

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
DE SÃO PAULO**

Campus São João da Boa Vista

Relatório Técnico

4º ano – Curso Técnico em Informática

Prof. Breno Lisi Romano

**UTILIZAÇÃO DE FRAMEWORKS NO DESENVOLVIMENTO DA  
ÁREA ADMINISTRATIVO DO PROJETO RECLAME SÃO JOÃO**

Aluno: Gustavo Henrique Moisés

Prontuário: 1420534

São João da Boa Vista – SP

2017

## Resumo

O projeto Reclame São João é uma plataforma web que visa estabelecer um ambiente de comunicação entre estabelecimentos e consumidores no município de São João da Boa Vista, além de permitir que os consumidores avaliem ou realizem determinada reclamação ao estabelecimento. É propósito deste trabalho esclarecer o funcionamento do painel administrativo do website Reclame São João, ressaltando sua importância para o funcionamento do mesmo, detalhando as funcionalidades da aplicação juntamente com os métodos de elaboração, que foi caracterizada em um processo de desenvolvimento utilizando *frameworks*, que são ferramentas que proporcionam funcionalidades que facilitam e organizam a criação de um software. Os resultados obtidos mostram o quão importante foi a utilização de *frameworks* no portal Reclame São João, conseguindo atender as necessidades atuais comumente impostas na criação de uma aplicação (organização, rapidez, usabilidade, layout bonito/responsivo...).

**Palavras-chave:** Internet. Website. Desenvolvimento. Ferramentas. Framework.

## Sumário

1	Introdução .....	4
2	Desenvolvimento .....	5
	2.1 Painel Administrativo .....	5
	2.1 MVC .....	9
	2.2 Bootstrap .....	9
	2.3 Datatables.....	11
	2.4 Validação de formulários - Bootstrap Validator .....	13
	2.5 jQuery .....	14
	2.6 Até que ponto utilizar <i>framework</i> é benéfico? .....	15
3	Conclusões e Recomendações .....	16
4	Referências Bibliográficas .....	17

# 1 Introdução

Com o advento da tecnologia e facilidade do uso da internet o número de pessoas conectadas vem aumentando significativamente, atualmente já são cerca de 3,2 bilhões de pessoas conectadas na rede por meio de celulares, tablets, computadores ou outros dispositivos, segundo dados divulgados pela União Internacional das Telecomunicações (UIT). Com isso chegamos à conclusão que a vida digital já é realidade comum e necessária para muitos.

A utilização da internet facilita nossas vidas, a ponto de não ser mais necessário se deslocar até uma loja para realizar compras, ou então, ir até mesmo a uma biblioteca para ler um livro, tudo isso já é possível graças à internet e seus conjuntos de páginas web[1], conhecidas também como sites. Com o número de usuários crescendo absurdamente, a tendência é que cada vez mais você consiga solucionar seus problemas por meio da rede de computadores, visto que novas aplicações webs estão sendo criadas a todo instante.

Diante dessa situação foi proposto a nossa turma do 4º ano em Informática a criação de uma página web que possibilitasse o consumidor da cidade de São João da Boa Vista a avaliar e efetuar uma reclamação sobre determinado estabelecimento da cidade, aplicação essa que recebeu o nome de “Reclame São João”.

Contudo, como em qualquer programa de computador, existe um código complexo por trás dessas páginas que é responsável pela manipulação e exibição das informações, e com o aumento do mercado e necessidade de aprimoramento das funcionalidades dos sites, torna-se cada vez mais importante a adoção de *frameworks* no desenvolvimento visando tornar a criação de sites mais fáceis e mais rápidos. Os *frameworks* webs são ferramentas que auxiliam nas principais atividades no desenvolvimento dessas páginas, como processamento de dados e separação em camadas da aplicação, além de poder proporcionar também componentes reutilizáveis para serem utilizados diversas vezes na aplicação[2] e por esses motivos que optamos pela utilização de alguns *frameworks* para facilitar na hora do desenvolvimento do painel administrativo do “Portal Reclame São João”

Com isso, apresentamos este trabalho que tem como base detalhar os *frameworks* utilizados por nós, buscando exemplificar suas funcionalidades juntamente com o destaque dos benefícios que a utilização de tal ferramenta trouxe para o nosso projeto.

