

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE  
SÃO PAULO – CAMPUS SÃO JOÃO DA BOA VISTA**

**LUCAS EDUARDO SPRENG**

**ESTUDO E APLICAÇÃO DE TÉCNICAS E METODOLOGIAS DE  
DESIGN RESPONSIVO NO RECLAME SÃO JOÃO.**

**SÃO JOÃO DA BOA VISTA – SP**

**2017**

**LUCAS EDUARDO SPRENG**

**ESTUDO E APLICAÇÃO DE TÉCNICAS E METODOLOGIAS DE  
DESIGN RESPONSIVO NO RECLAME SÃO JOÃO.**

Trabalho de Conclusão de Curso  
apresentado ao Curso Técnico Integrado  
em Informática, do Instituto Federal de  
Educação, Ciência e Tecnologia de São  
Paulo – Campus São João da Boa Vista.

Orientador: Prof. Breno Lisi Romano

SÃO JOÃO DA BOA VISTA – SP

2017

## Resumo

A partir do momento em que a tecnologia passou a evoluir com maior velocidade, tendo grande enfoque na praticidade aos usuários, inicia-se o surgimento de dispositivos móveis, sendo eles *tablets* e celulares, também surgiu uma demanda por sistemas compatíveis.

O enfoque desta pesquisa baseia-se em tornar uma das seções do portal Reclame São João compatível com todos os dispositivos que sejam capazes de acessar por meio da *Internet*. Para sua realização, o uso de uma análise das técnicas juntamente com sua posterior aplicação é indispensável para o cumprimento deste objetivo.

Dando sequência no trabalho, para que se torne mais fácil a criação de *designs* responsivos, existem *frameworks* capazes de auxiliar neste quesito, como o caso do *Bootstrap*, utilizado no desenvolvimento da pesquisa.

Com o presente trabalho, foi possível tornar a área de gráficos e relatórios do portal em questão totalmente responsivo, fazendo com que seja possível realizar seu acesso de qualquer dispositivo. Ainda sim deixando uma abertura caso em outra oportunidade seja possível realizar a aplicação de um *framework* mais completo que possa padronizar totalmente não só esta seção do portal, mas sim o projeto como um todo.

Palavras chaves: *framework*, *bootstrap*, *design*, responsivo, dispositivos, *site*.

## Lista de Figuras

Figura 1. Exemplo de utilização do <i>Media Types</i> no próprio arquivo. ....	9
Figura 2. Exemplo de utilização do <i>Media Types</i> na referência da folha de estilo. ....	9
Figura 3. Exemplo de utilização do <i>Media Queries</i> no CSS, com barra dinâmica. ....	10
Figura 4. Visualização da aplicação de <i>Media Queries</i> . ....	11
Figura 5. Visualização do <i>Grid System</i> do <i>Bootstrap</i> . ....	12
Figura 6. Estrutura de grade, HTML e CSS. ....	13
Figura 7. Visualização do modelo base do sistema de grades do <i>Bootstrap</i> . ....	13
Figura 8. Distribuição da estrutura do <i>Bootstrap</i> entre <i>container</i> , <i>rows</i> e <i>cols</i> . ....	14
Figura 9. Parâmetro de configuração para o <i>Datatables</i> habilitado caráter responsivo. ....	15
Figura 10. Reclame São João sem divisão responsiva para os conteúdos. ....	15
Figura 11. Adição de distância na lateral esquerda e centralização. ....	16
Figura 12. Centralização do conteúdo e espaçamento na lateral esquerda. ....	16
Figura 13. Visualização no dispositivo móvel ainda com sobreposição da navegação. ....	17
Figura 14. Folha de estilo correspondente a div do conteúdo. ....	17
Figura 15. Gráfico com distância desigual entre as laterais da página. ....	18
Figura 16. Uso da classe com sobreposição de estilo. ....	18
Figura 17. Tabelas em seu estado final na área de <i>rankings</i> . ....	19
Figura 18. Página com gráfico após modificações. ....	19

## Sumário

1	Introdução .....	6
2	Desenvolvimento .....	8
	2.1 Media Types e Media Queries.....	8
	2.2 Bootstrap Framework.....	11
	2.2.1 Aplicação de design responsivo no portal Reclame São João.....	14
3	Conclusão.....	20
4	Referências Bibliográficas.....	20

# 1 Introdução

O intuito deste capítulo gira em torno de gerar uma breve discussão sobre a necessidade da existência de um sistema capaz de se adaptar e possuir maior acessibilidade a todos usuários, visando suprir dada necessidade aos moradores e consumidores do município de São João da Boa Vista, podendo ser acessado pelos diferentes e atuais dispositivos, também respondendo de acordo com cada método de entrada, tais quais teclados, *mouses*, *touch screen*, teclados virtuais e diferentes áreas de visibilidade.

Além de problematizar esta questão, a recente técnica desenvolvida para solucioná-la também estará presente no capítulo, sendo detalhada logo a seguir, no capítulo de desenvolvimento.

## 1.1 Problematização

Partindo do momento em que o uso de sistemas *web* deixaram de ser acessados essencialmente por computadores com um navegador instalado, guiados por *mouses* e teclados, estes sistemas passaram a demandar uma nova perspectiva, incluindo novos desafios aos desenvolvedores (ZEMEL, 2015). Eis a necessidade da criação de sistemas capazes de interagir integralmente com todos os dispositivos, inclusive com seus novos métodos de entrada, como o *touch screen* nos *tablets*, *smartphones* e poucos modelos de *notebooks*.

