

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

MATHEUS SARDELI MALHEIROS

**DESENVOLVIMENTO *BACK-END* DAS REGRAS PARA CONTROLE
DOS USUÁRIOS DO PROJETO RECLAME SÃO JOÃO**

SÃO JOÃO DA BOA VISTA – SP

2017

MATHEUS SARDELI MALHEIROS

**DESENVOLVIMENTO *BACK-END* DAS REGRAS PARA CONTROLE
DOS USUÁRIOS DO PROJETO 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

Agradecimentos

Agradeço aos professores do Instituto Federal de Educação, Ciência e Tecnologia de Cão Paulo – Campus São João da Boa Vista que estiveram presentes durante os quatro anos do Curso Técnico Integrado em Informática. Agradeço aos funcionários, diretores e todos presentes durante a realização do curso.

Resumo

O desenvolvimento *Back-end* é essencial para qualquer aplicação do sistema *Web*. O objetivo deste projeto é apresentar a definição do desenvolvimento *Back-end* e sua utilização durante o desenvolvimento das regras de controle do módulo de usuário do projeto Reclame São João, referente ao módulo 01. Motivado pela falta de possibilidades de comunicação entre estabelecimentos e seus consumidores no município de São João da Boa Vista, o portal visa estreitar o canal de comunicação entre os dois comunicadores, e possibilitar que os consumidores avaliem ou realizem determinada reclamação. Contudo, há o controle de usuários que permite o cadastro, o *login* e posteriormente o *login* utilizar funções específicas do *WebSite*. Para cumprir os objetivos do desenvolvimento *Back-end* referentes as regras de controle de usuários, estão sendo utilizadas tecnologias específicas que permitem esse desenvolvimento, como linguagens de programação (PHP), *JavaScript*, HTML, CSS e *JQuery*. Com isso, os resultados são satisfatórios, visto que as funções essenciais para o funcionamento estão desenvolvidas, tais como, cadastro, *login*, e a permissão para utilizar funções específicas.

Palavras-chaves: Desenvolvimento, *Back-End*, Reclame São João.

Lista de Figuras

Figura 1 – Cronograma	5
Figura 2 – Macro-requisitos dos módulos do projeto	6
Figura 3 – Macro-requisitos do módulo de usuários	7
Figura 4 – Funcionamento <i>Front-end</i> e <i>Back-end</i>	9
Figura 5 – Comparação <i>Front-end</i> e <i>Back-end</i>	9
Figura 6 – Representação dos lugares de servidor, aplicação e banco de dados.....	10
Figura 7 – Notepad++	13
Figura 8 – HTML 5	14
Figura 9 – CSS 3	14
Figura 10 – <i>JavaScript</i>	15
Figura 11 – PHP	16
Figura 12 – <i>jQuery</i>	16
Figura 13 – <i>Microsoft Word</i>	17
Figura 14 – Página de cadastro do consumidor com <i>layout</i> estilizado	18
Figura 15 - Criar a Conexão com o Banco de Dados.....	19
Figura 16 – Criar novo Usuário, Consumidor e Localização.....	19
Figura 17 – Código para Estilizar o <i>Layout</i> da Página.....	20
Figura 18 – Selecionar dados inseridos na página de cadastro	21
Figura 19 – Classe que possui os Métodos <i>get()</i> e <i>set()</i>	22
Figura 20 – Página UsuárioDAO e função de inserir Usuário	23

Sumário

1	- Introdução	1
2	- Desenvolvimento	3
2.1	- Tema.....	3
2.2	- Objetivos – geral e específico (s).....	3
2.3	- Justificativa	3
2.4	- Metodologia	4
2.5	- Planejamento do projeto.....	4
2.5.1	- Cronograma	5
2.5.2	- Macro-requisitos dos Módulos do Projeto	6
2.5.3	- Macro-requisitos do Módulo de Usuários	7
2.6	- Desenvolvimento do projeto	8
2.7	- Instrumentos utilizados	13
2.7.1	- Notepad++.....	13
2.7.2	- HTML	13
2.7.3	- CSS.....	14
2.7.4	- <i>JavaScript</i>	15
2.7.5	- PHP.....	15
2.7.6	- <i>jQuery</i>	16
2.7.7	- <i>Microsoft Word</i>	17
2.8	- Página de Cadastro de Consumidor.....	18
	Conclusões e Recomendações	24
	Referências Bibliográficas	25

1 - Introdução

Os desenvolvedores *Front-end* e *Back-end* estão presentes em todos os *softwares* e *Websites* que existem e são criados. Entretanto, apesar de suas diferenças, os desenvolvedores de *Front-end* e *Back-end* se completam na maioria dos casos.