## **2 Desenvolvimento**

### **2.1 Painel Administrativo**

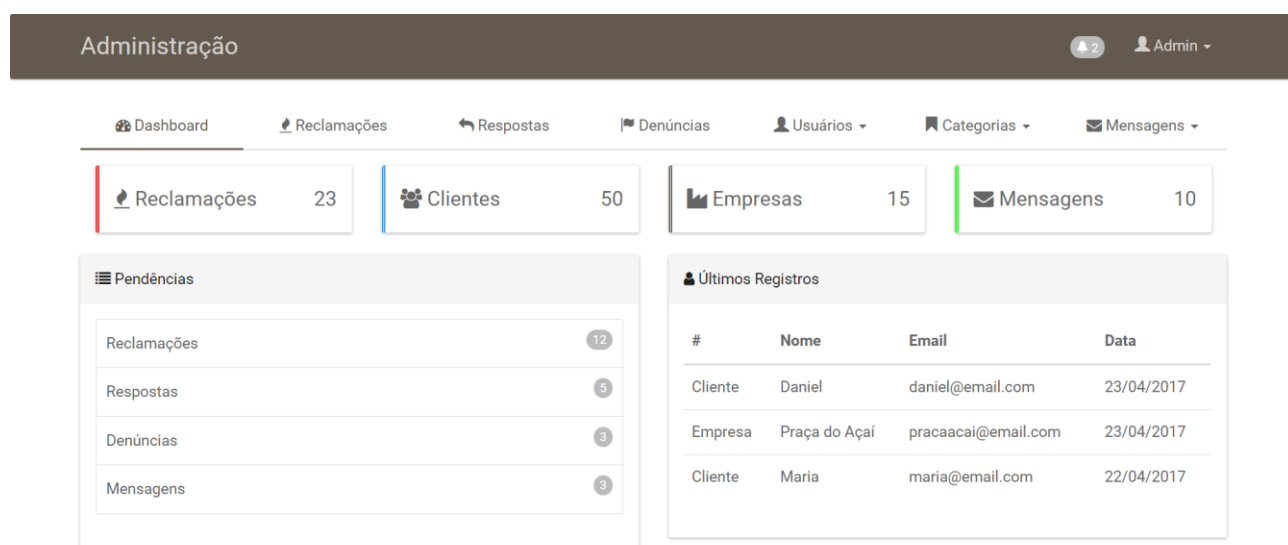
O projeto Reclame São João vai ser desenvolvido por toda a turma do 4º ano, totalizando 35 alunos, e por esse motivo o projeto foi dividido em 5 grupos e cada grupo responsável por uma parte do projeto, sendo os módulos caracterizados por:

- Usuários
- Reclamações
- Estabelecimentos
- Relatórios
- Administrativo

Todos eles interagem diretamente com o usuário final, exceto um, o administrativo, que é um painel restrito a determinadas pessoas, comumente membros da empresa ou organização detentora da aplicação, porém nem por isso é menos importante que os demais, muito pelo contrário, o painel administrativo possui o importante papel de gerenciar e controlar o portal em si, é a partir dele que podemos aprovar ou reprovar uma reclamação após uma validação, tendo o controle de realizar uma censura contra qualquer mensagem que não esteja de acordo com as regras, além de outras funções, como: controle de usuários, reclamações, punições... a ideia por trás de um painel de administração é que para todo dado/função do site tenhamos a possibilidade de manutenção dessas informações na área administrativa, por exemplo, se temos uma página de contato também necessitamos de algo que controle isso no painel administrativo.

