciocínio ao identificar em qual website está (Krug, 2006).

Krug (2006) exemplifica o uso de convenções no nosso cotidiano através do seguinte trecho:

> Espaços físicos como cidades e prédios têm seus próprios sistemas de navegação, com convenções que têm se desenvolvido com o tempo, como placas de ruas [...]. Colocá-los em um local padrão permite que os localizemos rapidamente e com o mínimo de esforço [...].
>
> Esperamos encontrar placas de ruas nas esquinas, esperamos encontrá-las olhando para cima (não para baixo) e esperamos que elas se pareçam com placas de rua (horizontais, não verticais). (Krug, 2006, p.60)

Uma página web deve ser autoexplicativa. Deve-se entendê-la sem muito esforço, saber o que ela é e como usá-la. Cada meio de publicação desenvolve convenções e continua a aperfeiçoá-las e a desenvolver novas no decorrer do tempo. A Web já tem muitas delas, a maioria derivada de convenções de jornais e revistas (Krug, 2006).

Nielsen e Loranger (2007) ainda afirmam que o uso de padrões na projeção de designs de websites assegura aos usuários algumas garantias:

- Saber quais recursos esperar.
- Reconhecer a aparência desses recursos na interface.
- Saber como operar cada recurso para se chegar aos objetivos.
- Não precisar adivinhar o significado de elementos de design desconhecidos.
- Não passar despercebido por elementos de design que não sejam padrões.
- Não obter surpresas desagradáveis quando algo não funciona como o esperado.

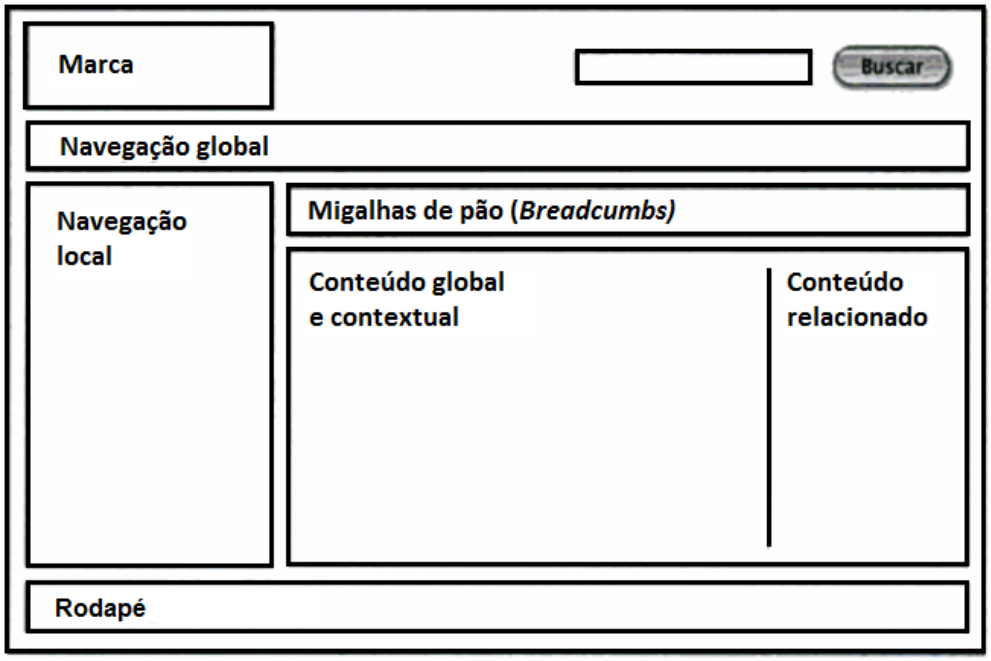
Segundo Memória (2005), a padronização de interfaces é um conceito básico e um dos mais importantes para quem projeta produtos para a internet. A solução

usada pela maioria está diretamente relacionada a conceitos de psicologia cognitiva, como facilidade de aprendizado e memorização. Na web, passados alguns anos, certas convenções já são utilizadas e respeitadas em uma série de páginas.

Nielsen, Adkinsson e Bernard, através de seus estudos, propuseram uma estrutura para o posicionamento das convenções da web (Memória, 2005).

Apresentando a marca, a barra de busca, a navegação global e local, as migalhas de pão, conteúdos e o rodapé, a Figura 5 ilustra essa estrutura através de um esboço.

Figura 5: Convenções web com base nos estudos de Nielsen, Adkinsson e Bernard



Fonte: MEMÓRIA (2005)

A estrutura das convenções sugeridas por Nielsen, Adkinsson e Bernard é proposta para dispositivos de maior resolução, como computadores desktop e notebooks. Entretanto, quando falamos de dispositivos de menor porte como tablets e smartphones, supõe-se que as convenções devem sempre ser mantidas, mesmo apresentando uma estrutura diferenciada, adequando-se conforme as diferentes resoluções.

Com base na estrutura proposta, algumas das convenções são detalhadas a seguir.

2.2.1 Marca

É o elemento mais alto na hierarquia de um website, representando-o por inteiro. Assim como o nome de um prédio e a placa de uma loja, aonde se espera encontra-los em cima de sua entrada, também é esperado ver a identificação do site no topo da página. O uso do logotipo utilizado em todas as páginas também serve como um link para a página inicial.

O logotipo, além de identificar o website, é a primeiro elemento da interface a ser visualizado pelos usuários (Nielsen; Loranger, 2007).

2.2.2 Navegação

A navegação é um meio de levar os usuários de uma página a outra dentro do website. Segundo Nielsen e Loranger (2007), os elementos navegacionais atuam como degraus para ajudar os usuários a passarem de uma área para outra. A navegação pode ser classificada em duas formas: navegação global e local.

A navegação global pode ser definida como o conjunto de elementos de navegação que aparece em todas as páginas de um website. Ela é responsável por passar consistência e a confirmação ao usuário de que este, independentemente da página, ainda está no mesmo website. Krug (2006) afirma que mantê-la igual e estável em todo o website faz com que o usuário só precise descobrir seu funcionamento uma vez.

A navegação local pode ser definida como uma forma de navegação entre seções e subseções dentro cada página acessada.

2.2.3 Busca

É uma alternativa aos menus de navegação para os usuários que preferem pesquisar ao navegar. Tradicionalmente localizada no topo da página, permite com que a pesquisa seja feita por meio de termos e outros elementos combinados.

É um dos elementos importantes em um website e os usuários esperam encontrar algumas características presentes nessa convenção. Nielsen e Loranger (2007), citam algumas delas:

- Uma caixa de texto em que se possam digitar palavras.

- Um botão rotulado "Pesquisar" em que se possa clicar para efetuar a busca.
- Uma lista com os resultados de forma linear, priorizada e que apareça em uma nova página.
- Esperam realizar a busca por meio de palavras chaves.

Krug (2006) descreve a busca como um mecanismo imprescindível em todas as páginas de um website, a menos que este seja muito pequeno e muito bem organizado.

2.2.4 Rodapé

É uma convenção localizada ao final de cada página do website. Ela contém links de navegação e também é um espaço utilizado para informações de direitos autorais do proprietário, data da última atualização, política de privacidade e links para voltar ao topo da página (Memória, 2006).

2.3 DESIGN RESPONSIVO

No início do desenvolvimento web muitas convenções adotadas para a criação de páginas eram oriundas de mídias impressas como jornais e revistas. As mídias impressas, quando criadas, apresentam layouts fixos, não permitindo alterar o tamanho de suas fontes, a organização dos elementos e o tamanho das imagens. Um website é uma mídia digital e, diferente de um jornal ou uma revista, necessita apresentar características de flexibilidade, independentemente do navegador, da plataforma ou da tela que o usuário escolher ou precise utilizar para acessar suas páginas (Allsopp, 2000).

No começo do século XX o projeto de websites que se adaptavam à diversas resoluções de tela era incomum devido a pequena diversidade de dispositivos com acesso ao conteúdo web. Conforme novos dispositivos foram surgindo, diferentes resoluções de tela vieram em conjunto, acompanhando a evolução tecnológica.

Segundo Lopes (2015), os layouts dos websites eram projetados conforme as resoluções encontradas na maioria dos monitores (800px ou 1080px). Por isso, não é mais viável projetar especificamente a uma resolução. Lopes (2015) ainda descreve o ambiente de desenvolvimento web relacionado às resoluções por volta dos anos 2000 através no seguinte trecho:

> [...] sempre tivemos monitores de 800px de largura, assim como 1024px. Hoje, as resoluções de 1280px e 1366px são as mais comuns. Mas muita gente usa as resoluções maiores, com 1600px ou 1920px. Nossos sites fixos nunca levaram isso em conta. Programávamos para a resolução mais comum e esquecíamos o resto. Nossos designs eram *pixel perfect* – copiados pixel a pixel do desenho original do Photoshop.
>
> Isso é ridículo. Isso não é a Web portável, acessível e universal que gostaríamos. (Lopes, 2015. p.30)

Outro fator a ser levado em consideração foi a evolução da Internet, que permitiu o acesso à websites através dos smartphones, que se destacaram no mercado devido à portabilidade e a possibilidade de conexão em diferentes locais. Quando se trata de meio de acesso à internet, esta plataforma chegou a porcentagem de 31% comparando-se aos demais dispositivos (Kemp, 2015). Assim, deve ser levado em consideração o desenvolvimento de aplicações que sejam adaptadas a esses dispositivos.

Dados representados na Tabela 1 revelam uma estimativa das vendas mundiais de dispositivos entre os anos de 2012 até 2017. Enquanto na Tabela 2,

são demonstradas as estatísticas de mercado dos mesmos aparelhos durante o mesmo período.

Tabela 1: Vendas mundiais de dispositivos até 2017 (em milhares de unidades)

Tipo de Dispositivo	2012	2013	2014	2017
PC (Desktop e Notebooks)	341,263	315,229	302,315	271,612
Ultramóveis	9,822	23,592	38,687	96,350
Tablets	116,113	197,202	265,731	467,951
Smartphones	1,746,176	1,875,774	1,949,722	2,128,871
Total	**2,213,373**	**2,411,796**	**2,556,455**	**2,964,783**

Fonte: GARTNER (2013)

O número de dispositivos móveis a serem vendidos a cada ano está aumentando e os de dispositivos convencionais, como computadores desktop e notebooks, ainda com destaque, estão sendo reduzidos.

Tabela 2: Mercado mundial de dispositivos até 2017 (em milhões de unidades)

Tipo de Dispositivo	Remessas de unidade em 2012	Estatísticas de mercado em 2012	Remessas de unidade em 2017	Estatísticas de mercado em 2017	Crescimento 2012-2017
Computadores Desktop	148.4	12.4%	141.0	6.0%	-5.0%
Computadores Portáteis	202.0	16.8%	240.9	11.0%	19.3%
Tablets	128.3	10.7%	352.3	16%	174.5%
Smartphones	722.4	60.1%	1,516	67%	109.9%
Total	**1,201.1**	**100.0%**	**2,250.3**	**100.0%**	**87.3%**

Fonte: MOHOROVIC (2013)

Estima-se que no ano de 2017 os dispositivos móveis façam parte de 67% de todo o mercado em relação aos demais, conforme os dados revelados na Tabela 2.

Ethan Marcotte (2010), previa que o acesso à internet através de dispositivos móveis superaria o acesso por dispositivos convencionais. A quantidade de aparelhos estava aumentando e algumas empresas necessitavam projetar para alguns dispositivos específicos como smartphones. Tornava-se inviável projetar um website específico a cada dispositivo novo que surgia no mercado.

Como solução, o autor propôs a criação de designs flexíveis e adaptáveis com o uso de layouts fluídos, imagens flexíveis e *media queries* (Seção 2.3.1), dando origem então, ao conceito do Design Responsivo. Layouts fluídos são os designs que não utilizam medidas fixas como pixels na construção dos elementos da interface e sim, medidas relativas.

Segundo Moon (2012), web design responsivo é uma abordagem que sugere que o design da interface deve responder ao comportamento do usuário e ao ambiente, baseado em tamanhos de tela, plataformas ou orientações.

Katajisto (2015) afirma que design responsivo é uma abordagem de web design que visa otimizar a experiência para os usuários, deixando que navegador faça todo o trabalho, evitando com que o leitor percorra, deslize ou redimensione a tela.

A Figura 6 ilustra o comportamento de um website criado com o uso de Design Responsivo, demonstrando sua flexibilidade através de diferentes dispositivos.

Figura 6: Website com o conceito de Design Responsivo visualizado através de diferentes dispositivos

Fonte: BATURAY E BIRTANE (2013)

2.3.1 Media Queries

As *Medias Queries*, ou Consultas de Mídia, são umas das principais ferramentas no desenvolvimento web moderno, baseadas na linguagem CSS. Com elas, é possível adicionar designs condicionais que só serão aplicados em determinadas situações. Isso permite adaptar o design da página conforme a resolução de tela específica de cada dispositivo.

Segundo Lopes (2015), as medias queries surgiram com grande utilidade através da terceira versão do CSS. A Figura 7 ilustra a codificação de um exemplo de *media querie* sugerido pelo autor.

> A sintaxe do bloco *@media* recebe uma condição e, dentro dele, as regras de CSS que só serão aplicadas caso a condição seja válida. No exemplo, o (max-width: 400px) indica que só vamos aplicar o CSS em questão quando a janela do navegador tiver, no máximo, 400px. (Lopes, 2015. p. 68)

Figura 7: Exemplo de Media Querie citada por Lopes

```
1  .noticia {
2          float: left;
3          width: 25%;
4  }
5  @media screen and (max-width: 400px) {
6          .noticia {
7                  float: left;
8                  width: 100%;
9          }
10 }
```

Fonte: DO AUTOR (2016)

No exemplo sugerido por Lopes, a *media querie* implementada a partir da linha 5 do código estabelece que os estilos que serão aplicados para a classe “*noticia*” são diferentes quando a resolução tiver no máximo 400 pixels de largura.

2.4 FERRAMENTAS PARA TESTES DE DESIGN RESPONSIVO

As ferramentas de testes de design responsivo surgem com o objetivo de reduzir custos àqueles que não tem condições de adquirir uma grande variedade de dispositivos para criar um laboratório de testes completo.

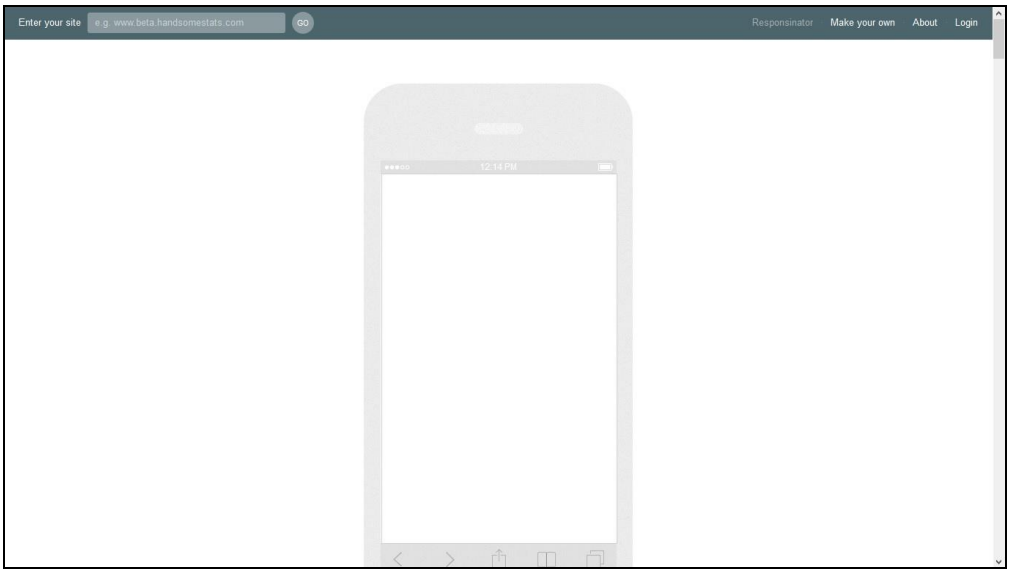
Embora algumas não simulem um ambiente de navegação real como, por exemplo, a navegação feita por meio da tecnologia de toque de tela em alguns dispositivos, elas são capazes de testar a adaptação do conteúdo web através da simulação de diferentes resoluções em dispositivos reais.

2.4.1 Reposinator

O Reposinator[2] é uma ferramenta online criada por Tama Pugsley e Andy Hovey para auxiliar nos testes de adaptação do layout de websites através da simulação de diferentes dispositivos encontrados no mercado atualmente.

A Figura 6 ilustra uma visão geral do website da ferramenta em que o teste pode ser feito. Para utilizar a ferramenta, é necessário inserir o endereço do website que se deseja testar no campo de inserção localizado no canto superior esquerdo e após, deve-se clicar no botão "*Go"*. A Figura 8 ilustra o campo de inserção.