Em frente à questão de um sistema *web* capaz de se adaptar em qualquer dispositivo, a qual já foi discutido em inúmeras pesquisas e por diversos desenvolvedores por todo o mundo, passou a surgir um conjunto de técnicas para realizar essa adaptação, conhecidas como *Design Responsivo* (SILVA, 2014).

## 1.2 Objetivos

Um problema recorrente entre os estabelecimentos e seus respectivos consumidores era o do surgimento de muitas reclamações não solucionadas, uma vez que os consumidores tinham imensa O presente trabalho tem como principal objetivo, o desenvolvimento de um único site capaz de ser acessado por todos e quaisquer dispositivos com acesso à internet.

Para que o foco do trabalho seja alcançado o desenvolvimento se dará respectivamente por três principais fases, divididas entre análise dos métodos e técnicas, aplicação dos mesmo no site em desenvolvimento e finalmente a realização de testes para verificar se a tarefa

foi cumprida com êxito. A análise dos métodos e técnicas buscará por técnicas e métodos de uso das mesmas para que os navegadores em diferentes dispositivos sejam capazes de exibir a melhor interface possível e oferecer boa experiência de uso.

Partindo da análise e discussão das técnicas, sua aplicação se dará pelo meio mais viável encontrado e através dos métodos estudados, buscando atender o princípio deste trabalho. Seguindo o desenvolvimento a verificação estará como final etapa do desenvolvimento, demonstrando os resultados obtidos através da aplicação da técnica e eficácia da mesma.

### 1.3 Metodologia

Para que haja o desenvolvimento deste trabalho será indispensável o uso dos itens a seguir:

- Computadores capazes de suportar o desenvolvimento o teste de uma aplicação *web*.
- Ferramentas instaladas no computador para o auxílio do desenvolvimento, tais quais o IDE *Sublime Text 3*, servidores PHP para a utilização de *scripts* ao lado do servidor gerando a página a ser visualizada pelo usuário e o conjunto de serviços e utilidades do *MySQL*, visando proporcionar a persistência dos dados no sistema.
- Documentação de todos os elementos utilizados no desenvolvimento do sistema, tais quais HTML, PHP, *Javascript*, CSS, *JQuery* e também das bibliotecas, como, por exemplo, *Highcharts*, *Highstocks* e *Autocomplete UI*.
- Para a estruturação do projeto foi determinado o uso do modelo MVC (*Model, View and Control*), buscando uma maior organização e capacidade de manutenção para o projeto.
- Dispositivos para a realização dos testes e verificações sobre o sistema.

### 1.4 Organização do texto

Seguindo após a introdução, o presente trabalho será articulado da seguinte maneira:

Capítulo 2: Será abordado o principal conteúdo, possuindo as três fases de desenvolvimento recentemente citadas. Além de serem descritos as técnicas e conceitos existentes, juntamente com o aprofundamento dos que forem utilizados, haverá o desenvolvimento detalhado seguido pelos testes de funcionamento. Este capítulo estará integralmente ligado ao desenvolvimento e resultados do trabalho como um todo.

Capítulo 3: Finalizando o conteúdo deste trabalho, será responsável por conter as considerações finais em relação aos resultados obtidos, demonstrando a eficácia da aplicação do sistema de *Design* Responsivo dentro do projeto Reclame São João.

## 2 Desenvolvimento

Partindo do princípio de que o *design* responsivo faz com que o site seja capaz de ser acessado de qualquer dispositivo, adaptando a disposição tanto da interface e do conteúdo na tela do dispositivo, visando sempre a melhor navegabilidade, estarão presentes nos próximos tópicos a chave para a responsividade como, também, o mais popular *framework*, o *Bootstrap*.

Posteriormente o desenvolvimento deste trabalho será voltado para a demonstração do *Bootstrap* na seção de rankings juntamente com o uso de *Media Queries* no portal Reclame São João.

### 2.1 *Media Types e Media Queries*

Tanto as *Media Queries* quanto as *Media Types*, foram propostas pela *World Wide Web Consortium* (W3C) organização responsável pela padronização dos conteúdos criados e interpretados nos navegadores da *Web* como solução para a compatibilidade dos sites em todos os dispositivos possíveis.

Em seu modelo inicial, nas *Media Types*, serviriam para adaptar o *site* em cada dispositivo, no caso, com os seguintes parâmetros, fazendo com que os códigos CSS sejam aplicados de acordo com determinado fator:

- *all*: Faz com que o código seja completamente aplicado ao site, independentemente do dispositivo.
- *braille*: Aplicado apenas a dispositivos táteis.
- *embossed*: Aplicado apenas a impressoras de *Braille*.
- *handheld*: Aplica apenas para dispositivos de mão, como os *handhelds*, que eram pequenos *notebooks*, e *palmtops*.
- *print*: Aplica apenas para impressões.
- *projection*: Utilizado para apresentações como as de *PowerPoint*, de *slides*.
- *screen*: Aplica em dispositivos com *display* colorido.
- *speech*: Aplica a sintetizadores de voz.
- *tty*: Aplica a *displays* simples com quantidade de caracteres limitados, como terminais e dispositivos móveis antigos.
- *tv*: Aplica em televisores com baixa resolução e alta quantidade de cores.



A aplicação deste parâmetro pode ocorrer de duas maneiras: podem ser inseridos no próprio arquivo ou quando se cria a referência à folha de estilo, como demonstrado na Figura 1 e na Figura 2.