O desenvolvedor *Front-end* é responsável por "dar vida" à interface. Trabalha com a parte da aplicação que interage diretamente com o usuário, ou seja, trabalha diretamente na parte visual da aplicação.

O desenvolvedor *Back-end* trabalha na parte de "trás" da aplicação. Ele é o responsável, em termos gerais, pela implementação da regra de negócio. [1]

O tema proposto é o Desenvolvimento *Back-end* das regras para controle dos usuários do projeto Reclame São João, dando ênfase no desenvolvimento da codificação do Sistema *Web*, ou seja, desenvolver cada função ou regra que o sistema irá realizar, enfatizando os controles relacionados aos usuários.

O tema foi escolhido devido a maior facilidade em desenvolver as partes das funções, ao invés de *layout*. E, portanto, há um interesse pessoal em aprimorar os funcionamentos e o desenvolvimento em questão.

O presente trabalho tem como base mostrar e esclarecer a importância que o desenvolvimento *Back-end* possui dentro de qualquer *Website* disponível, pois esse desenvolvimento é responsável por criar as regras e funções que o *Website* realiza, visto que grande parte da população não tem o conhecimento desse desenvolvimento, já que estão ocultas por trás das interfaces gráficas.

Sendo assim, a pesquisa será baseada no processo de desenvolvimento do módulo de usuários, no qual faz parte do projeto do Reclame São João.

Sabe-se que a comunicação é responsável pelo sucesso ou fracasso em qualquer situação, por exemplo: em um ambiente de trabalho ou até mesmo em uma comunicação dentro da sociedade, e por isso é necessário que sempre haja meios para facilitar esta comunicação, estreitando caminhos e abrangendo maior quantidade de lugares.

A comunicação entre estabelecimentos e seus consumidores no município de São João da Boa Vista está passando por dificuldades, e isso faz com que os consumidores não consigam expor seus pontos de vistas em relação a algum lugar, o que impossibilita o lugar de procurar melhorias para melhor atendê-los.

Com vista em estreitar o canal de comunicação entre os dois comunicadores, o Instituto Federal São Paulo – Campus São João da Boa Vista juntamente com a prefeitura de São João da Boa Vista realiza um projeto com os alunos cursando o 4º Ano do Técnico em Informática integrado ao Ensino Médio no desenvolvimento do sistema *Web* que irá facilitar e melhorar a comunicação consumidores e estabelecimentos, possibilitando a população reclamar, avaliar, apresentar sugestões de melhorias as empresas para melhor atender os clientes e buscar a localização de determinada corporação e sua reputação.

O projeto realizado pelos alunos do IFSP-Campus São João da Boa Vista apresenta um sistema *Web* para estreitar o canal de comunicação entre estabelecimentos da cidade de São João da Boa Vista, São Paulo – Brasil e os seus consumidores, possibilitando a população reclamar, avaliar, apresentar sugestões de melhorias as empresas para melhor atender os clientes e buscar a localização de determinada corporação e sua reputação; portanto é denominado Reclame São João.

Para melhor desempenho e desenvolvimento houve uma divisão entre os alunos em cinco módulos, cada módulo representa uma parcela do projeto final, sendo eles: Usuários, Reclamações, Estabelecimentos, Relatórios e Administrativo.

Com a melhoria na comunicação entre as duas partes, os consumidores poderão esclarecer o que pode ser melhorado e com isso permitirá que os estabelecimentos possam aprimorar para melhor atender seus clientes.

Tendo em vista que a tecnologia está presente em quase todo o território brasileiro e que existem diversos tipos de tecnologias: TV, computadores, *smartphones*; os consumidores que utilizam elas não possuem conhecimento de como eles são fabricados, como eles foram programados e como eles funcionam.

Entretanto, essa pesquisa auxiliará essas pessoas a terem um pouco de conhecimento referente as informações de um desenvolvimento *Back-end* e a sua influência de grande valor para os profissionais de informática e desenvolvedores de sites e programas.

O objetivo geral deste trabalho é apresentar o desenvolvimento *Back-end* do módulo de Usuários do Projeto Reclame São João.