Figura 8: Visão geral do website do Reposinator



Fonte: REPOSINATOR (2015)

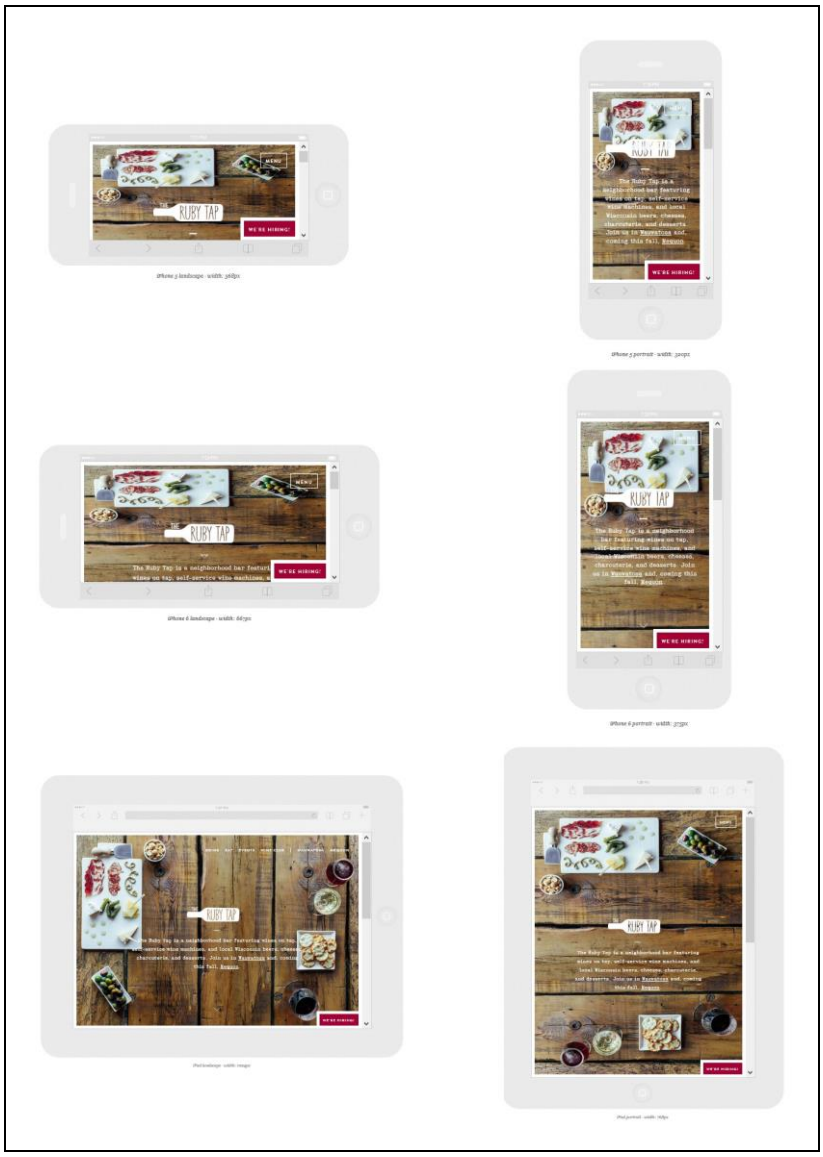
[2] Disponível em: *https://www.responsinator.com/*

Figura 9: Campo de inserção de endereços do Reposinator

Fonte: REPOSINATOR (2016)

Através de um teste realizado com um website no Reposinator, a Figura 8 ilustra capturas de tela obtidas em diferentes dispositivos, tanto em modo retrato como paisagem, demonstrando as diferentes organizações do layout com o emprego do design responsivo.

Figura 10: Resultados de testes no Reposinator



Fonte: ADAPTADA (REPOSINATOR, 2016)

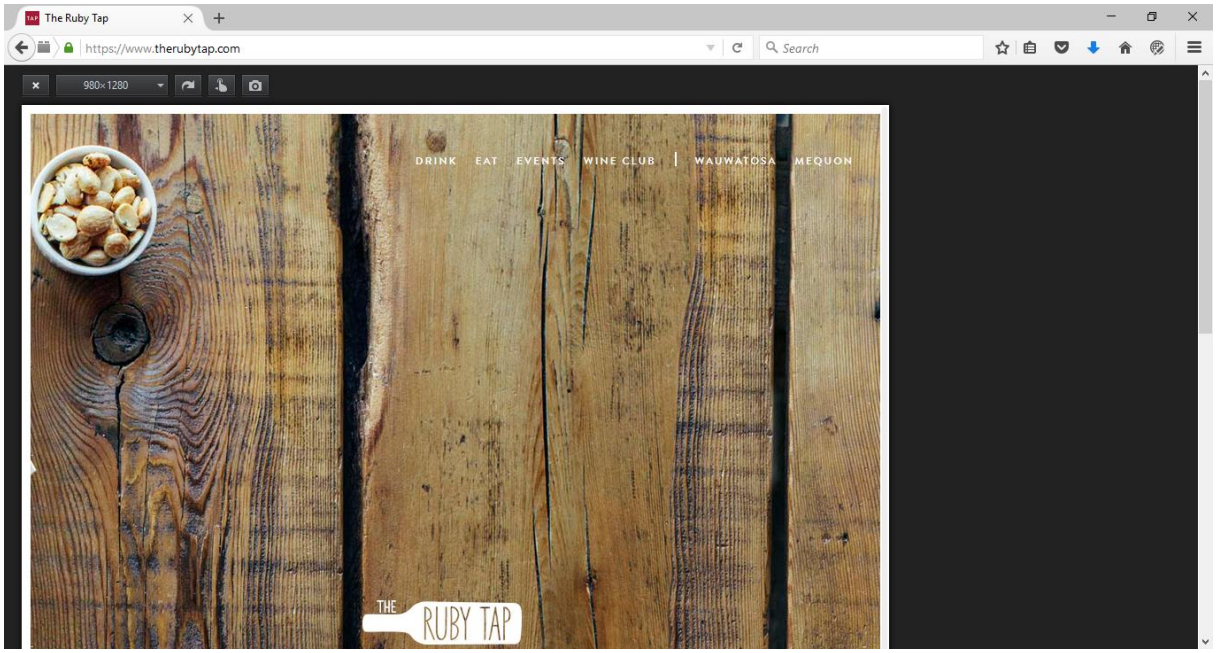
2.4.2 Plugins de Navegadores

Navegadores atuais como o Google Chrome e o Mozilla Firefox contam com ferramentas que vem como padrão para o auxílio aos desenvolvedores. Nessas ferramentas, chamadas de *plugins*, há a possibilidade de fazer a simulação de diferentes dispositivos e suas resoluções de tela.

2.4.2.1 Mozilla Firefox

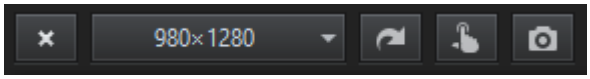
No Mozilla, esta ferramenta é chamada de *Responsive Design View que,* traduzindo para o português, significa visualização de design responsivo*.* A Figura 9 ilustra uma visão geral desta ferramenta e suas características. No canto superior esquerdo, a sua barra de ferramentas traz algumas funções como alterar a resolução, simular a navegação por touch screen e também fazer capturas de tela. A Figura 11 ilustra detalhadamente esta barra de ferramentas.

Figura 11: Visão geral - Plugin Responsive Design View do Mozilla Firefox



Fonte: FIREFOX, 2016

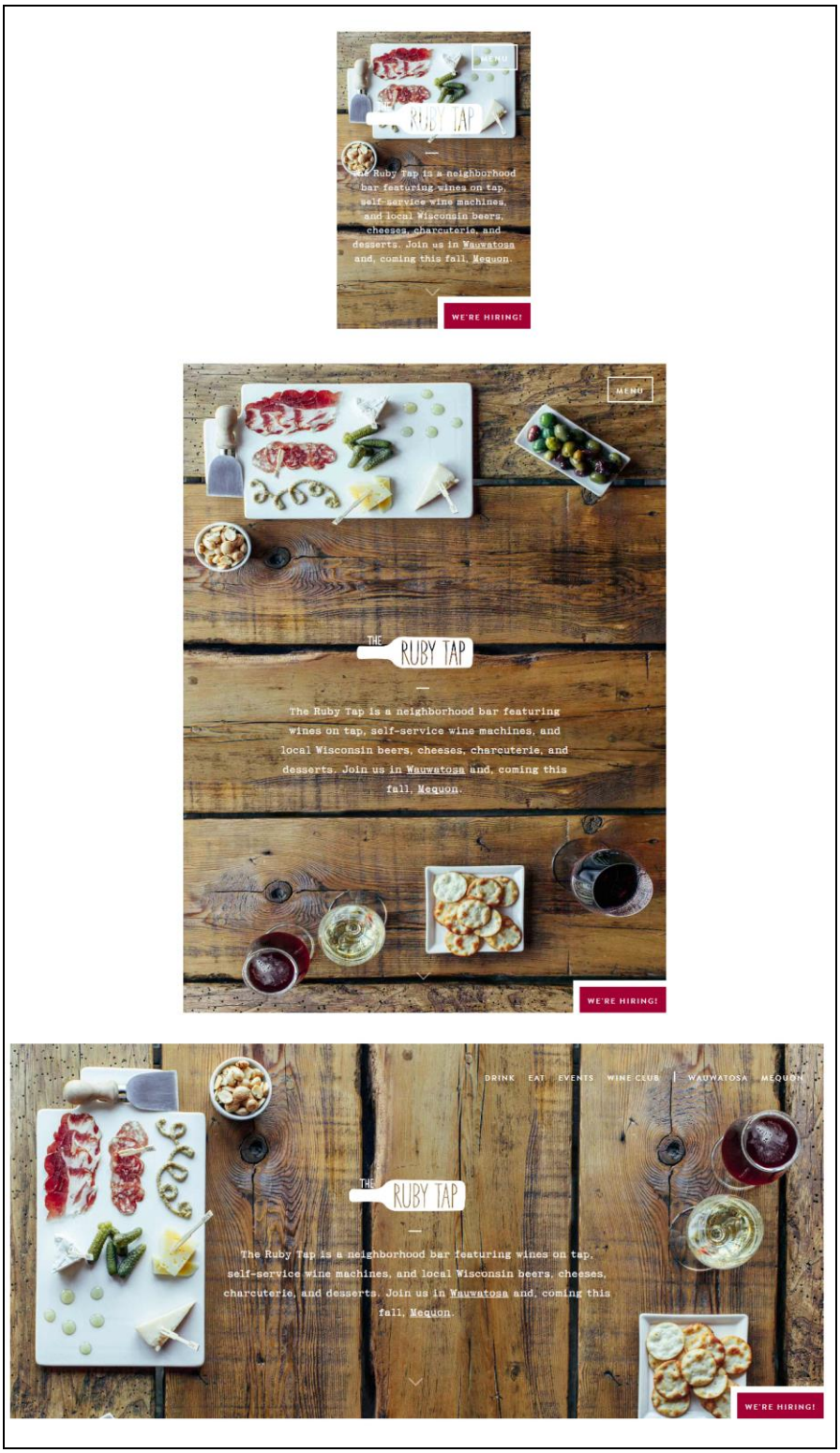
Figura 12: Barra de ferramentas – Plugin Responsive Design View do Mozilla Firefox



Fonte: FIREFOX, 2016

A Figura 11 ilustra o comportamento do website testado nesta ferramenta através de diferentes capturas de tela, através das seguintes resoluções: 320 x 480 pixels, 768 x 1024 pixels e 1280 x 600 pixels.

Figura 13: Capturas de tela - Plugin Mozilla Firefox



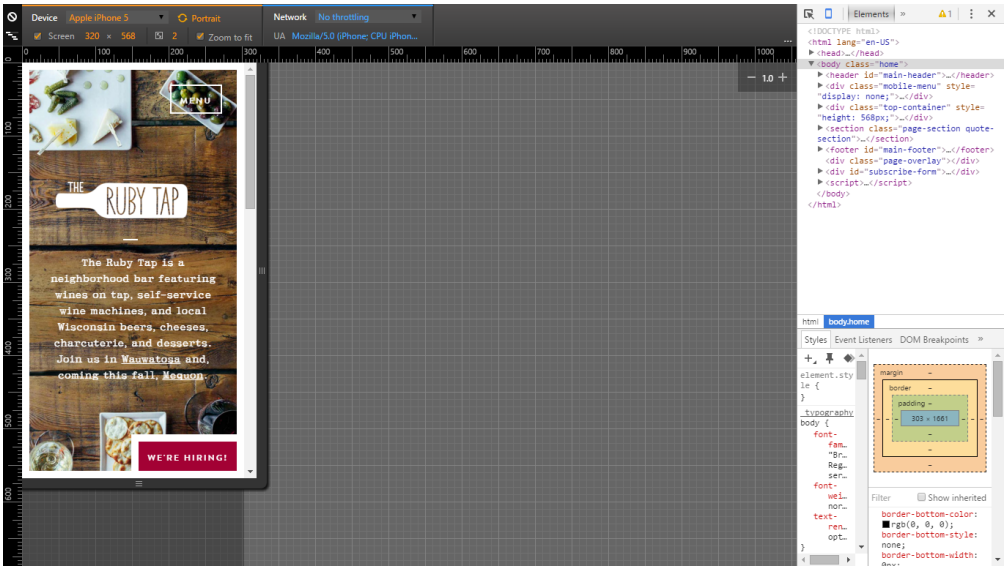
Fonte: ADAPTADA (MOZILLA FIREFOX, 2016)

2.4.2.2 Google Chrome

O *Device Mode*, plugin disponível no navegador Google Chrome fornece réguas auxiliares com medidas em pixels, a simulação de redes de internet, lista com dispositivos reais, orientações no modo paisagem e retrato e também consulta as media queries existentes no design da página.

A Figura 14 ilustra uma visão geral da ferramenta e seus aspectos. Já a sua barra de ferramenta pode ser visualizada de maneira ampliada, com mais detalhes, na Figura 15.

Figura 14: Visão geral - Plugin Device Mode no Google Chrome



Fonte: CHROME (2016)

Figura 15: Barra de ferramentas – Plugin Device Mode no Google Chrome

Fonte: CHROME (2016)

2.5 FRAMEWORKS DE DESIGN RESPONSIVO

A reutilização de códigos permite que ocorra a redução do tempo de desenvolvimento nas aplicações. Existem diversas formas de reuso, entre elas os frameworks. Pode-se dizer que um framework é uma abstração que une vários códigos que são comuns entre vários projetos e que pode assim, prover uma funcionalidade genérica (Müller, 2008).

Um framework permite a reutilização de elementos pré-definidos e fornece componentes que podem ajudar na rápida criação de uma aplicação. Ao invés de começar um projeto do zero, é possível partir de uma base pronta para continuar desenvolvendo a aplicação (Sommer e Krusche, 2013).

Sommer e Krusche (2013) afirmam que um framework multi-plataforma permite reutilizar partes do código-fonte de aplicações para múltiplos dispositivos. Entre os frameworks multi-plataformas existem os que permitem a implementação de layouts para websites, empregando o conceito de design responsivo. Tradicionalmente, eles se baseiam em grids flexíveis que ajudam a organizar os elementos das páginas de maneira responsiva.

Grid é uma malha que divide a tela em partes proporcionais e possibilita a distribuição do conteúdo de forma uniforme, proporcionando equilíbrio visual e estrutural. Construir um Design Responsivo se torna muito mais fácil se ele for apoiado em um Grid Flexível (Pacheco, 2014).

Entre os frameworks de Design Responsivo, podem-se citar o *Bootstrap, Foundation* e o *Skeleton*, que serão detalhados a seguir.

2.5.1 Bootstrap

É um framework para o desenvolvimento de aplicações responsivas através de HTML, CSS e Javascript (Bootstrap, 2015). O Bootstrap foi criado em 2011 por Mark Otto e Jacob Thorntown como solução interna do Twitter para resolver as incompatibilidades existentes nos códigos da equipe de desenvolvimento. Ele surgiu com o objetivo de implementar um padrão na nomenclatura de classes em *CSS* e proporcionar à equipe maior velocidade e efetividade no desenvolvimento dos projetos.

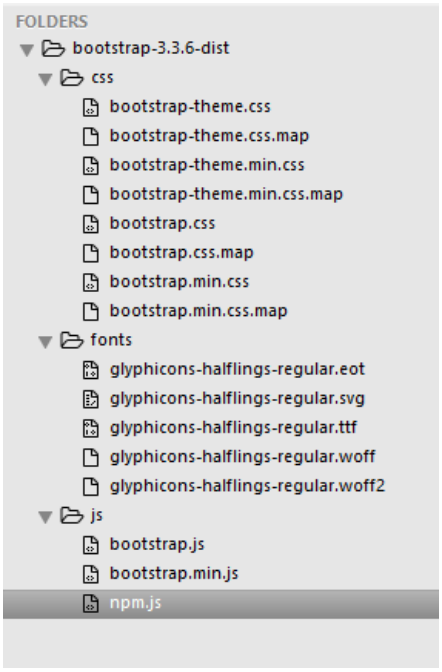
Em agosto de 2011 ele foi publicado no *GitHub*, aonde é hospedado e gerenciado, como um projeto de software-livre. Em pouco tempo seu sucesso fez com que milhares de desenvolvedores contribuíssem para o projeto, que se tornou o mais ativo do ano, fato ocorrido até hoje.

O Bootstrap depende, basicamente, de dois arquivos:

- *bootstrap.css:* arquivo que contém as classes em CSS.
- *bootstrap.js:* arquivo com os códigos de JavaScript.