A seguir temos uma imagem da tela inicial:

Figura 1 – Página inicial do painel administrativo



Fonte: Autoria própria (2017)

Sendo a estrutura do painel administrativo composta pelos itens:

## 1. Dashboard

O Dashboard refere-se a tela inicial do painel administrativo, exibindo informações gerais da aplicação, como quantidade de usuários, mensagens, validações e últimos usuários registrados.

## 2. Reclamações

É na aba reclamações que podemos manipular as reclamações de nossa aplicação, nela temos a possibilidade de buscar alguma em específico, visualizar e alterar determinada reclamação, além de controlarmos o status da reclamação, Aprovando/Reprovando/Removendo.

### 3. Respostas

Toda reclamação de um cliente deve ser capaz de receber replica da empresa alvo, e é nessa aba que controlamos todas as respostas de reclamações contidas no website, sendo importante destacar que para que uma resposta seja divulgada em nosso site, ela primeiramente precisa ser aprovada pelo administrador, e é nessa página que isso ocorre.

### 4. Denúncias

Em um sistema de troca de mensagens como o nosso, onde as pessoas buscam compartilhar algo insatisfatório, pode ocorrer da mensagem possuir ofensas ou algo que possa constranger o destinatário, e com a possibilitar de enviar denúncias aos administradores, criamos um cenário mais organizado no nosso portal, nesta seção o administrador recebe a denúncia com todas informações, como: denunciado, denunciante, tipo da denúncia, além da mensagem em si reportada. Possibilitando o administrador tomar as medidas cabíveis, podendo realizar advertência textual, restringir o acesso do usuário denunciado ou então bani-lo permanentemente de nosso sistema.

### 5. Usuários

#### 5.1. Clientes

A aba Clientes é a página que armazena todas as informações do usuário – cliente, que é a pessoa frequentadora do comercio, ela que poderá realizar uma reclamação ou avaliar um estabelecimento. Com a listagem dos usuários, podemos expandir seus dados pessoais e de acesso para possíveis necessidades de alteração, além de permitir aos administradores a ação de inativar o usuário, incapacitando-o de utilizar nosso portal.

#### 5.2. Estabelecimentos

A página Estabelecimentos segue o mesmo padrão da aba de Clientes, diferenciando apenas em certas informações, tais como, CNPJ, Público alvo, site e outros

#### 5.3. Administradores

É nesta seção que manipulamos a equipe administrativa, podendo remover determinado staff e também possibilitando a adição de um novo, sendo o grupo administrativo subdividido em 2 ranks: Administradores e Moderadores, definindo permissões inferiores aos Moderadores equiparado aos Administradores, como, incapacitar o staff Moderador de gerenciar os usuários.

#### 5.4. Banidos

Se podemos banir um usuário, também devemos possuir uma área capaz de visualizar os banimentos além de podermos também remover uma punição.

### 6. Categorias

#### 6.1. Tipos de Reclamações

Toda reclamação está vinculada a uma categoria que a define, e o administrador possui o papel de gerenciar os tipos de reclamação que serão utilizadas nas reclamações

#### 6.2. Tipos de Estabelecimentos

No cadastro de um estabelecimento, o mesmo precisa definir qual ramo de atividade a que ele pertence, e mais uma vez quem define os tipos é o administrador

#### 6.3. Tipos de Produtos e Serviços

O usuário estabelecimento é capaz de definir os produtos e serviços que ela presta e novamente o administrador é quem manipula os tipos possíveis

### 7. Mensagens

#### 7.1. Recebidas

O nosso portal disponibiliza-se de um meio comunicação direto entre usuários e administradores, o “Fale conosco” e é na página “Mensagens > Recebidas” que listamos os contatos realizados e possibilitamos também uma possível réplica do staff ao usuário

#### 7.2. Respondidas



Nessa aba permitimos que os administradores visualizem as mensagens já respondidas, contendo a mensagem do usuário, juntamente com a resposta do administrador