Os materiais que serão utilizados para a realização desse trabalho são as páginas de desenvolvimento do módulo de Usuários do portal Reclame São João, e para a programação será utilizado algumas linguagens de programação, tal como Linguagem de Marcação de Hipertexto (HyperText Markup Language – HTML), Pré Processador de Hipertexto (*Hypertext Preprocessor* – PHP), *JavaScript*, Folhas de Estilo em Cascata (*Cascading Style Sheets* – CSS),

jQuery, além de alguns programas como *MySQL* e *Sublime Text*, estruturas de *Framework* Modelo-visão-controlador (*Model-view-controller* – MVC).

2 – Desenvolvimento

2.1 - Tema

O tema proposto é o desenvolvimento *Back-end* do projeto Reclame São João que irá ajudar na comunicação entre os consumidores e os estabelecimentos, permitindo as pessoas avaliarem e reclamarem de qualquer serviço que desejarem. O desenvolvimento *Back-end* é responsável por criar as regras e funções que o *Website* realiza. Sendo assim, a programação destinada a esse desenvolvimento não é de grande conhecimento das pessoas, já que está oculta por trás das interfaces gráfica.

Uma importante parte do sistema é a parte de usuário, na qual permite o cadastro de usuário, validação do cadastro, recuperação da senha, alterar informações pessoais, comunicar-se com o administrador utilizando um chat, e também reclamar ou avaliar qualquer estabelecimento que esteja cadastrado.

2.2 - Objetivos – geral e específico (s)

Objetivos gerais:

- Gerenciamento do Portal Reclame São João.

Objetivo específico:

- Gerenciamento dos usuários do Portal Reclame São João, permitindo cadastrar, *logar* e realizar as funções disponíveis.

2.3 - Justificativa

Pretendemos realizar este projeto devido à grande dificuldade de comunicação entre os consumidores e os estabelecimentos no município de São João da Boa Vista. O setor que permite a comunicação entre os clientes e as empresas está em falta e, portanto, isso vem se tornando uma dificuldade, pois várias pessoas gostariam de se comunicar e entrar em contato com os estabelecimentos para realizar uma avaliação ou reclamação.

Este é o principal motivo do desenvolvimento do projeto, e das funções destinadas aos usuários, como cadastrar, *logar* e várias outras.

2.4 - Metodologia

Para a realização do *Website* a metodologia que está sendo utilizada é:

- Para a realização do *Website* está sendo utilizado o programa Notepad++, a tecnologia e linguagem HTML5, PHP, *JavaScript*, *Jquery* e para o *Layout* o CSS3;
- *PhotoShop* foi utilizado para o *design* do logo do Portal Reclame São João e imagens utilizadas no *Website*;

2.5 – Planejamento do projeto

Devido à falta de comunicação entre os consumidores e os estabelecimentos, nos planejamos fazer este projeto. Ele contará com a possibilidade de os usuários avaliarem ou reclamarem de alguma empresa que esteja cadastrada no Portal Reclame São João, e principalmente o cadastro e *login* de cada usuário.