A estrutura de arquivos que compõem o Bootstrap ao se iniciar um novo projeto pode ser visualizada através da Figura 16.

Figura 16: Estrutura de arquivos do Bootstrap



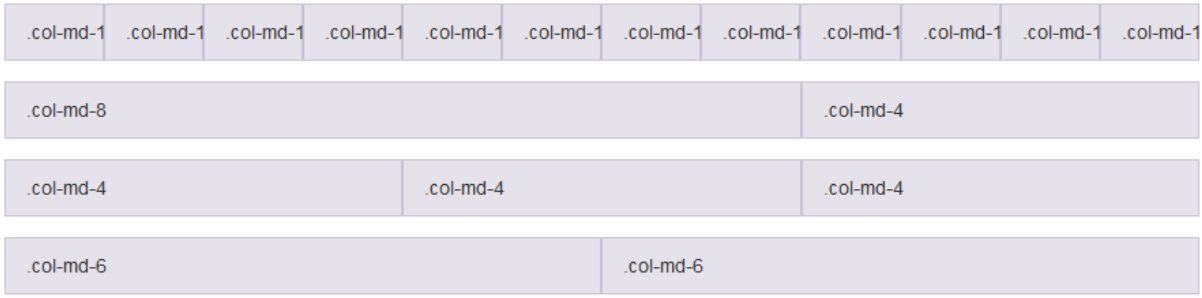
Fonte: DO AUTOR (2016)

Percebe-se que os arquivos se dividem em 3 pastas principais: arquivos em CSS, arquivo de fontes e arquivos JavaScript.

O Bootstrap fornece um grid que divide o layout em até doze colunas, conforme ilustrado na Figura 15, através de quatro resoluções diferentes: dispositivos com resolução menor ou igual à 768px de largura (smartphones), dispositivos com resolução maior ou igual à 768px (tablets), dispositivos com resolução maior que 992px (computadores desktop) e dispositivos com resolução maior do que 1200px (computadores desktop com alta resolução).

A Figura 17 ilustra um exemplo de implementação das colunas no Bootstrap em dispositivos médios, com resolução até 992px.

Figura 17: Implementação do grid do Bootstrap em dispositivos médios



Fonte: BOOTSTRAP (2016)

No Bootstrap, as classes utilizam uma sintaxe que envolve o uso de dois prefixos juntamente ao tamanho desejado da coluna. O prefixo "*col*" refere-se à palavra coluna, o segundo prefixo refere-se ao tamanho do dispositivo e logo após, o número equivalente ao tamanho da coluna. A implementação de uma coluna com a classe "*col-md-8*" significa que ela ocupará um espaçamento de oito células no layout de um dispositivo de tamanho médio.

2.5.2 Foundation

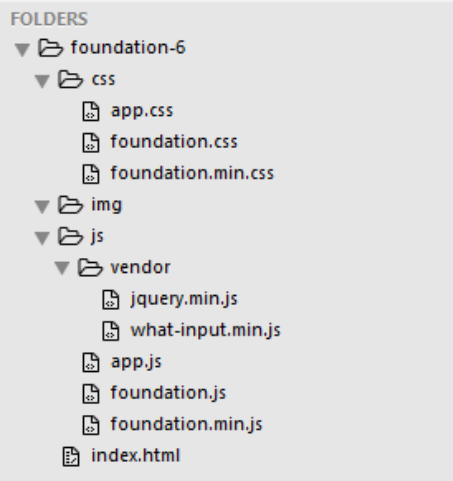
Outro framework para o desenvolvimento de aplicações responsivas através de HTML, CSS e Javascript é o Foundation. Ele teve sua origem em 2008 através do guia de estilos da ZURB, uma companhia de design que desenvolveu o layout de sites como o eBay e o Facebook (Foundation, 2015).

O Guia de Estilos ZURB, que era uma coleção de HTML, CSS e Javascript utilizada nos projetos de cada cliente, foi aprimorado e em 2010 foi solidificado e

nomeado como Foundation. O framework surgiu buscando aumentar a velocidade de desenvolvimento da equipe interna (Foundation, 2015).

O Foundation, atualmente na sua sexta versão, conta com a seguinte estrutura de arquivos por padrão, ilustrada na Figura 18.

Figura 18: Estrutura de arquivos do Foundation



Fonte: DO AUTOR (2016)

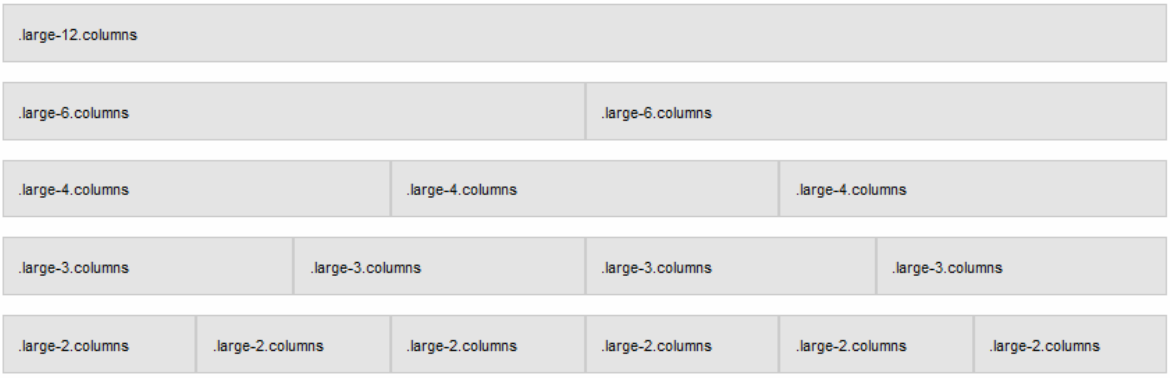
O Foundation não possui arquivos de fontes, apenas os arquivos contendo as classes em CSS e os scripts em JavaScript. Ele conta com uma pasta vazia chamada *img,* para organizar as possíveis imagens que serão utilizadas em um novo projeto.

Também caracterizado por fornecer um grid com até 12 colunas para organizar o conteúdo, o Foundation possibilita a organização do layout através das diferentes resoluções de tela:

- Dispositivos pequenos: 0px até 640px.
- Dispositivos médios: 641px até 1024px.
- Dispositivos grandes: maiores que 1025px.

A Figura 19 ilustra uma exemplificação de um grid implementado através do framework Foundation em dispositivos grandes.

Figura 19: Implementação do grid do Foundation em dispositivos grandes



Fonte: FOUNDATION (2016)

A classe *.large-6.columns* caracteriza que a coluna ocupará um espaço de 6 células (metade do layout) em um dispositivo grande (*large*).

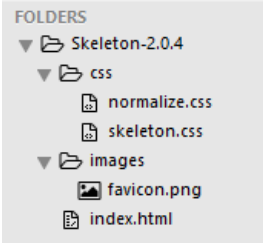
2.5.3 Skeleton

É um framework formado por um conjunto de folhas em CSS, desenvolvido por Dave Gamache em 2011. Segundo Arthur Silva (2014) o Skeleton é uma coleção de arquivos CSS que auxiliam no desenvolvimento de sites para qualquer resolução de tela, seja um monitor de 17 polegadas ou a tela de um smartphone.

O Skeleton consiste em um grid responsivo aonde suas colunas possuem até 960 pixels de largura e contém arquivos CSS para a implementação da tipografia, botões, tabelas, listas e campos de texto (Skeleton, 2015). Como o próprio nome sugere, *Skeleton*, da língua inglesa, significa esqueleto. Nesse caso, ele é um framework que tem seu foco no esqueleto da página, através da implementação de seus grids.

Capaz de organizar o seu conteúdo também através de 12 colunas, o Skeleton possui a seguinte estrutura de arquivos, ilustrada na Figura 20.

Figura 20: Estrutura de arquivos do Skeleton

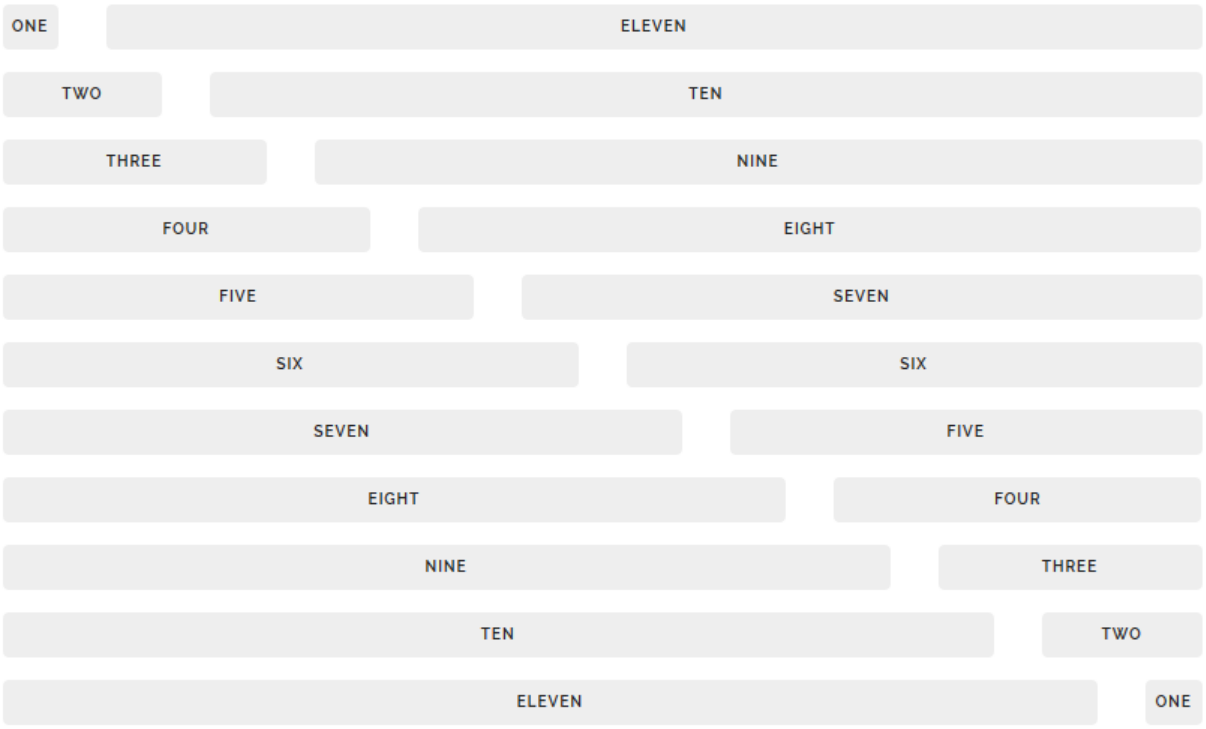


Fonte: DO AUTOR (2016)

Percebe-se que o Skeleton é formado por arquivos em CSS: *normalize.css* e o skeleton.css. Contém também um ícone por padrão dentro da pasta de imagens e uma página padrão em HTML chamada index.html.

Para a implementação do grid de colunas, é preciso apenas utilizar o número equivalente ao número de colunas em inglês, juntamente a classe *"columns"*. A Figura 21, a seguir, apresenta o grid implementado na própria página do framework, que tem sua codificação ilustrada na Figura 22.

Figura 21: Grid de colunas implementado através do framework Skeleton



Fonte: SKELETON (2016)

Figura 22: Codificação e estrutura HTML do grid exemplificado no framework Skeleton

```
<div class="example-grid docs-example">
 <div class="row">
  <div class="one column">One</div>
  <div class="eleven columns">Eleven</div>
 </div>
 <div class="row">
  <div class="two columns">Two</div>
  <div class="ten columns">Ten</div>
 </div>
 <div class="row">
  <div class="three columns">Three</div>
  <div class="nine columns">Nine</div>
 </div>
 <div class="row">
  <div class="four columns">Four</div>
  <div class="eight columns">Eight</div>
 </div>
 <div class="row">
  <div class="five columns">Five</div>
  <div class="seven columns">Seven</div>
 </div>
 <div class="row">
  <div class="six columns">Six</div>
  <div class="six columns">Six</div>
 </div>
 <div class="row">
  <div class="seven columns">Seven</div>
  <div class="five columns">Five</div>
 </div>
 <div class="row">
  <div class="eight columns">Eight</div>
  <div class="four  columns">Four</div>
 </div>
 <div class="row">
  <div class="nine columns">Nine</div>
  <div class="three columns">Three</div>
 </div>
 <div class="row">
  <div class="ten columns">Ten</div>
  <div class="two columns">Two</div>
 </div>
 <div class="row">
  <div class="eleven columns">Eleven</div>
  <div class="one column">One</div>
 </div>
</div>
```

Fonte: DO AUTOR (2016)

3 METODOLOGIA

Neste capítulo são descritas as etapas da metodologia empregada no desenvolvimento deste trabalho.

3.1 PROJETO DO WEBSITE

O website proposto como ambiente de testes para o estudo dos frameworks é um portal de notícias de filmes chamado *Filmamex*. Este apresenta vários formatos de conteúdo, como textos, imagens e vídeos.

A organização dos conteúdos é disposta através de cinco páginas: uma página inicial, uma página de notícias, trailers, filmes em cartaz e uma página com um formulário para contato. Para o projeto da interface das páginas, são empregadas as convenções web (Seção 2.2) de navegação global, marca, barra de busca, navegação local e rodapé. Presentes em todas as páginas, essas convenções são responsáveis por criar consistência no website.

Para cada página projetada, foi criado um esboço ilustrando a organização do conteúdo e das convenções web utilizadas. Assim, os esboços projetados são detalhados na seção a seguir.

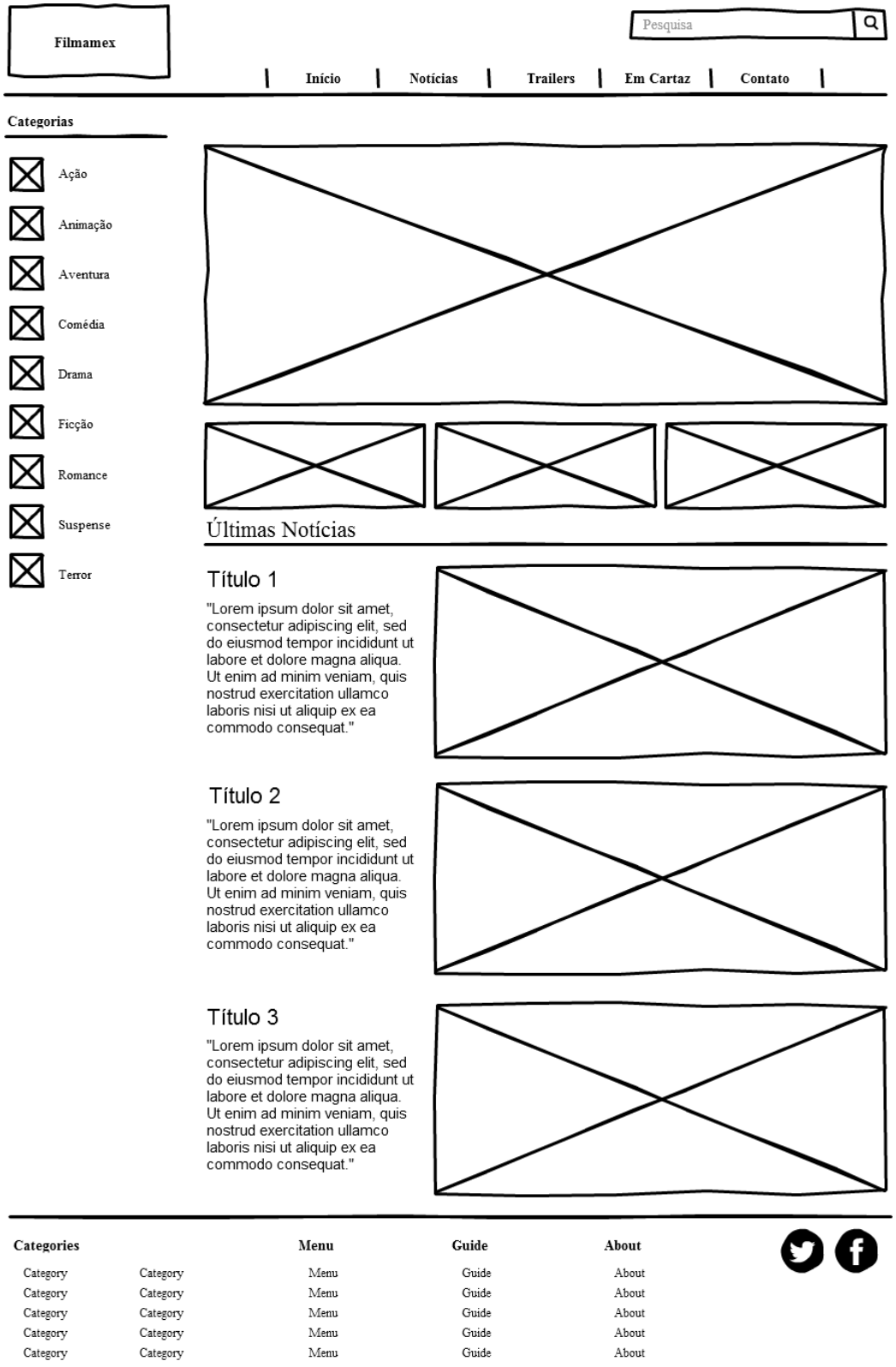
3.1.1 Esboços Projetados

Baseando-se no projeto do website proposto foram criados esboços, ilustrados na Figura 23 até a Figura 27, para cada página do website:

- Inicial
- Notícias
- Trailers
- Filmes em cartaz
- Contato

Na página inicial, estão dispostos banners informando as novidades que foram inseridas no website, como novos trailers, filmes em cartaz e as últimas notícias. A marca, os menus e o campo de busca são elementos presentes da mesma forma nas demais páginas. A Figura 23 ilustra o esboço da página inicial.