## 2.1 MVC

A princípio definimos que no desenvolvimento deveríamos utilizar o modelo Model-View-Controller (MVC) que é um padrão de arquitetura que separa a parte lógica da aplicação da parte visual, e com isso foram propostos diversos *frameworks* PHP que dispunham de tal arquitetura. Entretanto, devido a complexibilidade de utilização de tais ferramentas, optamos por utilizar o modelo sem a utilização de qualquer *framework* PHP, formando a estrutura de pastas da seguinte forma:

- Modelo  
Classes php referente aos dados de nosso sistema (usuários, mensagens, reclamações) com seus respectivos atributos e métodos
- Visão  
Parte visual: páginas, imagens, css, javascript...
- Controle  
Classes php responsáveis pelas queries com o banco de dados
- PHP  
Arquivos php comuns: validações de dados, fluxo de dados de formulários, envio de e-mail e outros...

Embora a lógica MVC implantada por nós seja questionável, ou seja, não aplicamos o conceito correto, ela atendeu nossas necessidades de separar boa parte lógica da visual, mesmo que ainda seja necessário reformulação em possíveis futuras versões.

## 2.2 Bootstrap

Já com a estruturação básica definida começou-se o processo de criação dos protótipos, que nada mais é que a versão funcional sem php, apenas com .html .css e .js com dados fixos. E a partir desse processo que se iniciou a utilização de *frameworks*, começando com o Bootstrap, que é um

famoso *framework* de HTML, CSS e JS responsivo, ou seja, além de termos todo o suporte com elementos padrões de um site, também possibilitaria a criação de nosso portal de maneira responsiva, capaz de se adaptar aos diferentes dispositivos existentes no mercado atual: celulares, tablets, desktops... A capacidade de criar sites responsivos foi determinante para a escolha de tal ferramenta, pois atualmente o universo de aplicações webs é absurdamente grande e atender os usuários da melhor forma independentemente do dispositivo já não é um diferencial e se faz necessário para o sucesso de uma aplicação[3].

O Bootstrap possui uma enorme gama de opções, praticamente tudo que você pode fazer em html ou javascript é suportado também pelo Bootstrap, porém de forma muito mais fácil, devido a sua grandeza, a busca por informações e tutoriais não foi algo dificultoso, boa parte dos recursos existentes no Bootstrap estão dispostos de maneira didática no website: <https://www.w3schools.com/>

Com esse *framework* conseguimos abandonar um pouco o processo de estilização com css, pois ele já possui classes estilizadas sendo necessário apenas defini-las no nosso código html, ou seja, praticamente poderíamos criar o painel administrativo sem escrever se quer uma linha de código css, mas claro, foi-se necessário alguns estilos para mantermos o padrão visual e definir certos elementos particulares.