2.5.1 – Cronograma

Figura 1 – Cronograma

Cronograma de Atividades do Projeto PDS 2017 Vespertino					
	Status	Data Real de Entrega	Data Inicialmente Prevista	Título da Atividade	Descrição
1º Bimestre	#01		06/fev	Estudo de Caso	Decidir qual será o Contexto do Projeto Acadêmico??
	#02		06/fev	Definição da Plataforma de Desenvolvimento	Linguagem de Programação e Banco de Dados
	#03		06/fev	Definição das Equipes	Análises de Sistemas, Desenvolvedores, DBAs, Testadores e Equipe de Documentação
	#04		13/fev	Definição dos Módulos (Macroações)	Definição dos Módulos do Estudo de Caso escolhido e os respectivos macro requisitos a serem contemplados
	#05		13/fev	Definição das Responsabilidades de Gerência	Divisão das responsabilidades de gerência entre os integrantes de cada um dos módulos do projeto.
	#06		20/fev	Termo de Abertura do Projeto	Definir o Contrato Inicial do Projeto
	#07	13/mar	06/mar	Identificação dos Stakeholders	Nome, informações de Contato e Classificação do Stakeholder (Cliente, Alunos, Governo, Fornecedor, Parceiro, etc.)
	#08	13/mar	06/mar	Documento de Visão para Cada Módulo	Utilizar o Template do RUP para definir Problema e Solução Proposta no Documento de Visão.
	#09	27/mar	20/mar	Definir o Template do Projeto	Os desenvolvedores de todos os módulos devem se reunir e propor um template para o projeto. Definir o logo da empresa. Já deve ser implementado este template em HTML+CSS+PHP. Deve ser definido o padrão de pastas para o desenvolvimento (imagens, css, javascript, etc...)
	#10	27/mar	20/mar	Documento de Especificação dos Principais Envolvidos para cada Módulo	Utilizar o Template do RUP para Especificação dos Principais Envolvidos (Stakeholders + Requisitos Funcionais)
	#11		03/abr	Definição Template de Apresentação Bimestral e Assuntos a Serem Abordados	Definir o Template da Apresentação Bimestral como se fosse uma apresentação em uma empresa e definir também os assuntos a serem abordados nesta primeira apresentação
	#12		24/abr	Cases de Uso para cada Módulo e Estimativa de Esforço	Elaborar o Diagrama de Cases de Uso para cada um dos Módulos (Preencher o Documento de MCU do RUP). Elaborar uma estimativa de esforços (custo/tempo) para cada um dos módulos.
	#13		24/abr	Elaboração dos Protótipos de Interface	Elaborar, baseado no template definido, os protótipos de IHVs para cada caso de uso do Projeto.
	#14		28/abr	Apresentação do Primeiro Bimestre	Apresentação das atividades realizadas no Primeiro Bimestre para cada um dos Módulos
2º Bimestre	#15		15/mai	Refinamento das Atividades do Primeiro Bimestre - Por Módulo	Nesta atividade, os alunos devem finalizar todas as outras atividades que estão pendentes até o momento, principalmente elaboração de casos de uso e seus respectivos protótipos.
	#16	22/mai	15/mai	Elaboração do Diagrama Contextual de cada Módulo	Os DBAs serão responsáveis em elaborar o Diagrama Entidade-Relacionamento (MER) para cada um dos módulos (Entidades, Relacionamentos e Atributos) e Definir o Dicionário de Dados.
	#17	22/mai	15/mai	Elaboração do Dicionário de Dados do BD de cada Módulo	Os DBAs serão responsáveis em elaborar o Diagrama Entidade-Relacionamento (MER) para cada um dos módulos (Entidades, Relacionamentos e Atributos) e Definir o Dicionário de Dados.
	#18	29/mai	22/mai	Elaboração do Diagrama de Classes de cada Módulo	O módulo deve ficar responsável em elaborar o Diagrama de Classes, a partir do MER definido, utilizando uma ferramenta CASE para auxiliá-lo. Deve entregar esta atividade na aula de AN2, pois será utilizada para iniciar o desenvolvimento do Projeto.
	#19	29/mai	22/mai	Planejamento: Definir as entregas de Cada Módulo para Implementação e Testes	Agrupar casos de uso diretamente relacionados e definir as entregas de cada módulo, ou seja, o que será implementado e testado em conjunto. Considerar as datas do terceiro e quarto bimestre e também a Semana de Tecnologia.
	#20	03/jun	26/jun	Elaboração dos Casos de Teste para cada Módulo	Os Testadores deverão elaborar, juntamente com a Equipe, os casos de teste para cada um dos Módulos (Preencher o Documento do RUP).
	#21		12/jun	Consolidação do Esquema Físico para o Projeto e do Dicionário de Dados	Os DBAs ficarão responsáveis para consolidar todos os esquemas físicos e dicionários de dados dos módulos em um único para ser utilizado por todos. Os DBAs ficarão responsáveis pelas seguintes atividades: Definir o Dicionário de Dados, Transformar o MER em Modelo Relacional, Converter a Transformação proposta pelo Entrelado e gerar o Modelo Físico.
	#22	03/jun	26/jun	Entrega da Iteração #01 de Casos de Uso	Entrega dos casos de uso da primeira iteração de Casos de Uso de Cada Módulo já implementados e testados (Terceiro Bimestre).
	#23		05/jul	Apresentação do Segundo Bimestre (Incremental)	Apresentação das atividades realizadas no Primeiro e Segundo Bimestres por cada um dos Módulos.
	#24	21/ago	14/ago	Definição do Tema e Apresentação da Objetiva do Relatório Técnico (Vulgo TFC)	Definição do Tema e Apresentação da Objetiva do Relatório Técnico (Vulgo TFC). Válerá parte da nota do 2º e 4º bimestre. Para os alunos que não ficaram entusiasmados, haverá de substituição.
	#25	28/ago	21/ago	Revisão da Iteração #01 de Casos de Uso	Revisão e Refinamento da entrega dos casos de uso da primeira iteração de Casos de Uso de Cada Módulo já implementados e testados (Terceiro Bimestre).
	#26	04/set	28/ago	Entrega da Iteração #02 de Casos de Uso	Entrega dos casos de uso da primeira iteração de Casos de Uso de Cada Módulo já implementados e testados (Terceiro Bimestre).
	#27	25/set	28/ago	Elaboração do Capítulo 01 do Relatório Técnico (Vulgo TFC)	Elaboração do Capítulo 01 do TFC, contendo Contextualização, Motivação da Pesquisa e Objetivo.
	3º Bimestre	#28	30/set	23/set	Entrega da Iteração #03 de Casos de Uso
#29		30/set	02/out	Entrega de Versão Parcial do Capítulo 02 do Relatório Técnico (TFC)	Elaboração de uma versão parcial do Capítulo 02 do TFC, incluindo a Metodologia do Trabalho.
#30			23/set	Apresentação do Terceiro Bimestre (Incremental)	Apresentação das atividades realizadas no Terceiro Bimestre por cada um dos Módulos.
#31			16/out	Integração entre todos os Módulos	Entrega dos casos de uso da primeira iteração de Casos de Uso de Cada Módulo já implementados e testados (Terceiro Bimestre).
4º Bimestre	#32		30/out	Entrega do Relatório Técnico do Projeto (Vulgo TFC)	
	#33		13/nov	Entrega da Iteração #04 de Casos de Uso	Entrega dos casos de uso da primeira iteração de Casos de Uso de Cada Módulo já implementados e testados (Terceiro Bimestre).
	#34		20/nov	Manual do Usuário	Seguir todas as recomendações da comissão de análise a partir do Portal da Disciplina.
	#35		20/nov	Portal de Documentação	
	#36		20/nov	Manual de Instalação	
	#37		27/nov	Apresentação do Quarto Bimestre (Incremental)	Apresentação das atividades realizadas no Quarto Bimestre por cada um dos Módulos.