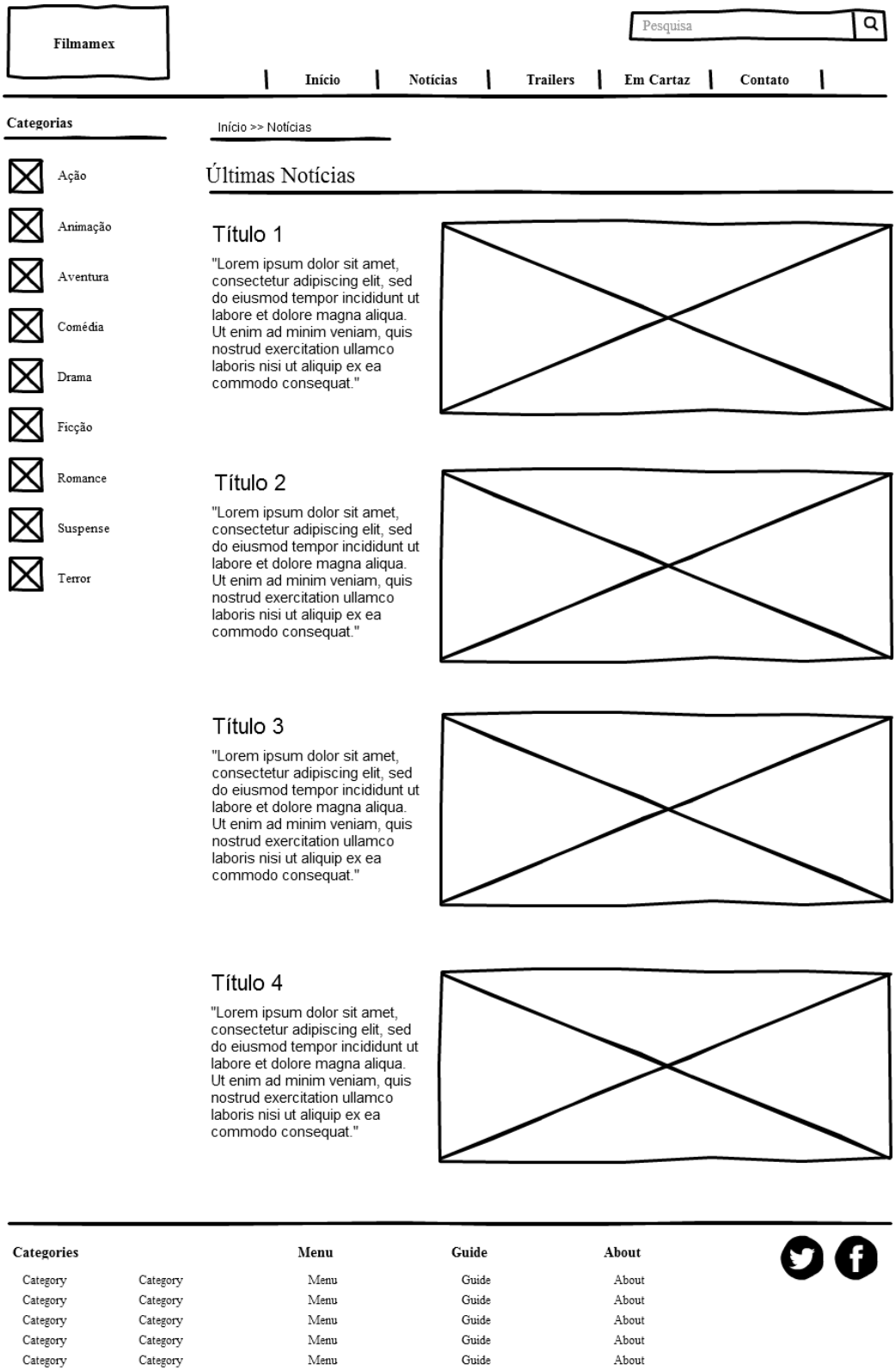
Figura 23: Filmamex - Página Inicial



Fonte: DO AUTOR (2016)

Cada notícia possui um título, um breve resumo e uma figura ilustrando-a ao lado. A Figura 24 ilustra o esboço da página de notícias.

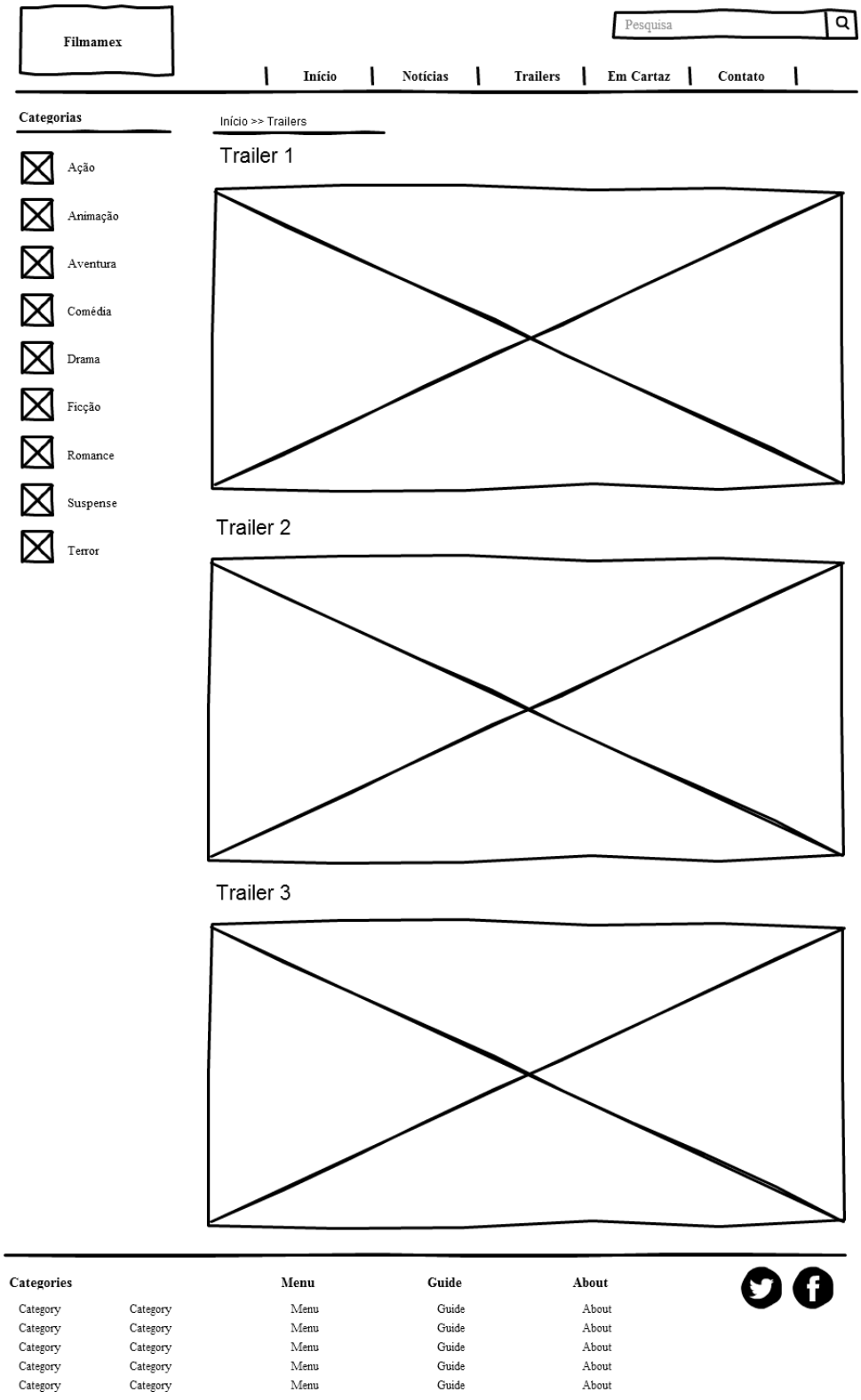
Figura 24: Filmamex - Notícias



Fonte: DO AUTOR (2016)

Nos trailers estão dispostos os vídeos e seus respectivos títulos acima. A Figura 25 ilustra o esboço da página de trailers.

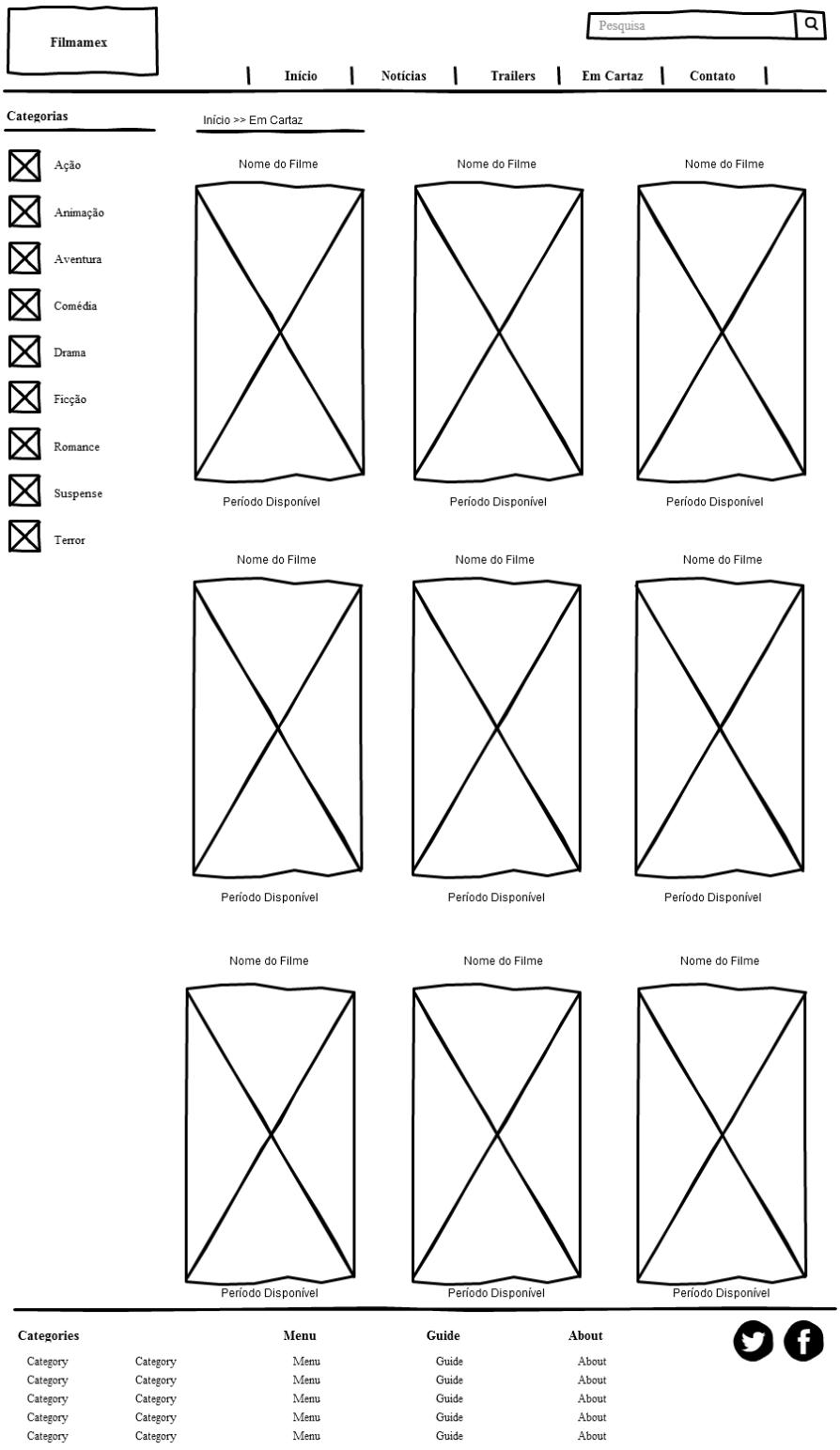
Figura 25: Filmamex - Trailers



Fonte: DO AUTOR (2016)

Nos filmes em cartaz, são apresentados os banners dos filmes com título e o período em que se manterá em cartaz. A Figura 26 ilustra o esboço da página de filmes em cartaz.

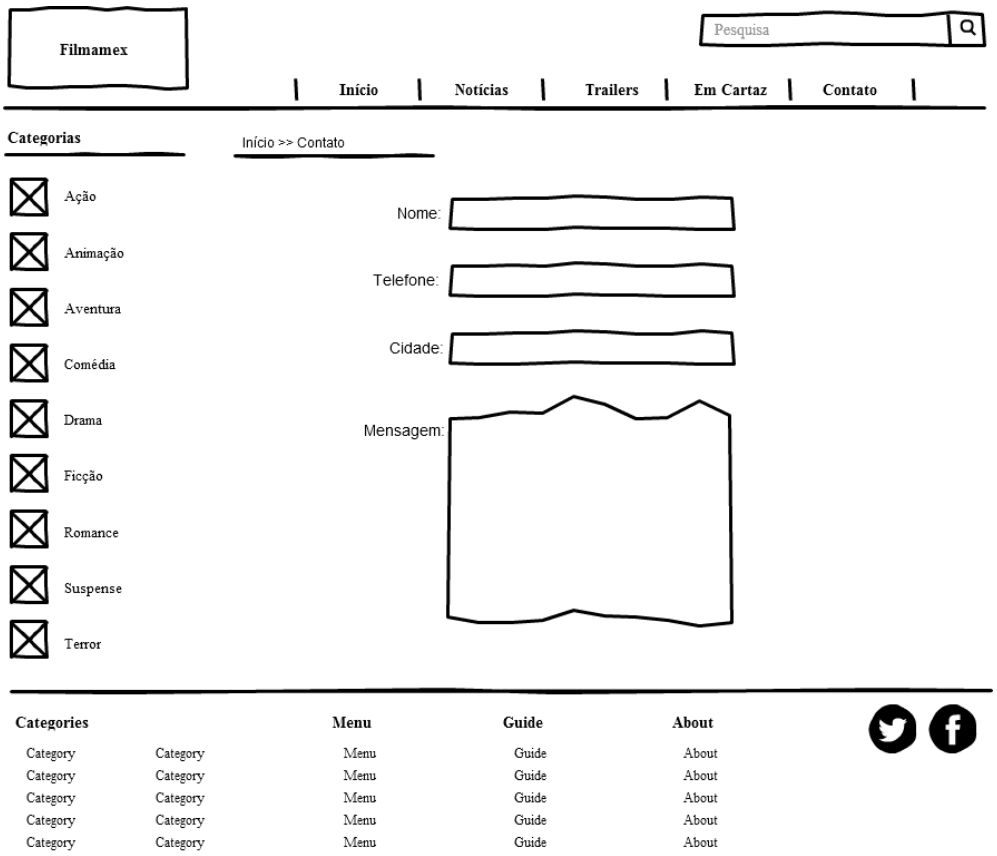
Figura 26: Filmamex - Filmes em Cartaz



Fonte: DO AUTOR (2016)

O contato é apresentado através de um formulário contendo campos para a inserção de nome, telefone, cidade e mensagem que o usuário desejar. A Figura 27 ilustra um esboço da página de contato.

Figura 27: Filmamex - Contato



Fonte: DO AUTOR (2016)

3.2 IMPLEMENTAÇÃO DO WEBSITE

O website Filmamex foi implementado em cada um dos frameworks escolhidos: Bootstrap, Foundation e Skeleton. Esses são licenciados pelo *MIT* (Instituto de Tecnologia de Massachusetts).

Três versões do website foram criadas: uma para cada framework. A codificação foi realizada no editor de texto *Sublime Text 3,* ferramenta de desenvolvimento presente no computador pessoal do autor e as linguagens utilizadas foram o HTML, em sua quinta versão, e o CSS, na sua terceira versão.

3.3 AVALIAÇÃO DOS FRAMEWORKS

A partir das três versões implementadas do website Filmamex, foram avaliados alguns aspectos como a construção do layout, a implementação, os recursos que são fornecidos, o emprego do design responsivo e a customização de componentes através dos frameworks de design responsivo estudados. Essas avaliações são descritas nas seções a seguir.

3.3.1 Construção do layout

Os frameworks Bootstrap, Foundation e Skeleton se assemelham na implementação do grid responsivo, possibilitando a divisão do layout em até doze colunas. Assim, a construção do layout foi avaliada através das classes que são utilizadas na implementação das colunas em relação a diferentes resoluções.

3.3.2 Implementação

A página inicial do website Filmamex foi designada para a avaliação da implementação, evitando a comparação de componentes que se repetem em outras páginas, como o cabeçalho, a navegação local e o rodapé.