Figura 1. Exemplo de utilização do Media Types no próprio arquivo.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Aplicação Media Types</title>
</head>
<body>
  <style type="text/css">

    @media screen {
      body{
        font-size: 12px;
      }
    }
    @media print{
      body{
        font-size: 15px;
      }
    }

  </style>
</body>
</html>
```

Figura 2. Exemplo de utilização do Media Types na referência da folha de estilo.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Aplicação Media Types</title>

  <link rel="stylesheet" type="text/css" href="estiloScreen.css" media="screen">
  <link rel="stylesheet" type="text/css" href="estloPrint.css" media="print">

</head>
<body>

</body>
</html>
```

Porém as *Media Types* foram desenvolvidas antes de existirem os *smartphones* e toda a linha de dispositivos móveis, fazendo com que não se adaptem totalmente. Por exemplo, não seria possível simplesmente utilizar o parâmetro *handheld* sendo que funcionam de maneira totalmente diferente aos *smartphones*, além de possuírem distintos tamanhos. Utilizar apenas o *screen* também não, já que computadores e dispositivos móveis possuem diferentes usabilidades.

Com isso, a W3C propôs a evolução da qual estamos falando, as *Media Types*, fazendo com que seja possível utilizar não só o tipo de dispositivo como parâmetro, mas também características específicas, tais quais as dimensões do *display*, suas proporções, orientação, resolução e muitas outras.

A evolução proposta no *Media Queries* funciona fazendo uso adicional das características recém citadas, ao declarar o tipo de mídia, como podemos ver na Figura 3.

Figura 3. Exemplo de utilização do *Media Queries* no CSS, com barra dinâmica.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Aplicação Media Types</title>
</head>
<body>
  <div class="barra"></div>
  <style type="text/css">

    body{
      padding:0;
      margin: 0;
    }

    .barra{
      position: fixed;
      background-color: #000000;
    }

    @media screen and (min-width: 1000px){
      .barra{
        width: 100%;
        height: 20%;
      }
    }

    @media screen and (max-width: 1000px){
      .barra{
        width: 20%;
        height: 100%;
      }
    }