Fonte: <https://sites.google.com/site/blromano/> (2017)

2.5.2 – Macro-requisitos dos Módulos do Projeto

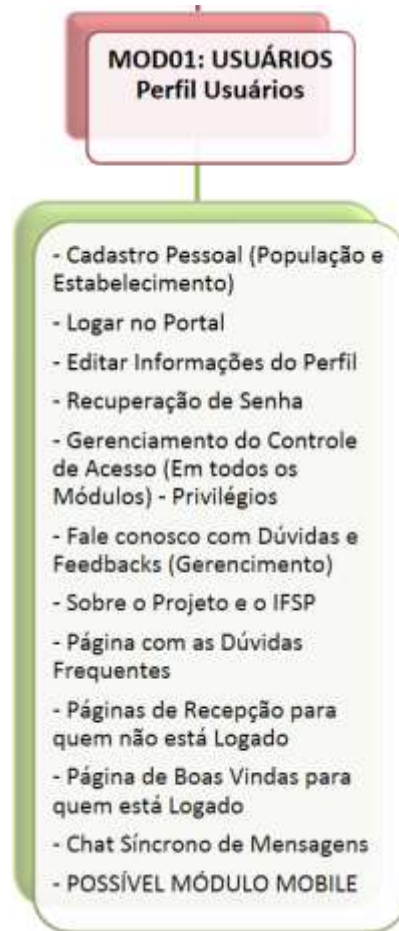
Figura 2 – Macro-requisitos dos módulos do projeto



Fonte: <https://sites.google.com/site/blromano/> (2017)

2.5.3 – Macro-requisitos do Módulo de Usuários

Figura 3 – Macro-requisitos do módulo de usuários



Fonte: <https://sites.google.com/site/blromano/> (2017)