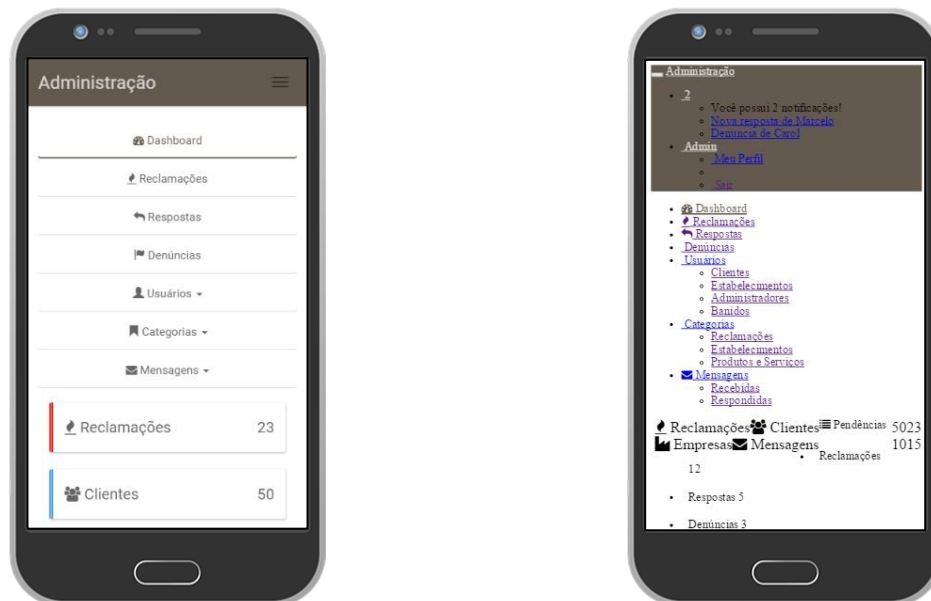
Os benefícios de se utilizar o Bootstrap são tantos: não precisamos nos preocupar diretamente com o css para criar uma aplicação com uma aparência legal, economizamos dramaticamente em linhas de código e tornamos o processo de desenvolvimento muito mais rápido e ainda não precisamos nos preocupar em ficar criando condições de layout para os possíveis diferentes dispositivos dos usuários que visitarão nosso site.

Abaixo temos um belo exemplo da diferenciação do site com o bootstrap e sem:

Figura 2 – Comparação na utilização ou não do *framework* Bootstrap

Com bootstrap:

Sem bootstrap:



Fonte: Fonte: Autoria própria (2017)

## 2.3 Datatables

A área administrativa em si manipula muitos dados e uma maneira de exibir esses dados de maneira organizada ocupando pouco espaço é através da utilização de tabelas, porém se faz necessário mais que simplesmente exibir os dados em tabelas! E se eu precisar pesquisar uma informação de uma tabela com centenas ou até milhares de dados? Ou então, se realmente existir muitos dados, eu irei lista-los todos em uma única tela? A partir dessas e outras perguntas optamos por utilizar o *framework* conhecido como “Datatables”, ele traz uma série de vantagens:

- Paginação no caso de muitos resultados
- Buscador capaz de pesquisar de maneira instantânea qualquer dado da tabela
- Ordenação dos dados através de colunas
- Suporta diversas configurações (remover ex: paginação)
- Compatibilidade com diversos temas, inclusive o Bootstrap

A seguir temos um exemplo da utilização do Datatables:

Figura 3 – Demonstração das funcionalidades do *framework* Datatables

Administração

Dashboard Reclamações Respostas Denúncias Usuários Categorias Mensagens

Mostrar 10 registros

Procurar:

ID	Título	Reclamação	Usuário	Empresa alvo	Data	Status	Ações
0	Produto com defeito	texto texto texto texto...	Rosangela	BigBom	22/04/2017	Pendente	Gerenciamento
1	Produto com defeito	texto texto texto texto...	Rosangela	BigBom	22/04/2017	Publicado	Gerenciamento
2	Produto com defeito	texto texto texto texto...	Rosangela	BigBom	22/04/2017	Suspensa	Gerenciamento
3	Produto com defeito	texto texto texto texto...	Rosangela	BigBom	22/04/2017	Pendente	Gerenciamento
4	Produto com defeito	texto texto texto texto...	Rosangela	BigBom	22/04/2017	Suspensa	Gerenciamento
5	Produto com defeito	texto texto texto texto...	Rosangela	BigBom	22/04/2017	Suspensa	Gerenciamento
6	Produto com defeito	texto texto texto texto...	Rosangela	BigBom	22/04/2017	Publicado	Gerenciamento
7	Produto com defeito	texto texto texto texto...	Rosangela	BigBom	22/04/2017	Pendente	Gerenciamento
8	Produto com defeito	texto texto texto texto...	Rosangela	BigBom	22/04/2017	Suspensa	Gerenciamento
9	Produto com defeito	texto texto texto texto...	Rosangela	BigBom	22/04/2017	Pendente	Gerenciamento