Sommerville (2011) cita algumas métricas de software através do comprimento dos códigos. Geralmente, quando maior o tamanho do código de um componente, mais complexo e sujeito a erros ele tende a ser. O comprimento de código tem sido uma das métricas mais confiáveis para prever a incidência de erros em componentes.

Com base nessas premissas, o tamanho do código HTML utilizado na construção das páginas foi avaliado. Na linguagem HTML, embora o código possa ser escrito em uma única linha, o número de tags e o número de classes empregadas para estilização refletem na complexidade de implementação. Assim, foram contabilizadas as tags HTML e classes CSS empregadas no desenvolvimento da página inicial.

3.3.3 Recursos fornecidos

A utilização das linguagens HTML e CSS pelos frameworks permite a implementação de componentes prontos e já estilizados, os quais nesse trabalho são denominados como recursos fornecidos. Desta forma, foi realizado o levantamento dos componentes de interface necessários para a implementação do layout do website e quais frameworks foram capazes de fornecê-los.

3.3.4 Design Responsivo

A responsividade do layout do website implementado foi avaliada através dos dispositivos físicos e ferramentas como os plug-ins de navegadores e o website Reposinator, analisando o comportamento da interface gráfica através de diferentes resoluções.

3.3.5 Customização

Ao utilizar os componentes dos frameworks de design responsivo, a identidade visual do website apresenta as características definidas por padrão em cada framework. A necessidade de modificar esses componentes para atingir o aspecto customizado exige a modificação de atributos nas classes CSS de cada framework.

Para identificar o nível de complexidade de customização dos componentes de interface em cada framework, foi avaliado a quantidade de classes e atributos necessários ao modificar um determinado componente de um framework. Assim, com base nesses aspectos, a avaliação da customização é realizada através da modificação dos elementos que compõem uma barra de busca, conforme mostra a Figura 28.

Figura 28: Modelo de barra de busca

Fonte: DO AUTOR (2016)

Alguns frameworks, como o Bootstrap e Foundation, permitem a customização de componentes antes mesmo de serem baixados. Entretanto, nessa avaliação será levada em consideração a customização de componentes da distribuição padrão.

4 RESULTADOS

Neste capítulo são apresentados os resultados obtidos através da aplicação da metodologia proposta no Capítulo 3. Os resultados foram divididos nas sessões de construção do layout (Seção 4.1), implementação (Seção 4.2), recursos fornecidos (Seção 4.3), design responsivo (Seção 4.4) e customização (Sessão 4.5).

4.1 CONSTRUÇÃO DO LAYOUT

A avaliação da construção do layout foi baseada na implementação do grid responsivo, o qual é caracterizado pelas classes que são utilizadas na construção de suas colunas.

A lógica de implementação do grid na construção de um layout através dos frameworks Bootstrap, Foundation e Skeleton é similar. As classes que podem ser empregadas na implementação das colunas geralmente contêm o termo referente ao tamanho do dispositivo, junto ao número de colunas em que o layout será dividido.

Na Tabela 3 pode-se visualizar a faixa de resolução equivalente a cada dispositivo no qual se permite, em cada framework, implementar a divisão do layout. Esses dispositivos foram classificados em quatro diferentes categorias como extra pequenos, pequenos, médios e grandes.

Tabela 3: Faixas de resoluções equivalentes aos dispositivos suportados em cada framework

Framework	**Dispositivos extra pequenos**	**Dispositivos pequenos**	**Dispositivos médios**	**Dispositivos grandes**
Bootstrap	< 768 pixels de largura	>= 768 pixels de largura	>= 992 pixels de largura	>= 1200 pixels de largura
Foundation	S/ Especificação	< 640 pixels de largura	>= 640 pixels de largura	>= 1024 pixels de largura
Skeleton	S/ Especificação	S/ Especificação	S/ Especificação	S/ Especificação

Fonte: DO AUTOR (2016)

Através da análise da Tabela 3, percebe-se que o Bootstrap é o framework com o maior controle sobre a implementação do layout, permitindo a manipulação de suas colunas em até quatro faixas de resoluções distintas. Já o Foundation, permite a implementação das colunas em até três resoluções e o Skeleton, por sua vez, não permite com que seu grid seja controlado, fazendo todo redimensionamento da interface automaticamente.

Já na Tabela 4, são exemplificadas as classes utilizadas pelos frameworks para a implementação de uma e duas colunas, referenciando o tamanho dos dispositivos suportados.

Tabela 4: Classes utilizadas na implementação do grid através dos Frameworks

Framework	**Número de Colunas**	**Classes – Dispositivos Extra Pequenos**	**Classes – Dispositivos Pequenos**	**Classes – Dispositivos Médios**	**Classes – Dispositivos Grandes**
Bootstrap	1	col-xs-1	col-sm-1	col-md-1	col-lg-1
	2	col-xs-2	col-sm-2	col-md-2	col-lg-2
Foundation	1	-	small-1 columns	medium-1 columns	large-1 columns
	2	-	small-2 columns	medium-2 columns	large-2 columns
Skeleton	1	-	-	-	one columns
	2	-	-	-	two columns

Fonte: DO AUTOR (2016)

Na Tabela 4, percebe-se que nas classes utilizadas pelo Bootstrap, são utilizados prefixos da palavra coluna e do tamanho do dispositivo, ambos derivados da língua inglesa. Juntamente ao número, referente ao tamanho, é construído a sintaxe de classes que podem ser empregadas na construção de suas colunas.

No Foundation, percebe-se classes mais sugestivas ao evitar o uso de prefixos. Sua sintaxe utiliza o termo referente ao tamanho do dispositivo na língua inglesa e o tamanho da coluna a ser implementada.

O Skeleton também apresenta classes mais sugestivas ao exigir somente o número referente ao tamanho de suas colunas na língua inglesa. Entretanto, ele e o

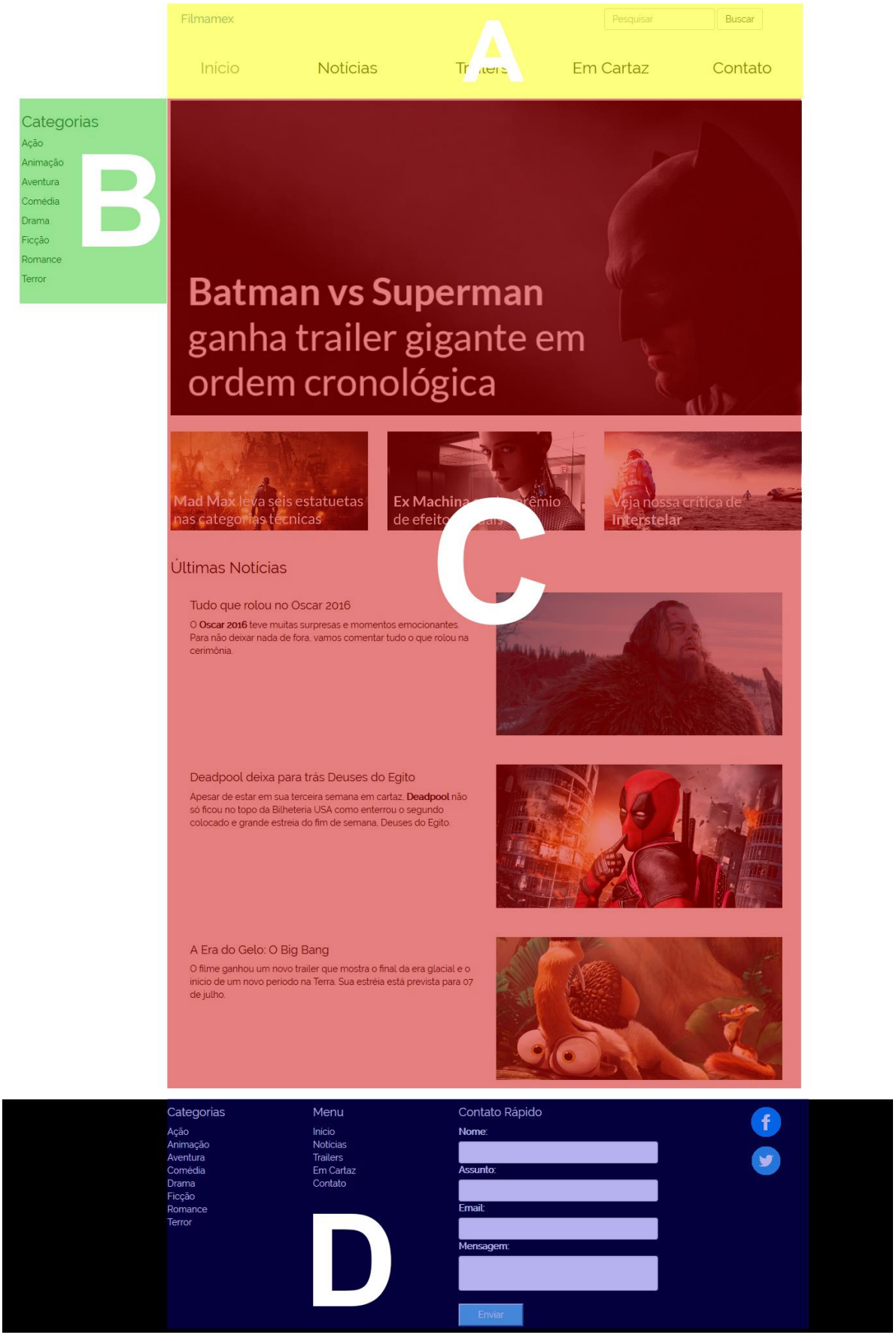
Foundation exigem a adição da classe *columns* na implementação de suas colunas, diferentemente do Bootstrap.

4.2 IMPLEMENTAÇÃO

A avaliação da implementação do website Filmamex através dos frameworks Bootstrap, Foundation e Skeleton é apresentada nessa seção. Para evitar a comparação de componentes que se repetem em todas as páginas do website, como o cabeçalho, a navegação local e o rodapé, somente a página inicial foi designada para essa avaliação. Sua divisão em quatro módulos, conforme mostra a Figura 29, é descrita a seguir:

- Cabeçalho (Módulo A): contém o logotipo do website, ferramenta de busca e o menu principal.
- Navegação local (Módulo B): contém a navegação para as categorias de páginas através de gêneros de filmes.
- Conteúdo (Módulo C): apresenta os últimos conteúdos postados, acompanhados de notícias.
- Rodapé (Módulo D): contém links de navegação, um formulário de contato rápido e links para redes sociais.

Figura 29: Página inicial do website Filmamex

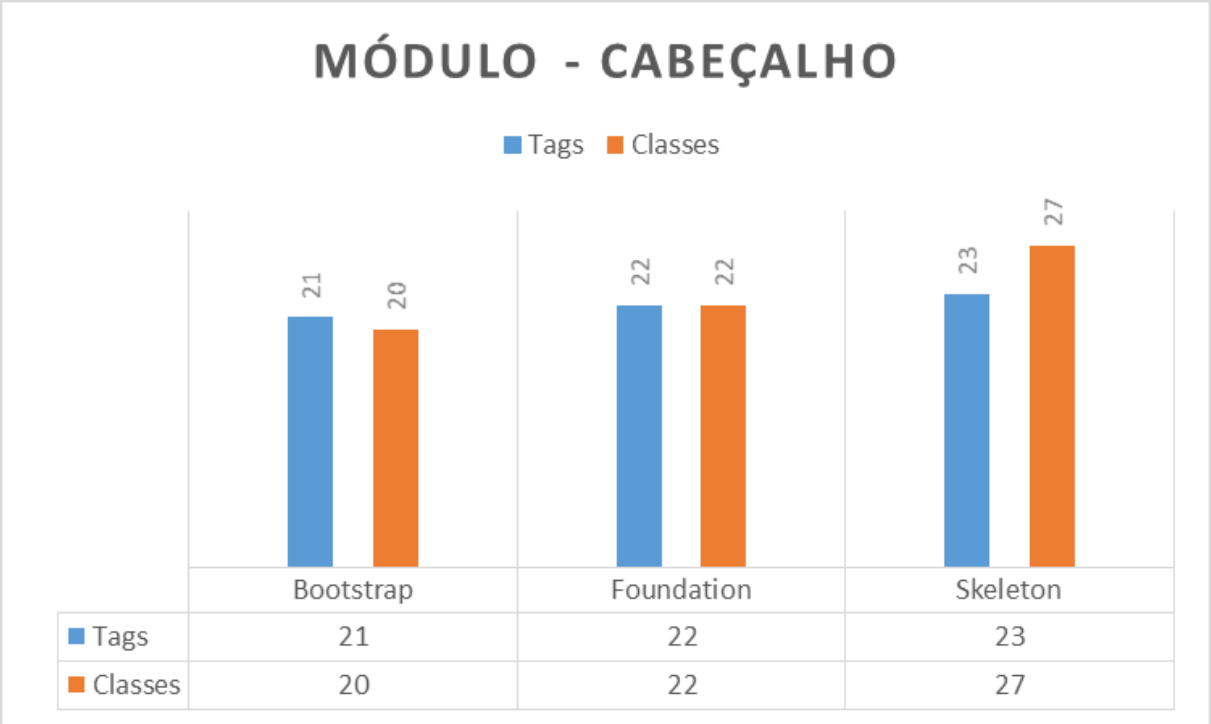


Fonte: DO AUTOR (2016)

A fim de identificar o nível de complexidade de implementação dos módulos descritos, foram contabilizadas a quantidade de tags HTML e classes CSS empregadas na construção de cada módulo.

Os resultados da implementação do módulo do cabeçalho estão ilustrados na Figura 30. Percebe-se que o Skeleton leva desvantagem na implementação desse módulo. Isso justifica-se porque diferente do Bootstrap e do Foundation, ele não fornece alguns recursos prontos como menus, logotipos e barras de busca. Assim, de modo que esses componentes fossem criados de forma similar às outras versões implementadas, foi necessário implementar classes e tags adicionais. O Bootstrap e o Foundation, capazes de fornecer esses recursos, equivaleram-se na implementação, apresentando um número de tags e classes aproximado.

Figura 30: Avaliação da implementação do cabeçalho na página inicial

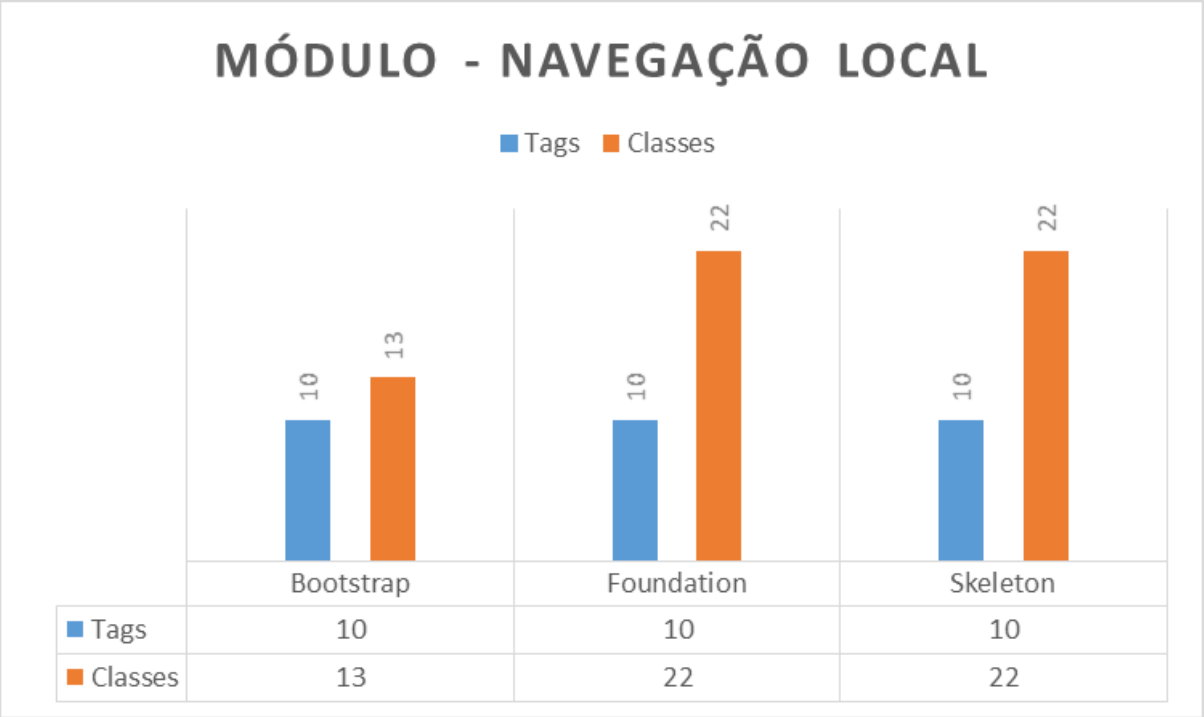


Fonte: DO AUTOR (2016)

A implementação da navegação local através de cada framework teve seus resultados ilustrados na Figura 31. A navegação local apresentou a mesma quantidade de tags HTML utilizadas em sua construção em todos os frameworks. Isso ocorre porque ela é implementada através de uma lista de links composta da mesma estrutura em todas as versões.