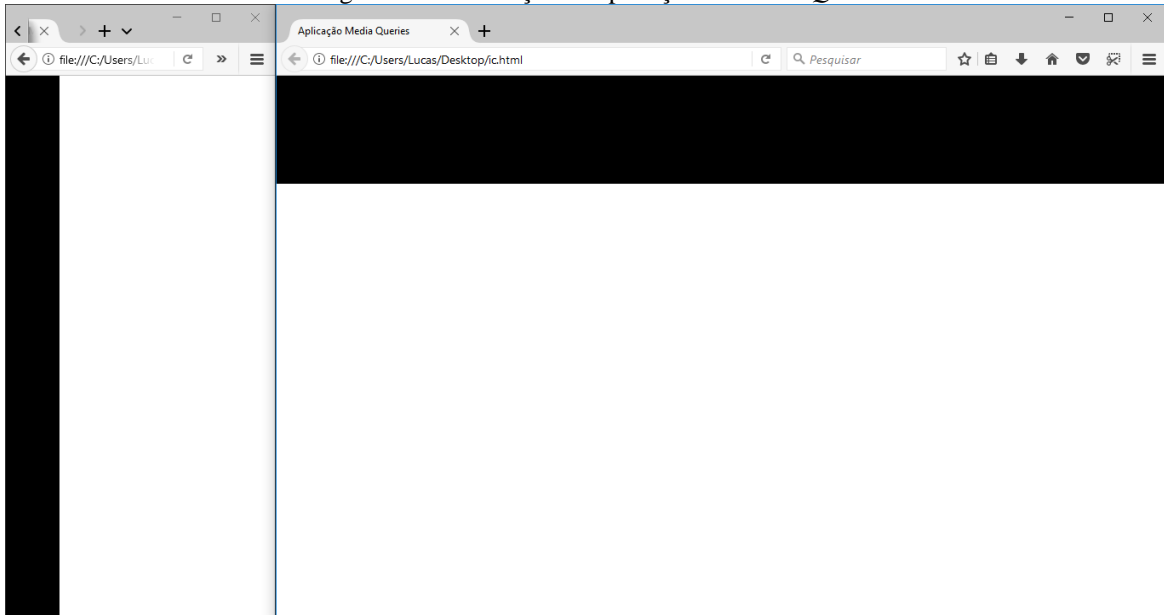
  </style>
</body>
</html>
```

Na Figura 3 foram utilizadas as características relativas às dimensões da tela, fazendo com que o conteúdo da página seja adaptado de acordo com a largura. Neste exemplo foi criada uma barra, exibida na Figura 4, a qual modifica seu posicionamento e tamanho sendo que o navegador possua mais ou menos de 1000 *pixels*.

O resultado deste exemplo é visível na Figura 4, representando um navegador executando a página em duas instâncias, sendo uma acima e outra abaixo de 1000 *pixels*. Com isso, o uso básico do *Media Queries* deriva destas características demonstradas, as quais fazem com que nossa barra modifique dinamicamente sua altura e largura.

Como também pode-se ver neste último exemplo, para que a barra se adapte ao tamanho da tela, é necessário utilizar a propriedade *width* e *height* do CSS, as quais são respectivamente referentes à altura e largura.

Figura 4. Visualização da aplicação de *Media Queries*.



Ambos estes parâmetros são essenciais para que o conteúdo possa se adaptar ao *display* conforme seu tamanho muda.

## 2.2 *Bootstrap Framework*

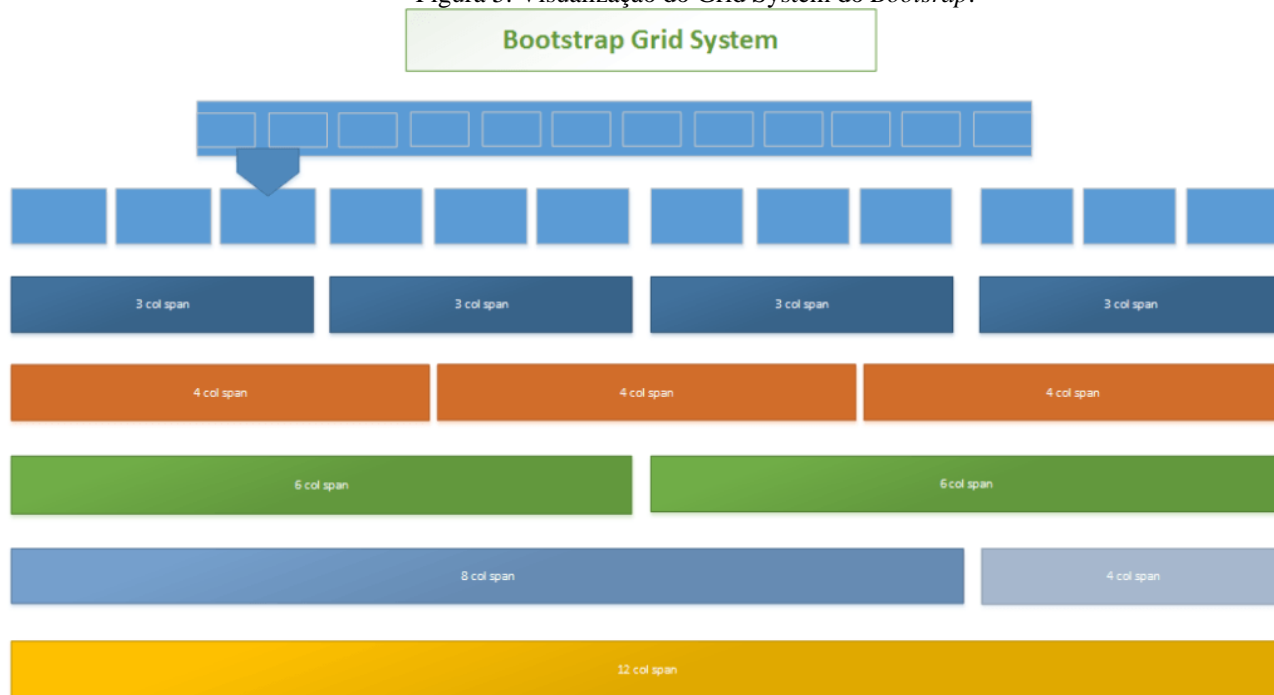
O *Bootstrap* foi criado inicialmente visando solucionar questões internas de organização do *Twitter*, sendo que a equipe de desenvolvimento não seguia padrão algum, criando códigos distintos, dificultando principalmente na integração das tarefas. Posteriormente houve um *release* público da ferramenta, sendo aprimorada com o passar do tempo.

Contudo, além de o *Bootstrap* oferecer um modelo padronizado para o desenvolvimento, também faz com que se torne mais prática a criação de designs responsivos, sendo uma solução ideal para a grande parte dos sites.

Dentro de suas classes, este *framework* já possui a padronização das dimensões utilizando valores percentuais referentes às dimensões da tela em que o *site* é exibido. Isso faz com que se adapte com maior facilidade e eficiência em todos os diferentes tamanhos.

Com isso, um dos grandes motivos do sucesso deste *framework* é sua organização dos elementos formada através de uma grade virtual no *display*, o *grid system*. Este sistema se encontra ilustrado na Figura 5.

Figura 5. Visualização do Grid System do *Bootstrap*.



Fonte: <http://www.webdevelopmenthelp.net/2016/10/bootstrap-interview-questions.html> (2017)

O sistema possui uma divisão igualitária da tela em doze colunas para dispositivos com grandes displays, sendo que cada uma das doze colunas preenche 8,33% do espaço disponível do navegador horizontalmente. Já no caso de dispositivos reduzidos, como no caso de celulares e inclusive *tablets* menores, todas as colunas passam a possuir toda a largura do *display*.

Seguindo este modelo, a divisão básica do *Bootstrap* possui sua base no modelo de CSS representado pela Figura 6.

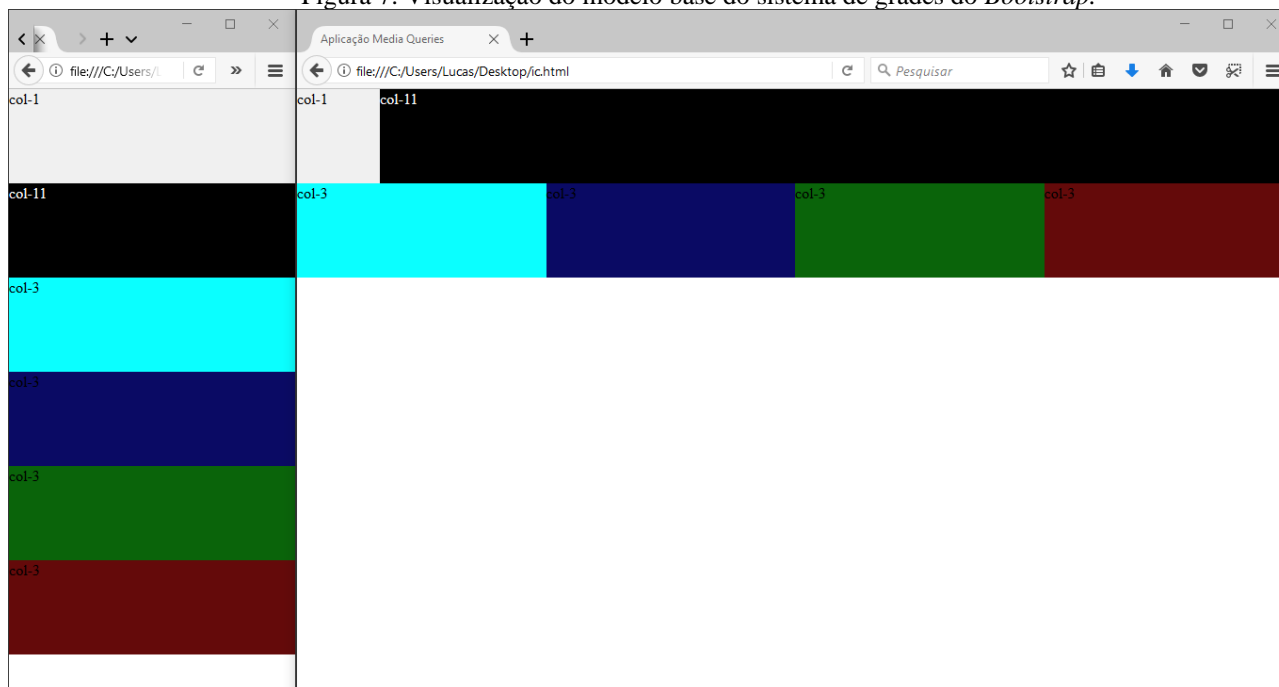
Figura 6. Estrutura de grade, HTML e CSS.

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="UTF-8">
5   <title>Aplicação Media Queries</title>
6   <link rel="stylesheet" type="text/css" href="
7     estilo.css">
8 </head>
9 <body>
10  <div class="col-1" style="background-color: #f0f0f0
11    ">col-1</div>
12  <div class="col-11" style="background-color: #
13    000000; color: white">col-11</div>
14  <div class="col-3" style="background-color: rgb(10,
15    255,255)">col-3</div>
16  <div class="col-3" style="background-color: rgb(10,
17    10,100)">col-3</div>
18  <div class="col-3" style="background-color: rgb(10,
19    100,10)">col-3</div>
20  <div class="col-3" style="background-color: rgb(100
21    ,10,10)">col-3</div>
22 </body>
23 </html>
24
25 body{
26   margin: 0;
27   padding: 0;
28 }
29
30 .col-1 {width: 8.33%;}
31 .col-2 {width: 16.66%;}
32 .col-3 {width: 25%;}
33 .col-4 {width: 33.33%;}
34 .col-5 {width: 41.66%;}
35 .col-6 {width: 50%;}
36 .col-7 {width: 58.33%;}
37 .col-8 {width: 66.66%;}
38 .col-9 {width: 75%;}
39 .col-10 {width: 83.33%;}
40 .col-11 {width: 91.66%;}
41 .col-12 {width: 100%;}
42
43 [class*="col-"] {
44   float: left;
45   height: 100px;
46 }
47
48 @media screen and (max-width: 320px){
49   [class*="col-"] {
50     width: 100%;
51   }
52 }
  