Mostrando de 1 até 10 de 26 registros

Anterior 1 2 3 Seguinte

Fonte: Autoria própria (2017)

Observe que podemos limitar a quantidade de registro por página<sup>1</sup> e ainda buscar algum dado específico da tabela<sup>2</sup>, sem contar a ordenação por coluna<sup>3</sup> e a paginação<sup>4</sup>. Tudo isso feito automaticamente pelo Datatables em uma tabela criada com html comum, sendo necessário apenas definir o nome do “id” da tabela no seguinte javascript no lugar do #example:

Figura 4 – Exemplo de código para implementação do Datatables

```
1 $(document).ready(function() {  
2     $('#example').DataTable();  
3 } );
```

Fonte: [https://datatables.net/examples/basic\\_init/zero\\_configuration.html](https://datatables.net/examples/basic_init/zero_configuration.html) (2017)

## 2.4 Validação de formulários - Bootstrap Validator

Praticamente toda manutenção de informação passa através de um formulário, seja para inserir ou editar, e em sistemas seguros esse dado precisa ser tratado antes de fazer alterações no banco de dados, atualmente possuímos dois tipos de validação de dados, um no lado cliente com Javascript e um no lado servidor com PHP. Em um mundo perfeito apenas a validação no lado cliente já seria o bastante, porém se garantir apenas com ela é algo inseguro atualmente, pois o Javascript pode ser manipulado pelo o usuário, se tem uma validação de campo, ele facilmente consegue burla desativando o javascript do browser, o que não ocorre nas validações do lado servidor que já são mais complexas de serem burladas. Muitos se perguntam, para que criar a validação no lado cliente se criando apenas no lado servidor já seria o suficiente?... Pois a validação pelo javascript torna a aplicação mais profissional podendo exibir mensagens de erro instantaneamente nos campos antes do usuário tentar submeter o formulário.

Para realizar tais validações também optamos por utilizar um *framework*, conhecido como “Bootstrap Validator” ele é capaz de criar toda uma validação complexa dos dados com a inserção de poucos elementos no código, com tal aplicação, podemos validar campos do tipo “repita a senha” da seguinte maneira:

Figura 5 – Exemplo de código com a utilização do Bootstrap Validator

```
<div class="form-group">
  <label for="inputPassword" class="control-label">Password</label>
  <div class="form-inline row">
    <div class="form-group col-sm-6">
      <input type="password" data-minlength="6" class="form-control" id="inputPassword" placeholder="Password" r
      equired>
      <div class="help-block">Minimum of 6 characters</div>
    </div>
    <div class="form-group col-sm-6">
      <input type="password" class="form-control" id="inputPasswordConfirm" data-match="#inputPassword" data-mat
      ch-error="Whoops, these don't match" placeholder="Confirm" required>
      <div class="help-block with-errors"></div>
    </div>
  </div>
</div>
```

Fonte: <http://1000hz.github.io/bootstrap-validator/> (2017)

Nos formulários que iremos utilizar esse *framework* de validação precisamos definir os seguintes atributos:

Figura 6 – Código inserido para a implementação do Bootstrap Validator no formulário

```
<form data-toggle="validator" role="form">
```










Fonte: <http://1000hz.github.io/bootstrap-validator/> (2017)

E em cada campo declarar o “required” o resto ele faz automaticamente:

- Mensagens de erro instantânea
- Compatibilidade com diversos input types: email, number, date...
- Possibilidade de requisição Ajax para validar o campo

A seguir temos o exemplo da utilização do “Bootstrap Validator” na edição de usuários:

Figura 8 – Demonstração das validações/mensagens de erro do Bootstrap Validator

Nome:	E-mail:		
 <input type="text" value=""/>	<input type="text" value="cliente01"/>		
 Preencha este campo.	 Inclua um "@" no endereço de e-mail. "cliente01" está com um "@" faltando.		
CPF:	Data de Nascimento:		
 <input type="text" value="0932133412"/>	 <input type="text" value="09/02/1998"/>		
Rua:	Bairro:	Número:	Complemento:
 <input type="text" value="Rua 965"/>	 <input type="text" value="Bairro 965"/>	 <input type="text" value="2"/>	 <input type="text" value="Apartamer"/>

Fonte: Autoria própria (2017)

Perceba que ele cria as mensagens da necessidade de inserção de um valor no campo “nome” e identifica a invalidade do “e-mail” informado.