Na quantidade de classes utilizadas nesse módulo, percebe-se uma grande vantagem do Bootstrap. Diferentemente do Foundation e Skeleton, ele não necessita da adição da classe *columns* na implementação das colunas. Assim, Foundation e Skeleton tiveram um número equivalente de classes empregadas.

Figura 31: Avaliação da implementação da navegação na página inicial

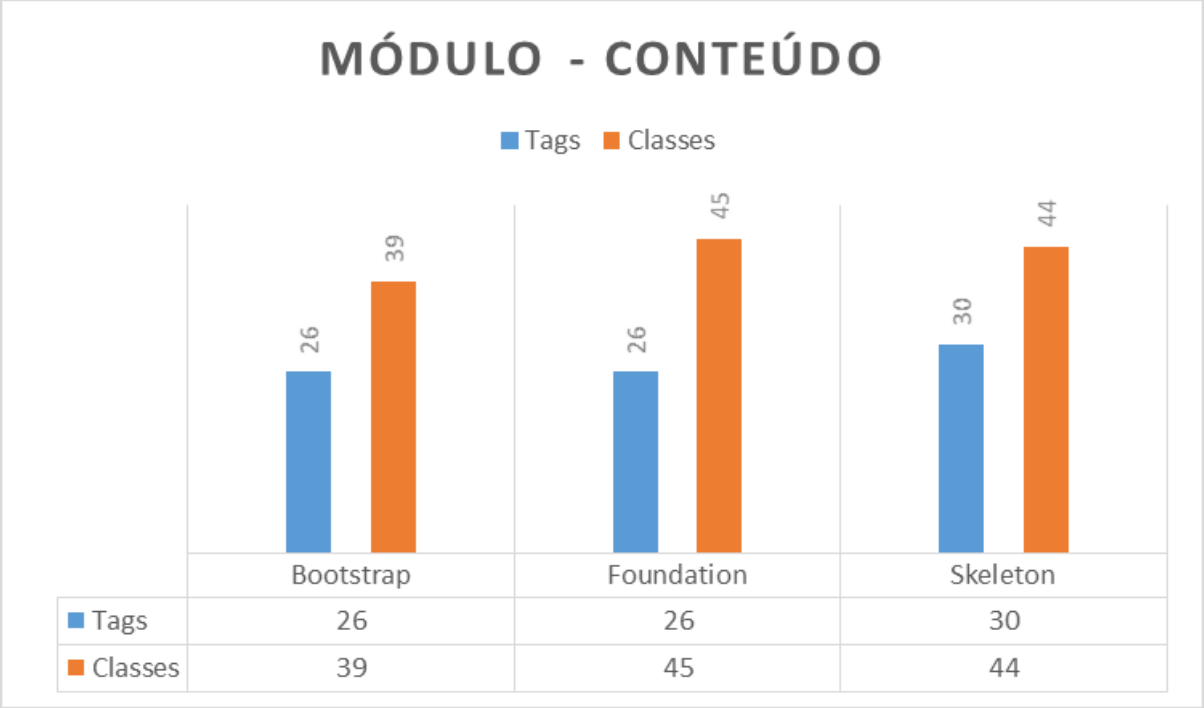


	Bootstrap	Foundation	Skeleton
Tags	10	10	10
Classes	13	22	22

Fonte: DO AUTOR (2016)

Os resultados da implementação do módulo do conteúdo são ilustrados na Figura 32 e detalhados a seguir. Assim como no resultado apresentado anteriormente, o Bootstrap leva vantagem ao implementar esse módulo, pois utiliza menos classes em relação aos demais frameworks, uma vez que não exige a adição da classe *column* ao implementar suas e da classe *row* ao alinhar o conteúdo em linha. Em relação ao número de tags, obteve-se uma quantidade equivalente entre Bootstrap e Foundation.

Figura 32: Avaliação da implementação do conteúdo na página inicial



Fonte: DO AUTOR (2016)

O Skeleton exigiu uma maior quantidade de tags para a construção deste módulo. Isso se justifica porque foram necessárias divs adicionais para dividir o conteúdo em linha nas postagens principais, conforme apresenta a Figura 33.

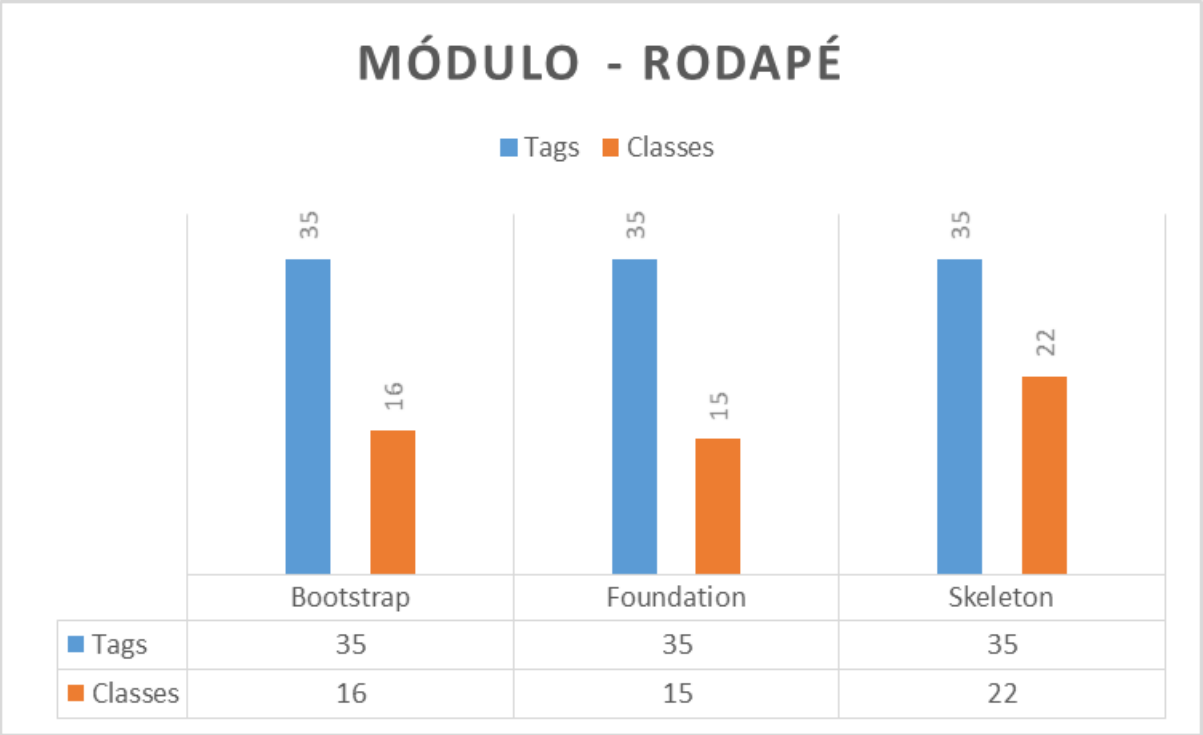
Figura 33: *Divs* adicionais utilizadas na implementação do conteúdo através do Framework Skeleton

```
<div class="row">
  <div class="four columns">
    <!-- Primeira postagem principal -->
    <a href="#"><img class="twelve columns banner" src="images/notice-1.jpg"></a>
  </div>
  <div class="four columns">
    <!-- Segunda postagem principal -->
    <a href="#"><img class="twelve columns banner" src="images/notice-2.jpg"></a>
  </div>
  <div class="four columns">
    <!-- Terceira postagem principal -->
    <a href="#"><img class="twelve columns banner" src="images/notice-3.jpg"></a>
  </div>
</div>
```

Fonte: DO AUTOR (2016)

Na implementação do módulo do rodapé, percebe-se que o número de tags utilizadas foram equivalentes em todos frameworks. Já no número de classes, o Skeleton exigiu uma maior quantidade em relação ao Foundation e o Bootstrap. A ilustração do gráfico com os resultados da implementação desse módulo está na Figura 34.

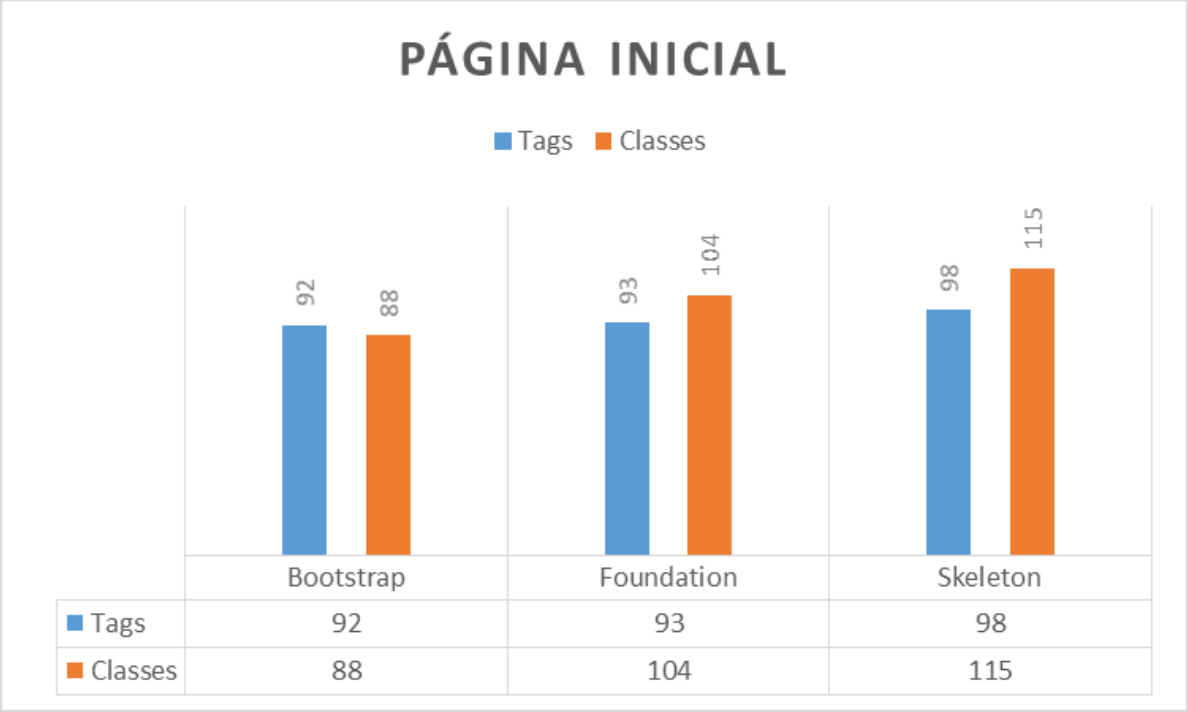
Figura 34: Avaliação da implementação do rodapé na página inicial



Fonte: DO AUTOR (2016)

A partir dos resultados analisados através da implementação de cada módulo, um gráfico com o número total de tags HTML e classes em CSS utilizadas na construção da página inicial foi gerado. A partir deste, ilustrado na Figura 35, é possível ter uma visão geral da implementação através de cada framework.

Figura 35: Resultado final da implementação da página inicial



Fonte: DO AUTOR (2016)

O Bootstrap levou vantagem ao exigir menos classes e tags na sua implementação. Isso pode ser justificado porque, mesmo possuindo uma sintaxe de classes menos sugestivas ao desenvolvedor, ele não necessita a adição da classe *columns* ao implementar suas colunas na divisão do conteúdo, diferentemente do Foundation e Skeleton. Assim, quanto mais colunas forem necessárias para a implementação de um layout, mais eficiente o Bootstrap pode ser nesse aspecto.

4.3 RECURSOS FORNECIDOS

Nessa seção são avaliados os recursos fornecidos por padrão através dos frameworks, que fazem do uso das linguagens HTML e CSS para a implementação de seus componentes. Assim, serão verificados os frameworks que satisfazem a necessidade de recursos a serem implementados para a construção do layout do website. A Tabela 5 apresenta os recursos e os respectivos frameworks que os forneceram.

Tabela 5: Recursos fornecidos através de cada framework estudado

	Bootstrap	Foundation	Skeleton
Logotipo	**X**	**X**	
Ferramenta de Busca	**X**	**X**	
Menu	**X**	**X**	
Botões	**X**	**X**	**X**
Campos de Texto	**X**	**X**	**X**
Imagens Responsivas	**X**	**X**	**X**
Vídeos Responsivos	**X**	**X**	
TOTAL	7	7	3

Fonte: DO AUTOR (2016)

O Skeleton foi o framework com menos recursos fornecidos para a implementação de sua versão do website. Alguns componentes tiveram que ser implementados manualmente como, por exemplo, menus e vídeos responsivos. Esses aspectos serão melhores detalhados na Seção de avaliação do design responsivo, a seguir.

4.4 DESIGN RESPONSIVO

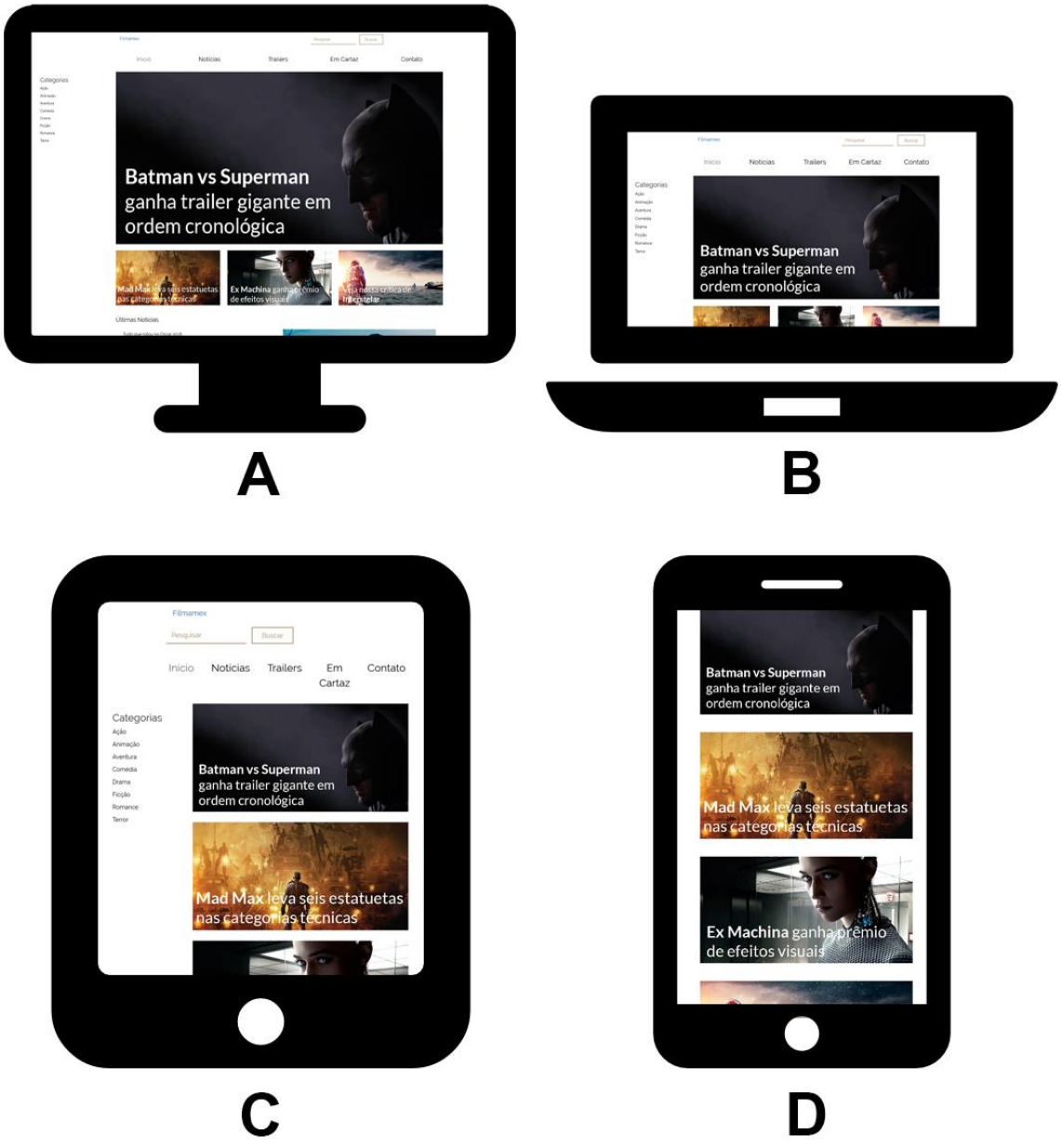
Nessa seção são avaliados os aspectos responsivos em cada framework através das ferramentas de teste como o website Reposinator, os plug-ins dos navegadores e também dos dispositivos físicos.

Após a implementação do website através dos frameworks Bootstrap, Foundation e Skeleton ser concluída, foi avaliado o comportamento do design responsivo empregado nas páginas construídas, verificando-se a organização do conteúdo e erros de layout através de quatro diferentes resoluções.