```

Fazendo uso desta folha de estilo simulando o sistema de grades, é possível visualizar a maneira como os elementos respondem de acordo com as dimensões da tela, como encontra-se na Figura 7. Foram adicionados apenas algumas colorações extras na estilização das *divs*, tornando-as mais visíveis.

Figura 7. Visualização do modelo base do sistema de grades do *Bootstrap*.



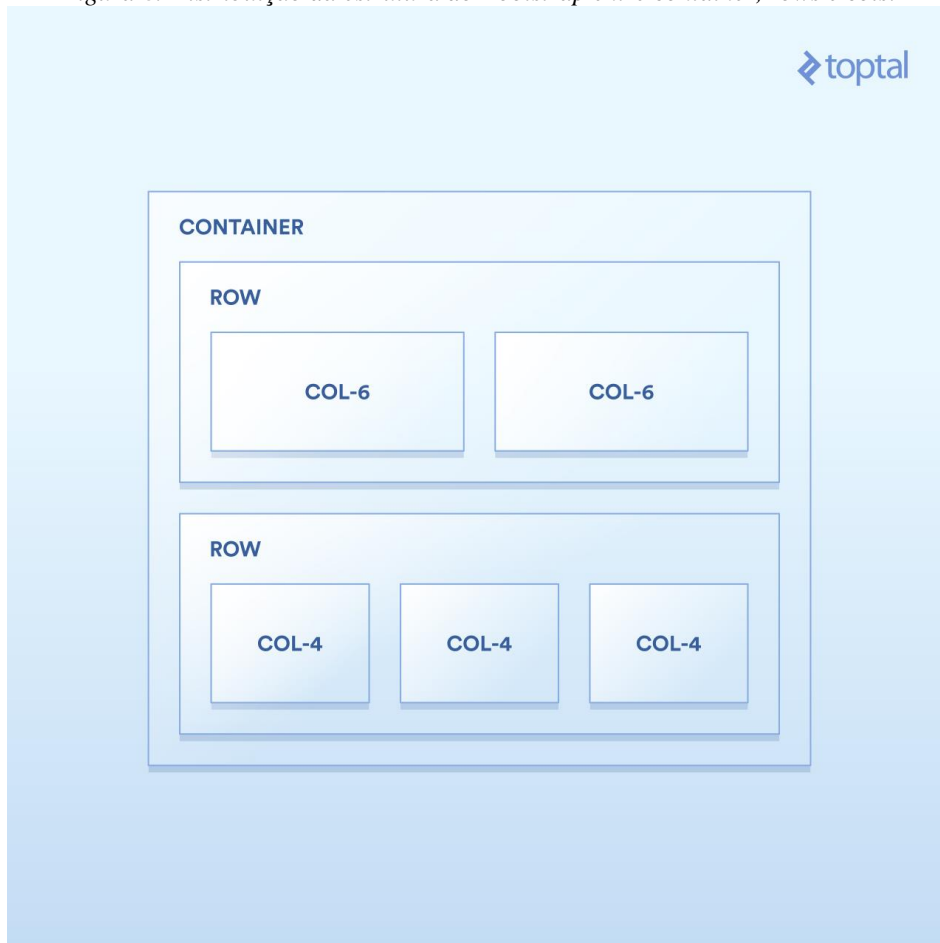
Como pode-se ver, esta modificação do conteúdo entre toda a largura e algumas frações faz com que a disposição dos elementos se torne mais próxima ao que podemos ver em sites atualmente.