## 2.5 jQuery

E por último, mais não menos importante, o jQuery, que é definido como uma biblioteca e não *framework*, porém de extrema utilizada, só para se der ideia, todos *frameworks* mostrados anteriormente requisitam inserção do jQuery para funcionarem... O jQuery é uma biblioteca JavaScript, ou seja, muda a forma padrão de se utilizar JavaScript, tornando a codificação muito

mais rápida e concisa que simplifica percorrer documentos HTML, manipular eventos, criar animações e interações Ajax. Tarefas comuns que necessitam várias linhas de código JavaScript são substituídas no jQuery em métodos que podemos chamar em uma única linha.

Figura 9 – Logo da biblioteca JavaScript - JQuery



Fonte: [http://www.vikaskbh.com/wp-content/uploads/2014/01/jquery\\_logo.png](http://www.vikaskbh.com/wp-content/uploads/2014/01/jquery_logo.png) (2017)

“escreva menos, faça mais”

## 2.6 Até que ponto utilizar *framework* é benéfico?

Demonstramos diversos *frameworks* utilizados no nosso projeto e seus benefícios, porém se faz necessário esclarecer o outro lado, as desvantagens que a utilização de tais ferramentas pode trazer.

Informamos que para a implantação de tais *frameworks* a escrita de código é mínima, e isso é verdade, porém por trás desses *frameworks* que incluímos em nosso site costumam possuir centenas ou milhares de linhas de código o que pode deixar o nosso site mais lento, normalmente as pessoas se utilizam de poucas funcionalidades dessas ferramentas e com isso abrimos um questionamento: Será que não compensa eu mesmo criar a funcionalidade que irei utilizar, ao invés de importar bibliotecas imensas? Certamente com a importação dos *frameworks* será mais fácil, porém, será que é o melhor?

Outro ponto a ser levantado é em relação a desenvolvedores iniciantes, pois com o costume da utilização dos *frameworks* o conhecimento passa a ser limitado a tais recursos e não se consegue expandir o conhecimento para fora disso, por exemplo, sempre utilizar o Bootstrap quando se tratar de responsivo e não buscar saber como fazer o mesmo sem utilizar tal ferramenta

### **3 Conclusões e Recomendações**

Como visto, a utilização de *frameworks* pode trazer uma série de benefícios no desenvolvimento de aplicações webs, como: redução de tempo de desenvolvimento, maior organização do projeto, otimização de código escrito e melhores métodos de criação. Porém, se faz necessário também uma análise da real utilidade, buscando coibir sua utilização demasiada que pode prejudicar o aprendizado pessoal na criação de funcionalidades e otimização de aplicações.



## 4 Referências Bibliográficas

[1] VEICER, Sammy. A terceira geração do e-commerce. Disponível em: <<https://ecommercenews.com.br/artigos/cases/a-terceira-geracao-do-e-commerce/>>. Acesso em: 04 nov. 2017.

[2] KNUPP, Jeff. What is a Web Framework? Disponível em: <<https://jeffknupp.com/blog/2014/03/03/what-is-a-web-framework/>>. Acesso em: 04 nov. 2017.

[3] Entenda a importância do layout responsivo para o seu site. Disponível em: <<http://portal.teraware.com.br/entenda-a-importancia-do-layout-responsivo-para-seu-site/>>. Acesso em: 04 nov. 2017.