A página inicial, implementada através dos frameworks Bootstrap, Foundation e Skeleton, é ilustrada na Figuras 36, 37 e 38 respectivamente, seguindo a esquematização de resoluções de tela citadas abaixo:

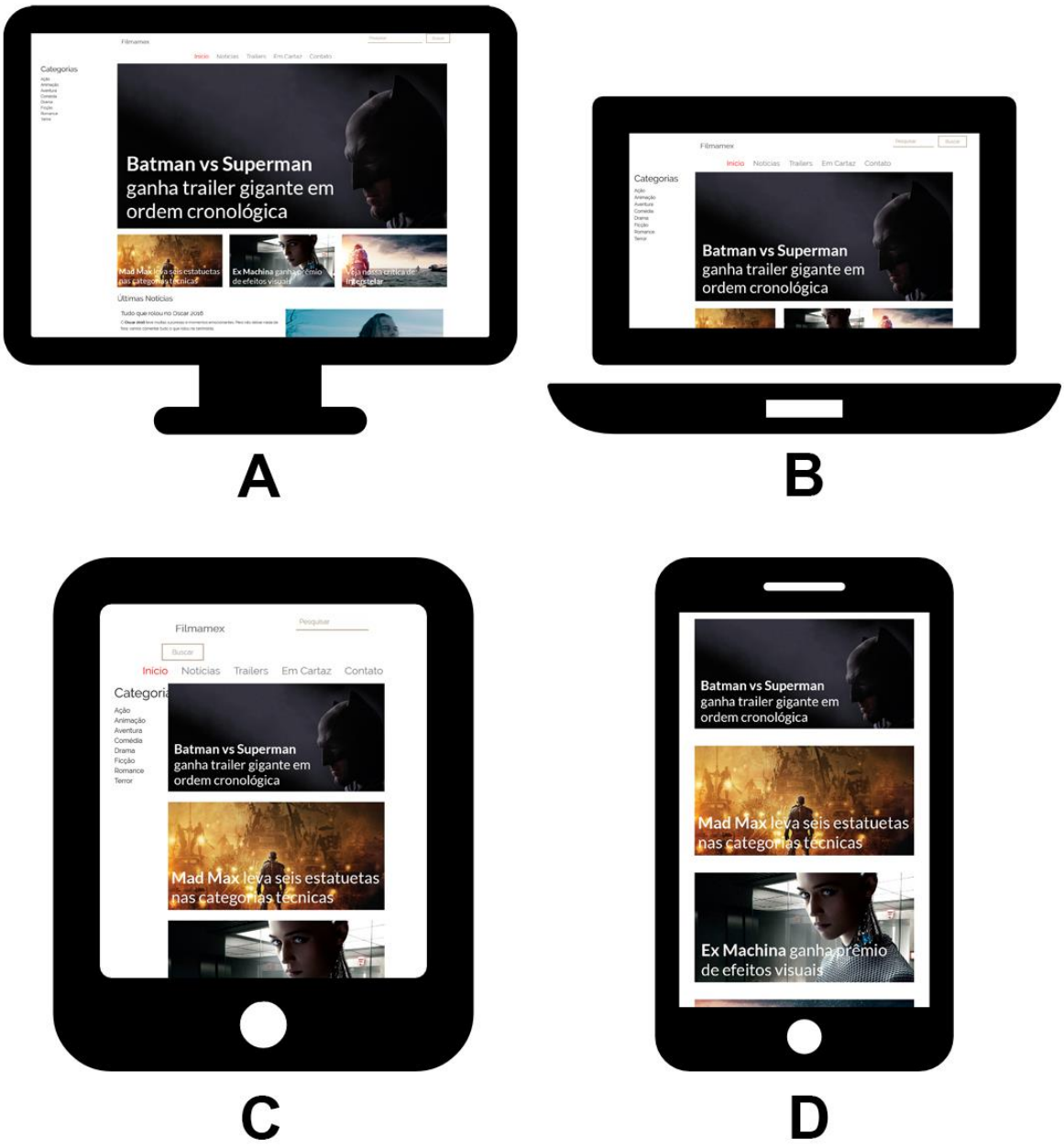
- A: Dispositivos com 1920 pixels de largura.
- B: Dispositivos com 1280 pixels de largura.
- C: Dispositivos com 768 pixels de largura.
- D: dispositivos com 360 pixels de largura.

Figura 36: Página Inicial implementada através do framework Bootstrap



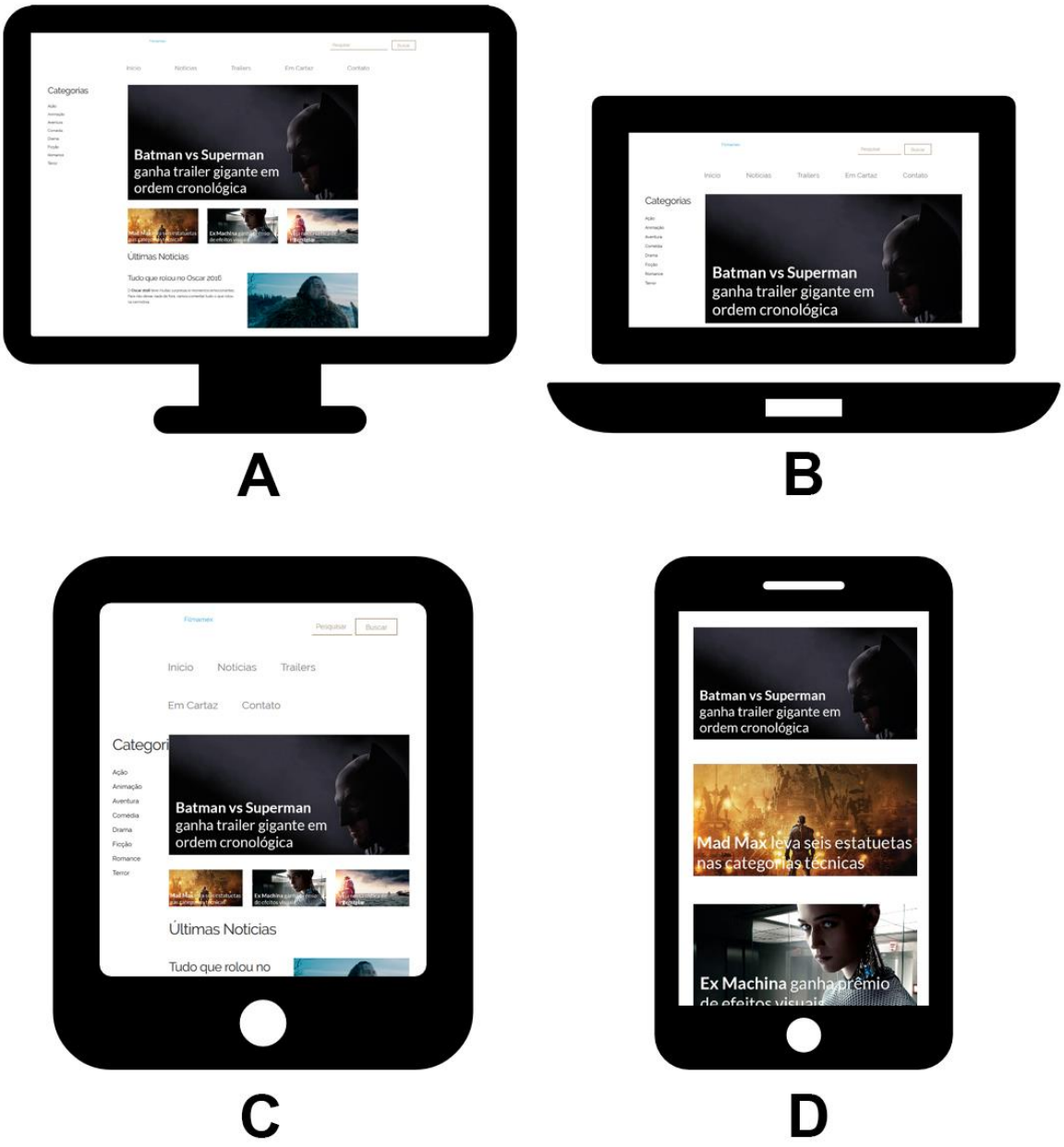
Fonte: DO AUTOR (2016)

Figura 37: Página Inicial implementada através do framework Foundation



Fonte: DO AUTOR (2016)

Figura 38: Página Inicial implementada através do framework Skeleton

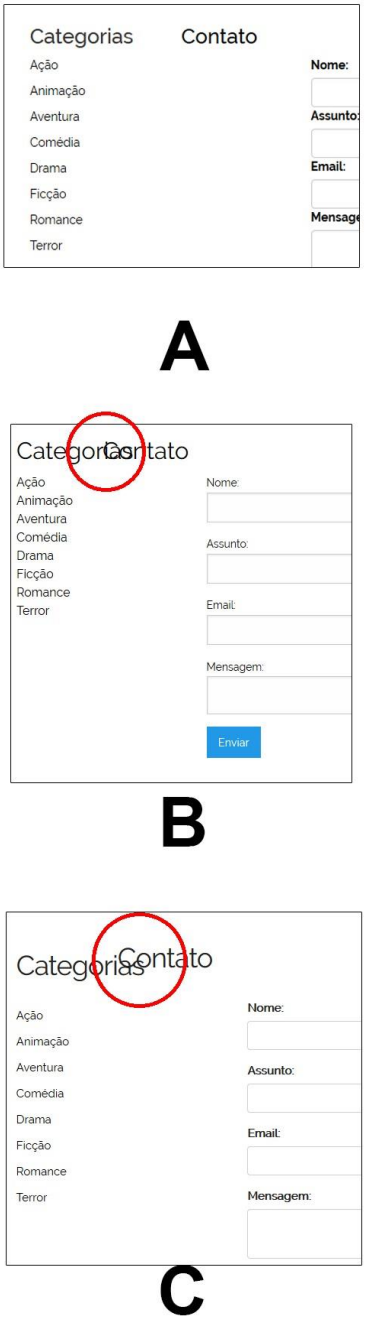


Fonte: DO AUTOR (2016)

Percebe-se que todos os frameworks empregaram bem o design responsivo, apresentando layouts semelhantes e em conformidade com os esboços projetados (Seção 3.1.1). Entretanto, em determinadas resoluções, podem ser encontrados alguns erros e comportamentos diferenciados, os quais são descritos a seguir.

Nos frameworks Foundation e Skeleton ocorreram erros entre as faixas de resolução entre 550 e 1200 pixels de largura, comportando uma grande variedade de dispositivos como smartphones e tablets. Esse erro ocorreu devido a tag <*h3*> que, marcando o título do menu local, ultrapassou o limite de suas colunas. Os erros, destacados em vermelho, são apresentados na Figura 39, através da página de contatos do website implementado.

Figura 39: Erros apresentados. Frameworks Bootstrap (A), Foundation (B) e Skeleton (c)

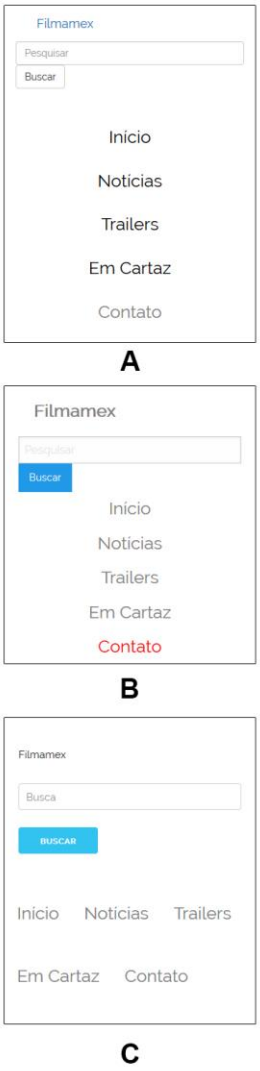


Fonte: DO AUTOR (2016)

Entretanto, esse problema pode ser solucionado ao modificar a implementação da navegação local, ao utilizar maior número de colunas através dos frameworks Foundation e Skeleton.

Além do erro de interface apresentado, o Skeleton não fornece menus responsivos, exigindo a implementação do recurso manualmente. Em dispositivos com resolução de 360 pixels de largura, verificou-se que o menu do Skeleton teve um comportamento irregular em relação aos implementados no Foundation e Bootstrap. A Figura 40, a seguir, ilustra os menus implementados através dos frameworks. Pode-se concluir que a compreensão do menu no Skeleton não organiza-se através de uma lista, dificultando a identificação das opções de navegação.

Figura 40: Menus responsivos dos frameworks Bootstrap (A), Foundation (B) e Skeleton (C)



Fonte: DO AUTOR (2016)

Para solucionar esse comportamento no menu no Skeleton, foi adicionada uma media querie. Essa permite a estilização da lista de links de modo que o menu fique similar ao do Bootstrap e Foundation. A media querie proposta como solução é apresentada na Figura 41, já o aspecto visual do menu obtido pode ser visualizado na Figura 42.

Figura 41: Adequando o menu do skeleton através de uma media querie

```
@media screen and (max-width: 361px) {
  /*A classe que vai ser alterada e o seu elemento*/
   .menu a {
      display: block;
      /*Largura alterada, levando em consideração o espaçamento lateral*/
      width: 340px;
      /*Alinhando o texto no centro*/
      text-align: center;
   }
}
```

Fonte: DO AUTOR (2016)

Figura 42: Menu adaptado no framework Skeleton através do uso de uma media querie

Fonte: DO AUTOR (2016)

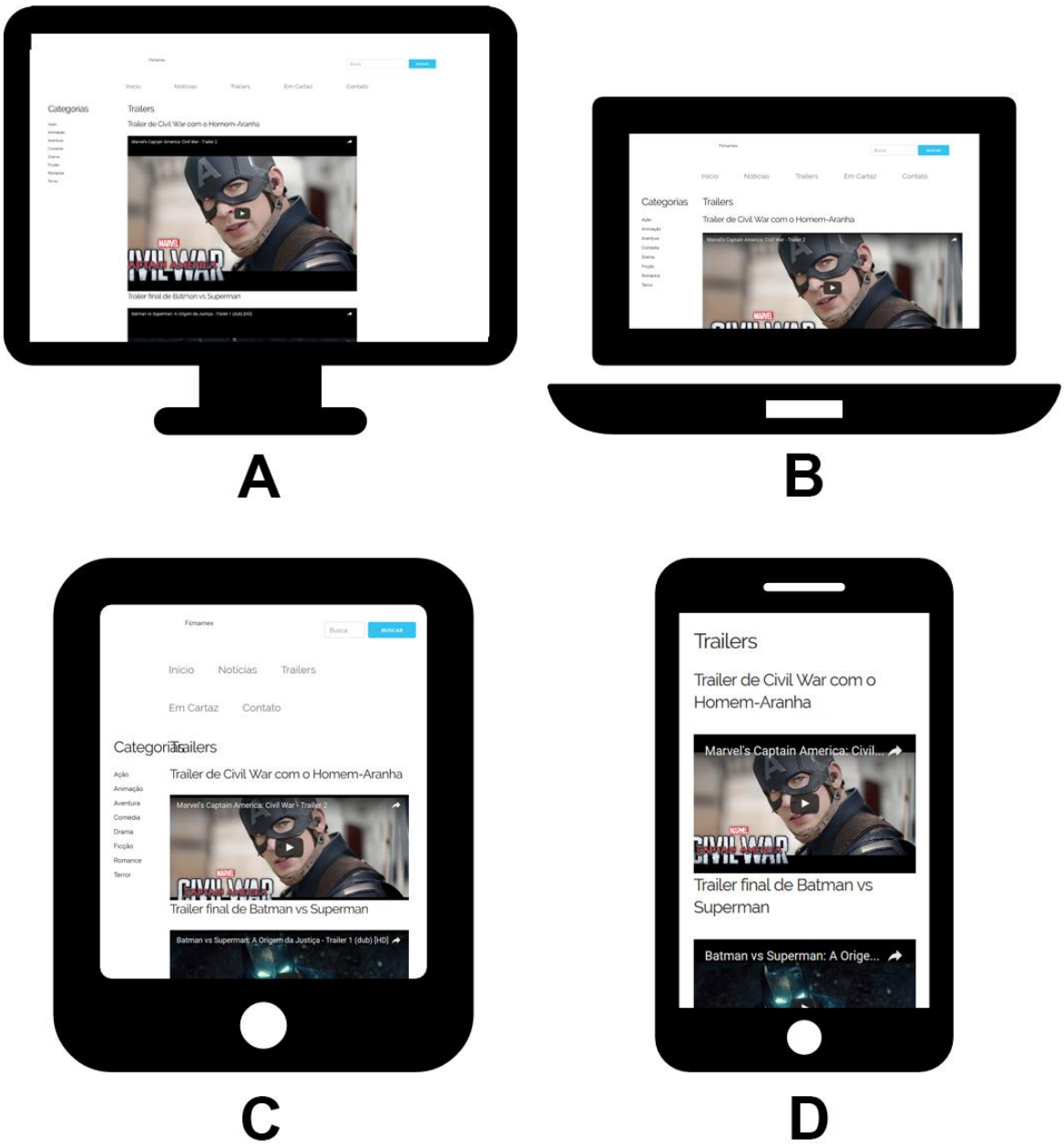
Outra dificuldade encontrada no framework Skeleton foi a necessidade de implementar manualmente os trailers do website. Para a implementação dos vídeos responsivos, foram necessárias *media queries* em CSS para adequar a altura de *iframes* em diferentes resoluções, conforme os códigos apresentados na Figura 43.