É importante ressaltar que no sistema de grades do *framework*, ao utilizar colunas, é necessário inseri-las dentro de linhas, e, também, consequentemente todas as linhas em um

*container*. Na Figura 8 é possível visualizar a estrutura da interface utilizando as divisões corretamente, para que o conteúdo possa ser adaptado e distribuído.

A declaração destas divisões na estrutura HTML, assim como no caso das colunas, funciona com a adição de sua respectiva classe para a função, sendo atribuída a classe *row* para uma nova linha, e a classe *container* para o *container*.

Figura 8. Distribuição da estrutura do Bootstrap entre *container*, *rows* e *cols*.



Fonte: <https://www.toptal.com/sass/css3-flexbox-sass-grid-tutorial> (2016)

### 2.2.1 Aplicação de *design* responsivo no portal Reclame São João

Por conta de o portal possuir seu desenvolvimento com base no *template Oxygen*, o qual já possui *design* responsivo em sua página principal, além de suas folhas de estilo customizadas, o *bootstrap* já faz parte do projeto. Com isso, toda a parte mantida do *template* na seção de *rankings* já possui integração responsiva com os dispositivos, restando assim, apenas a necessidade de adaptar o conteúdo da página.

Para que os rankings e relatórios tabulares sejam feitos, o uso de algumas bibliotecas baseadas em *jQuery* é imprescindível. Nesta sessão do portal os requisitos do sistema, juntamente com a análise, apontam para que sejam criados alguns gráficos que possam exibir o desempenho, realizar comparações e alteração do posicionamento dos estabelecimentos; e também, tabelas exibindo todos os estabelecimentos cadastrados, filtrando-os entre si, juntamente com feedbacks deles.

Na busca pelas bibliotecas, um dos enfoques estava em encontrar algumas que suprissem a necessidade do portal em relação a conteúdo, também havendo características responsivas, fazendo com que sejam compatíveis com a proposta do *site*.

Visando a criação de todos os gráficos possíveis, o *Highcharts* juntamente com sua extensão para dados com periodicidade de tempo entre eles, o *Highstocks*, possibilitam o a criação de gráficos totalmente responsivos e com características de usabilidade aprimoradas para cada dispositivo.

Quanto à criação de tabelas, a biblioteca *Datatables* cumpre todos os quesitos, inclusive criando a possibilidade de realizar pesquisas entre os dados das tabelas. Dentre seus parâmetros de configuração existe um determinante para a transformação dinâmica nos dispositivos, como visto na Figura 9.

Figura 9. Parâmetro de configuração para o *Datatables* habilitado caráter responsivo.

```
1 $( '#myTable' ).DataTable( {  
2     responsive: true  
3 } );
```

JavaScript

Voltando então para a questão de ajustar o conteúdo na página de *rankings* do portal, o que precisa ser feito é ajustar o conteúdo para que se encaixe na tela de acordo com o seu tamanho. Podemos ver na Figura 10 como esta página se encontra sem que seja realizada a adequação do conteúdo.

Figura 10. Reclame São João sem divisão responsiva para os conteúdos.



Ranking completo dos estabelecimentos

10 resultados

Pesquisar:

ID	Estabelecimentos	Categoria	Pontuação
1	CASA DOS PARAFUSOS	salão de beleza	5
2	STEALTH EVENTOS E SERVICOS S/C LTDA	loja	0
3	HATAKEYAMA & SOARES LTDA ME	escola	0
4	INNERMAN CONFECCOES E COMERCIO LTDA	grafica	0
5	COVOLAN BENEFICIAMENTOS TEXTIS LTDA	academia	0
6	QUADRA ESPORTIVA SPORTBOLL LTDA MEL	clube esportivo	0
7	MULTI FLON INDUSTRIA E COMERCIO LTDA ME	barbearia	0
8	AVANCINI S CORRETORA DE SEGUROS LTDA	cabeleleiro	0
9	MADEMAT DISTRIBUIDORA LTDA ME	salão de beleza	0
10	MILIONI EMBALAGENS LTDA - E.P.P.	provedor de internet	0

Exibindo 1 / 10 entre 1 012 resultados

Como pode-se visualizar, o conteúdo em tela cheia de um computador está sendo sobreposto pela barra lateral, criando assim uma primeira necessidade de adicionar uma distância maior na lateral esquerda e posteriormente centralizar o conteúdo, da maneira como foi feito na Figura 11.

Figura 11. Adição de distância na lateral esquerda e centralização.



Fazendo o uso do *padding-left* já cria uma distância do lado esquerdo do *container* suficiente para que não seja sobreposto pela barra lateral. As margens automáticas de ambos os lados complementam, fazendo com que todo o conteúdo da *div* que utilize esta classe seja centralizado, como pode-se visualizar na Figura 12.

Figura 12. Centralização do conteúdo e espaçamento na lateral esquerda.



Em passos seguintes, a correção da caixa de pesquisa para que fique alinhada ao fim da tabela, juntamente com o tratamento para dispositivos com telas menores se torna prioridade. Na Figura 13 encontra-se a visão em um dispositivo móvel nesta etapa do desenvolvimento.



Figura 13. Visualização no dispositivo móvel ainda com sobreposição da navegação.



Neste caso o conteúdo ainda possui uma determinada sobreposição de elementos, incluindo também a má organização dos elementos de busca. Para esta correção, reduzir o tamanho da *div* portadora da classe *graficoResponsivo* já fará com que o *Datatables* reconheça devidamente o espaço para exibir as informações.

Visando manter uma proporção semelhante entre as margens entre o conteúdo e as duas laterais, esta redução deve ocorrer apenas em dispositivos menores, fazendo o uso do *Media Queries*.

Por questões de melhor visualização, é esperado que haja a centralização dos formulários desta página em telas reduzidas, porém a definição dos mesmos nos extremos da tabela em *displays* um pouco maiores.

Na Figura 14 pode-se observar as mudanças realizadas na folha de estilo para que estas modificações sejam aplicadas devidamente.

Figura 14. Folha de estilo correspondente a *div* do conteúdo.

```
.graficoResponsivo{
  padding-left: 30px;
  margin-left: auto;
  margin-right: auto;
  width: 90%;
}

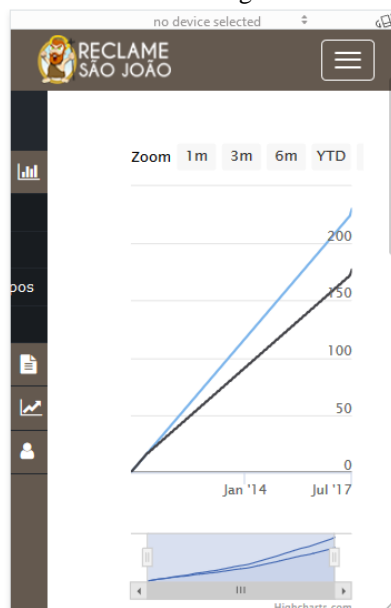
.dataTables_filter{
  float: none;
  margin: 0;
  padding: 0;
}

.container{
  text-align: center;
}

29
30 @media (min-width: 768px) {
31   .graficoResponsivo{
32     padding-left: 30px;
33     width: 100%;
34   }
35   .dataTables_filter{
36     float: right;
37   }
38   .container{
39     text-align: justify;
40   }
41 }
42
43
44
```

Com estas últimas alterações apenas a página incluindo os gráficos gerados pela biblioteca *Highcharts* dependem de sutis alterações, para que tenha melhor visibilidade ao usuário.

Figura 15. Gráfico com distância desigual entre as laterais da página.



Como já foi criada e adaptada a classe *graficoResponsivo* na folha de estilo, basta adicioná-la na *div* responsável pelo gráfico e retirar o espaçamento na lateral esquerda, da maneira como se encontra na Figura 16.

Figura 16. Uso da classe com sobreposição de estilo.

```
<div class="graficoResponsivo" id="ranking2" style="min-width: 260px
; max-width: 1000px; padding-left: 0"></div>

<!-- FIM PROBLEMAS SOLUCIONADOS-->

<!-- NÚMERO DE RECLAMAÇÕES NOS ÚLTIMOS TEMPOS -->

<div class="graficoResponsivo" id="ranking3" style="min-width: 260px
; max-width: 1000px; padding-left: 0"></div>

<!-- FIM RECLAMAÇÕES ULTIMOS TEMPOS -->
```

Apenas sobrepondo o *padding* que é determinado na classe *graficoResponsivo*, também definindo uma largura mínima e máxima para o gráfico, a qual é utilizada pela biblioteca para o seu funcionamento, já chegamos no resultado esperado, possuindo uma página responsiva e dinâmica, visualizado nas Figura 17 e 18.

Figura 17. Tabelas em seu estado final na área de *rankings*.