Figura 43: *Media Queries* utilizadas na implementação dos vídeos responsivos através do framework Skeleton

Fonte: DO AUTOR (2016)

Assim, a partir das medias queries implementadas no CSS para adequar a altura dos vídeos responsivos, a página de trailers pode ser visualizada através dos diferentes dispositivos especificados no começo dessa sessão, na Figura 44.

Figura 44: Vídeos responsivos implementados com o auxílio de media queries no Skeleton



Fonte: DO AUTOR (2016)

Outra característica apresentada pelos frameworks quando se fala de design responsivo é a tipografia. O Foundation e o Skeleton utilizam medidas relativas conhecidas como *REM*[3] para controlar o tamanho de suas fontes, que podem ser redimensionadas conforme diferentes resoluções. Já o Bootstrap, utiliza em suas

[3] "*REM*" é uma unidade de medida em CSS que especifica o tamanho de uma fonte em relação ao elemento raíz da estrutura HTML.

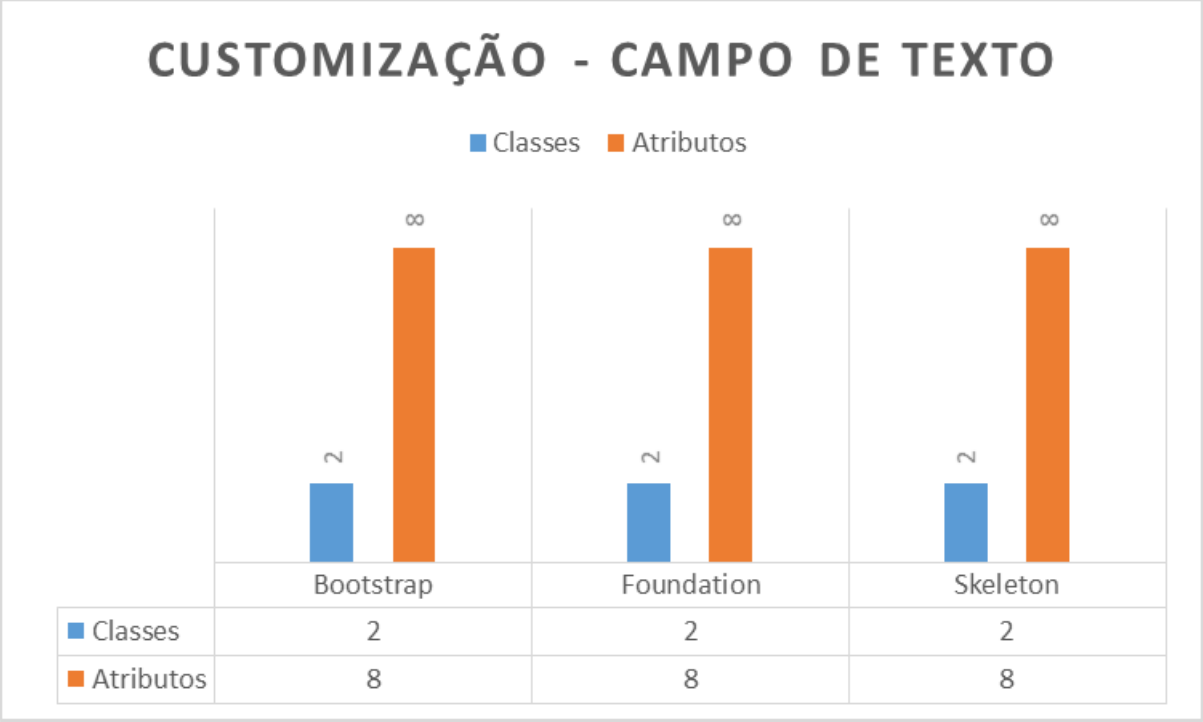
fontes a unidade de medida fixa *pixels,* não redimensionando o tamanho seus textos em dispositivos menores. O uso de diferentes unidades de medida na tipografia de um website pode trazer sutis diferenças na apresentação do conteúdo em diferentes resoluções.

4.5 CUSTOMIZAÇÃO

Nesta seção é avaliada a capacidade de customização dos elementos dos frameworks a partir do modelo de barra de busca proposto na Seção 3.3.5, com base no número de classes e atributos modificados para obter o aspecto visual desejado.

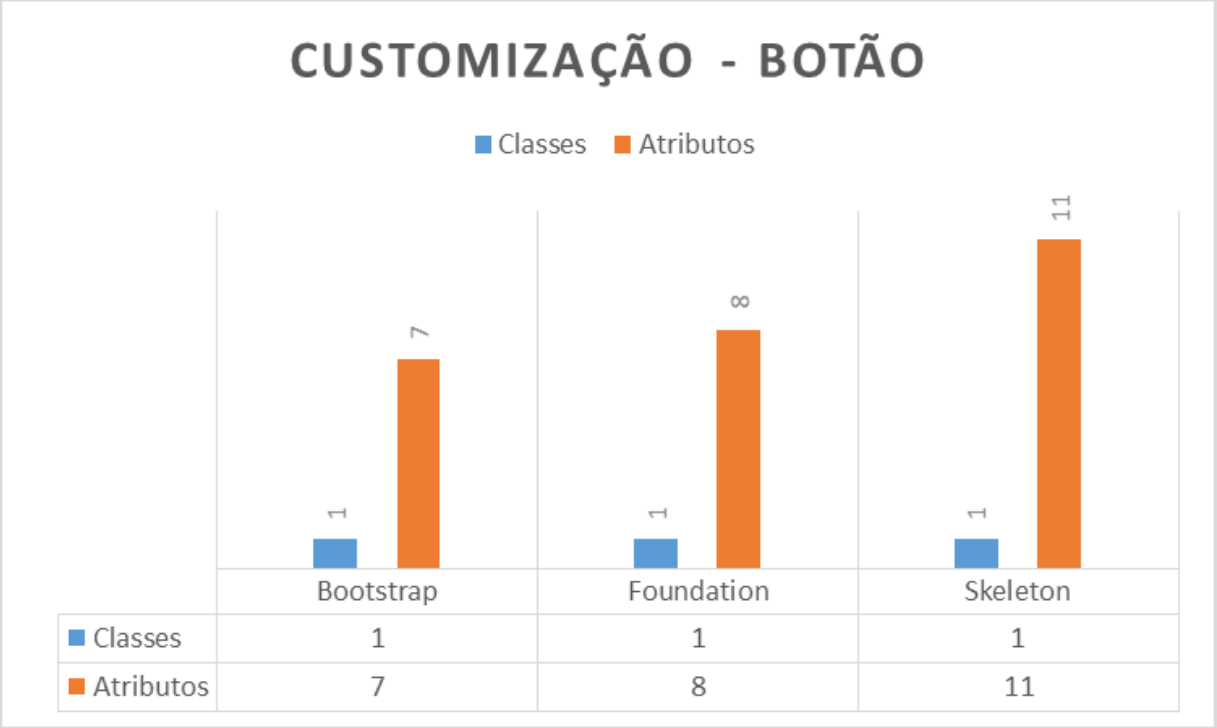
A Figura 45 mostra os resultados obtidos na customização do campo de texto e a Figura 46, apresenta os resultados na customização do botão, ambos os elementos que compõem a barra de busca proposta.

Figura 45: Classes e atributos alterados para a customização de um campo de texto através dos frameworks



Fonte: DO AUTOR (2016)

Figura 46: Classes e atributos alterados para a customização de um botão através dos frameworks



Fonte: DO AUTOR (2016)

O número de classes customizadas foram iguais em todos os frameworks. Em relação ao número de atributos, constatou-se que, na customização do campo de texto, foram equivalentes ao alterar os estilos da borda, do tamanho e das cores de fontes, das sombras e cores de fundo. Já na customização do botão, percebeu-se um maior número de atributos modificados no Skeleton pois, além de alterar as mesmas estilizações citadas anteriormente, também foram necessárias adaptações para alinhar o texto no centro do botão.

Por fim, conclui-se que a complexidade da customização de componentes de interface está relacionada à quantidade de recursos que podem ser alterados para obter a identidade visual desejada.

CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

Atualmente é imprescindível que as aplicações web sejam desenvolvidas empregando o conceito de design responsivo, permitindo fornecer aos usuários uma melhor visualização do conteúdo e facilidade de navegação, independentemente da resolução de tela empregada no seu dispositivo ao acessar o conteúdo web.

A implementação manual de layouts que contemplem a variedade de dispositivos existentes atualmente pode ser uma tarefa complexa, pois exige o projeto de websites específicos a cada resolução existente. Com o intuito de simplificar a implementação, reaproveitar códigos e reduzir o tempo desenvolvimento, surgem os frameworks de design responsivo.

Assim, o presente trabalho teve como objetivo o estudo comparativo entre os frameworks de design responsivo Bootstrap, Foundation e Skeleton. A partir de três versões implementadas de um website proposto como ambiente de testes, foi possível identificar aspectos relevantes em cada um dos frameworks. Vale ressaltar que todos os frameworks comparados nesse estudo cumpriram com o objetivo de implementar o conceito design responsivo no website desenvolvido.

O Bootstrap, mesmo possuindo uma sintaxe de classes através de prefixos, mostrou-se mais eficiente ao utilizar uma menor quantidade de classes e tags na construção de layouts. Diferentemente do Foundation e Skeleton, ele não necessita da adição da classe *columns* na implementação das colunas do grid. Além de fornecer os recursos necessários para a implementação do website proposto, foi o único framework a não apresentar erros de interface em determinada resolução.

O Foundation utiliza uma sintaxe de classes mais sugestiva na implementação de suas colunas, evitando o uso de prefixos. Forneceu todos os recursos necessários para a implementação do website proposto, porém, apresentou um erro de interface em uma determinada resolução. Ele demonstrou-se mais discreto ao não apresentar características relevantes em relação ao Bootstrap e Skeleton.

O Skeleton foi o framework com a sintaxe de classes mais claras e sugestivas, já que utiliza apenas o número do tamanho das colunas do grid. Ele apresentou um erro de interface em determinada resolução e exigiu a implementação manual de alguns recursos, como vídeos e menus responsivos. Seu principal destaque é uma proposta de diferenciada de design responsivo, já que faz

a adaptação do conteúdo automaticamente em dispositivos menores, sem exigir a implementação de novas colunas como o Bootstrap e Foundation.

Como planejamentos para trabalhos futuros, pode-se estender o estudo comparativo entre os frameworks através de avaliações de performance, desenvolver uma aplicação que contemple todos os recursos fornecidos pelos frameworks e até mesmo o desenvolvimento de um framework com os recursos mais vantajosos identificados nesse trabalho. Além disso, aperfeiçoamentos no website construído podem ser realizados, como a expansão para um portal com maior número de páginas com novas funcionalidades e também a conexão com uma base de dados.

REFERÊNCIAS

ALLSOPP, John. *A Dao of Web Design*. 7 de abr. 2000. Disponível em: <http://alistapart.com/article/dao/>. Acesso em: 01 set. de 2015.

BATURAY, Meltem; BIRTANE, Murat. *Responsive web design: a new type of design for web-based instructional content.* Artigo (4th International Conference on New Horizons in Education) – Turquia – 2013.

BOOTSTRAP. About Bootstrap. Disponível em <http://getbootstrap.com/about/#history/>. Acesso em: 02 set. 2015.

CASTRO, Elizabeth; HYSLOP, Bruce. *HTML5 e CSS3***.** Rio de Janeiro, RJ: Alta Books, 2013. 552 p. ISBN 9788576088035.

CHROME, Google. Disponível em: < https://www.google.com.br/chrome/browser/>. Acesso em: 01/07/2016

FIREFOX, Mozilla. Disponível em: <https://www.mozilla.org/pt-BR/firefox/>. Acesso em: 01/07/2016

FLANAGAN, David. *JavaScript: o guia definitivo*. 4. ed. Porto Alegre, RS: Bookman, 2004. 818 p.

FLANAGAN, David. *JavaScript: o guia definitivo*. 6. ed. Porto Alegre, RS: Bookman, 2013. 1062 p.

FREEMAN, Elisabeth; FREEMAN, Eric. *Use a cabeça!: HTML com CCS e XHTML*. 2. ed. Rio de Janeiro, RJ: Alta Books, 2008. 580 p. (Use a cabeça).

FOUNDATION. History of Foundation. Disponível em <http://foundation.zurb.com/learn/about.html/>. Acesso em: 26 jun. 2015.

GARTNER. Gartner Says Worldwide PC, Tablet and Mobile Phone Combined Shipments to Reach 2.4 Billion Units in 2013. 2013. Disponível em: <http://www.gartner.com/newsroom/id/2408515>. Acesso em: 05 de ago. 2015.

HAROLD, Elliotte Rusty. *Refatorando HTML: como melhorar o projeto de aplicações WEB existentes*. Porto Alegre: Bookman, 2010. 360 p. ISBN 9788577806317.

KATAJISTO, Laura. *Creating Support Content for Responsive Websites at Microsoft Mobile*. 2015. 4 páginas. Artigo (Microsoft Mobile) – IEEE – 2015.

KEMP, Symon. We Are Social - Digital, Social & Mobile In 2015 - We Are Social's Compendium Of Global Digital Statistics. 21 de jan. 2015. Disponível em: <http://wearesocial.net/blog/2015/01/digital-social-mobile-worldwide-2015/>. Acesso em: 13 mai. 2015.

KRUG, Steve. *Não me faça pensar: Uma Abordagem de Bom Senso à Usabilidade na Web*. 2ª Edição Traduzida. Rio de Janeiro: Alta Books, 2006. 201 páginas.

LOPES, Sérgio. *A Web Mobile: Design Responsivo para uma Web adaptada ao mundo mobile.* 2ª Edição Ampliada. São Paulo: Casa do Código, 2015. 273 páginas.

MARCOTTE, Ethan. *Responsive Web Design*. 15 de mai. 2010. Disponível em: <http://alistapart.com/article/responsive-web-design/>. Acesso em: 22 set. 2015.

MAZZA, Lucas. *HTML5 e CSS3 – Domine a web do futuro.* 1ª Edição. São Paulo: Casa do Código, 2012. 217 páginas

MEMÓRIA, Felipe. *Design Para a Internet: Projetando a experiência perfeita.* 7ª Edição. Rio de Janeiro: Elsevier. 2005.

MOHOROVICIC, S. *Implementing Responsive Web Design for Enhanced Web Presence*. Mai. 2013. 5 páginas. Artigo (MIPRO 2013) – Universidade de Rijeka – Rijeka – Croácia – 2013.

MOON*, JaeWon. Advanced Responsive Web Framework based on MPEG-21*. 3 páginas. Artigo (2012 IEEE Second International Conference on Consumer Electronics) – Berlin – 2012.

MULLER, Nícolas. *Framework, o que é e para que serve?*. 2008. Disponível em: < http://www.oficinadanet.com.br/artigo/1294/framework_o_que_e_e_para_que_serve>. Acesso em: 9 out. 2015.

NIELSEN, Jakob; LORANGER, Hoa. *Usabilidade na web: projetando websites com qualidade*. Rio de Janeiro, RJ: Elsevier, 2007. 406 p.

PACHECO, Andrea. *Um guia completo sobre grids para design responsivo*. Out. 2014. Disponível em: <http://arquiteturadeinformacao.com/design-de-interacao/guia-completo-sobre-grids-para-design-responsivo/> Acesso em: 14 nov. 2015.

PHP METRICS. Disponível em <http://http://www.phpmetrics.org//>. Acesso em: 30 jun. 2016.

SILVA, Arthur. *Design Responsivo: Técnicas, Frameworks e Ferramentas.* Dez. 2015. 75 páginas. Monografia (Bacharelado em Sistemas de Informação) – Universidade Federal do Estado do Rio de Janeiro – Rio de Janeiro – 2014.

SILVA, Maurício[1]. *Criando sites com HTML: Sites de alta qualidade com HTML e CSS.* 1ª Edição. São Paulo - SP: Novatec Editora, 2008.

SILVA, Maurício[2]. *Construindo sites com CSS e (X)HTML: sites controlados por folhas de estilo em cascata.* São Paulo - SP: Novatec Editora, 2008.

SILVA, Maurício[3]. Jquery**:** a biblioteca do programador JavaScript. 2. ed. São Paulo, SP: Novatec, 2010. 543 p

SKELETON. Disponível em <http://getskeleton.com/>. Acesso em: 02 set. 2015.

SOMMER, Andreas; KRUSCHE Stephan. *Evaluation of cross-platform frameworks for mobile applications.* 2013. 16 páginas. Artigo.

SOMMERVILLE, Ian. *Engenharia de Software*. 9. ed. São Paulo, SP: Pearson, 2011. 529 p. ISBN 9788579361081.

W3C. *HTML5 Introduction*. Disponível em: <www.w3schools.com/html/html5_intro.asp> Acesso em: 01 jul. 2016.

W3CSCHOOLS. *CSS Introduction*. Disponível em: <http://www.w3schools.com/css/css_intro.asp> Acesso em: 14 nov. 2015.

W3CSCHOOLS. Responsive Web Design – Frameworks. Disponível em: <http://www.w3schools.com/css/css_rwd_frameworks.asp>. Acesso em: 28 ago. 2015.

W3CSCHOOLS. CSS – Introduction. Disponível em: <http://www.w3schools.com/css/css_intro.asp> Acesso em: 14 nov. 2015.