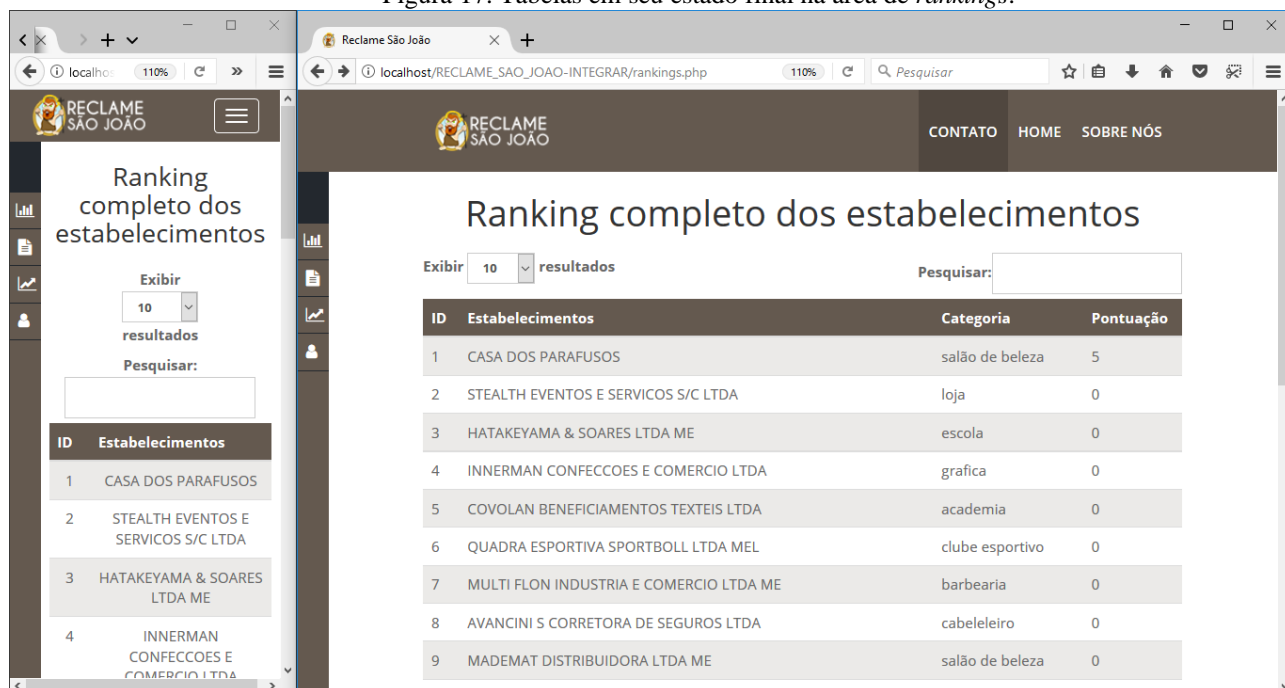
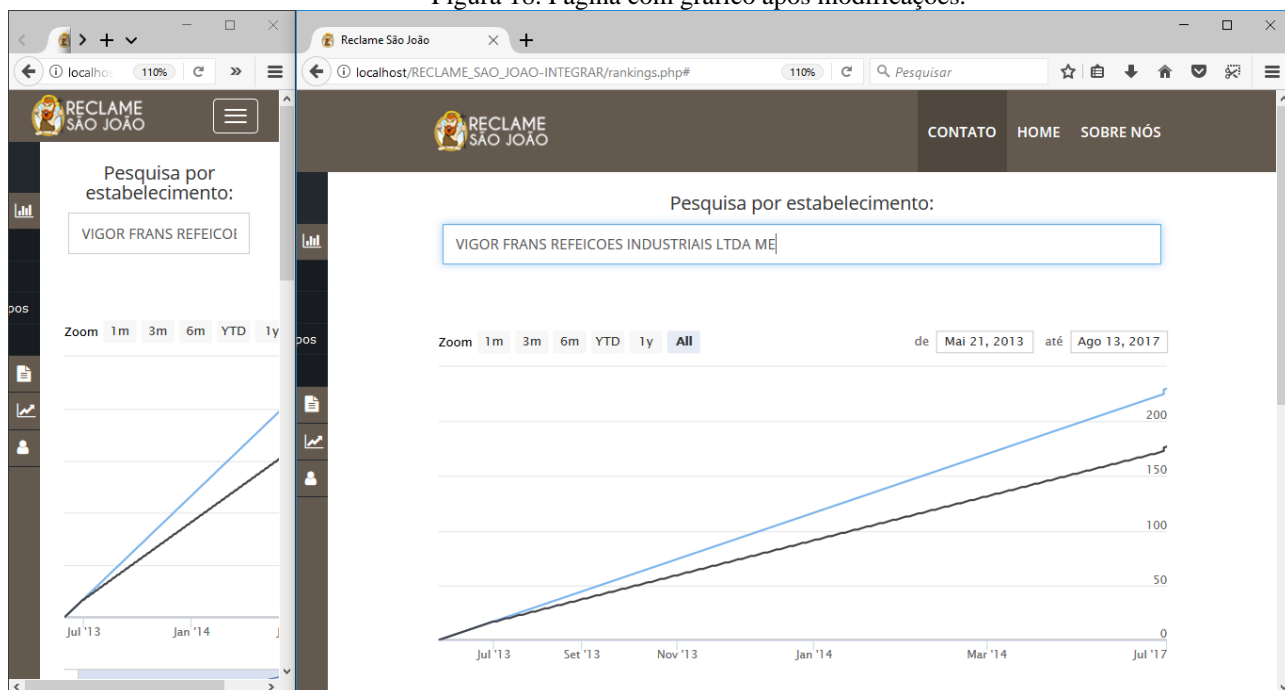


Figura 18. Página com gráfico após modificações.



### 3 Conclusão

Com base no enfoque deste trabalho em tornar a fração do portal Reclame São João responsável pela exibição de gráficos e relatórios responsiva, juntamente com a aplicação e análise do principal método criação de *designs* responsivos, foi possível fazer com que esta seção seja acessada com boa navegabilidade em quaisquer dispositivos.

O uso do *framework Bootstrap* pode ser dado como o principal facilitador desta questão, sendo que tornou necessário apenas o uso de algumas classes previamente definidas no decorrer da estrutura da página.

Seguindo toda a metodologia destacada no corpo do trabalho, as análises necessárias foram dadas, assim como a fase de adaptação da página e seus testes alinhados a aplicação das técnicas. Ainda que hajam muitos outros *frameworks* que possam facilitar ainda mais a criação de trabalhos como este, os quais não foram citados, para o cumprimento do principal objetivo o uso deles não se mostrou necessário.

A necessidade de observar cuidadosamente a documentação do *framework* utilizados pode ser dada como o maior dos problemas durante o desenvolvimento deste trabalho, apesar de não se tornar uma barreira para sua realização.

Em um futuro momento, buscando aprimorar o portal, o uso de um *framework* ainda mais evoluído, tal qual o *Laravel*, tem o potencial de tornar o desenvolvimento mais eficiente, auxiliando não só nos ajustes da estrutura da página, como também processadores de hipertexto.

### 4 Referências Bibliográficas

DA SILVA, ARTHUR DE ALMEIDA PEREIRA. **Design Responsivo: técnicas, frameworks e ferramentas**. 2014.

ZEMEL, Tércio. **Web Design Responsivo: páginas adaptáveis para todos os dispositivos**. Editora Casa do Código, 2015.