2.6 – Desenvolvimento do projeto

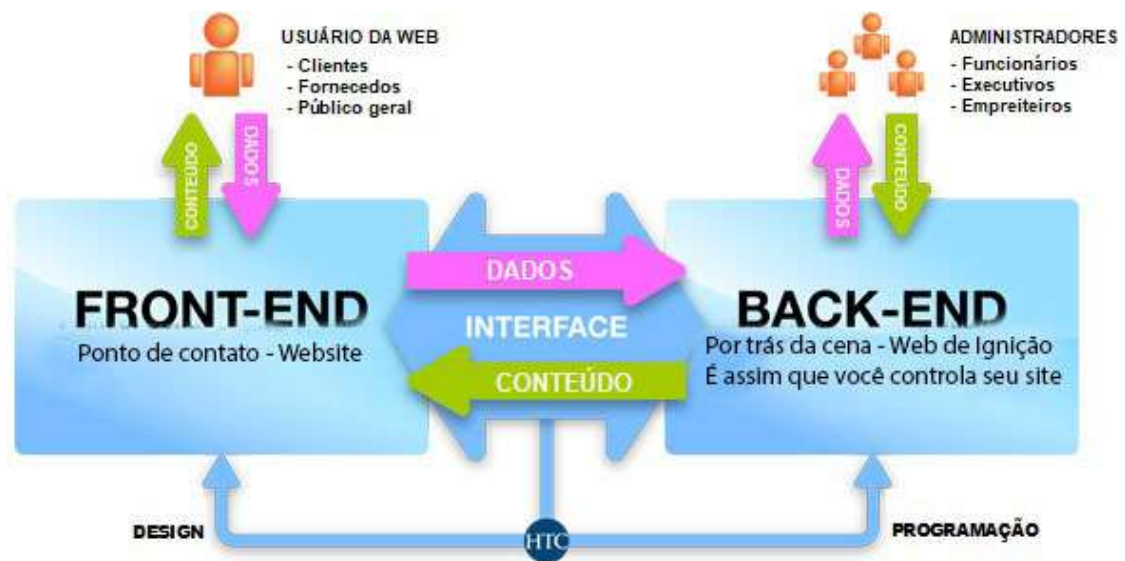
A ideia do projeto surgiu do princípio de que as pessoas necessitavam de um encurtamento para se comunicarem com os estabelecimentos do município de São João da Boa Vista, possibilitando a elas reclamarem ou fazer alguma sugestão de alguma empresa. Com o intuito de solucionar esse problema foi decidido a criação de um *WebSite* que possibilitasse aos consumidores divulgar seu ponto de vista em relação a alguma empresa que desejasse.

Começamos a pôr em pratica dividindo as tarefas em grupos, os quais cada módulo (grupo) era encarregado em realizar o desenvolvimento de uma parte específica do *software*, sendo eles: módulo de usuários, estabelecimentos, reclamações, relatórios e administrativo.

Por ser um projeto grande e que já será implementado no mercado, temos algumas dificuldades, porém com muitas pesquisas, ajudas de professores e colegas de classe, é possível que o projeto seja desenvolvido. A prefeitura do município de São João da Boa Vista que se destina nosso projeto nos ajudou fornecendo a ideia de que era necessária a criação de um *Website* no qual as pessoas pudessem entrar em contato com as empresas, pois diversos cidadãos reclamaram que deveria haver um meio pelo qual pudessem se comunicar com os estabelecimentos para avaliar, fazer alguma sugestão ou alguma sugestão e consequentemente essas empresas tomassem as devidas soluções.

O desenvolvimento do módulo de usuário é responsável pelas funções destinadas aos usuários, principalmente aos consumidores, permitindo a eles realizar o cadastro, ativar cadastro, *logar*, editar informações de perfil, acessar a página de visualização do mapa dos estabelecimentos, entrar em contato com o Reclame São João, recuperar senha, acessar sobre o projeto e o IFSP, acessar a página de boas-vindas após o *login* e acessar a página de dúvidas frequentes, além de comunicar com o administrador do portal através de um *chat* síncrono. Entretanto há uma certa dificuldade no desenvolvimento *Back-end* do mesmo, pois esse desenvolvimento envolve tudo o que não está visível na página da *Web*, mas é necessário para suportar o trabalho do desenvolvedor do *Front-end*.

Figura 4 – Funcionamento *Front-end* e *Back-end*



Fonte: <http://www.3nytechnology.com/website-frontend-and-backend/>, tradução nossa (2017)

Figura 5 – Comparação *Front-end* e *Back-end*



Fonte: <https://blog.mxcursos.com/front-end-ou-back-end-entenda-as-diferencas-e-descubra-o-seu-perfil/> (2017)

O desenvolvimento *Back-end* acontece nos seguintes três lugares:

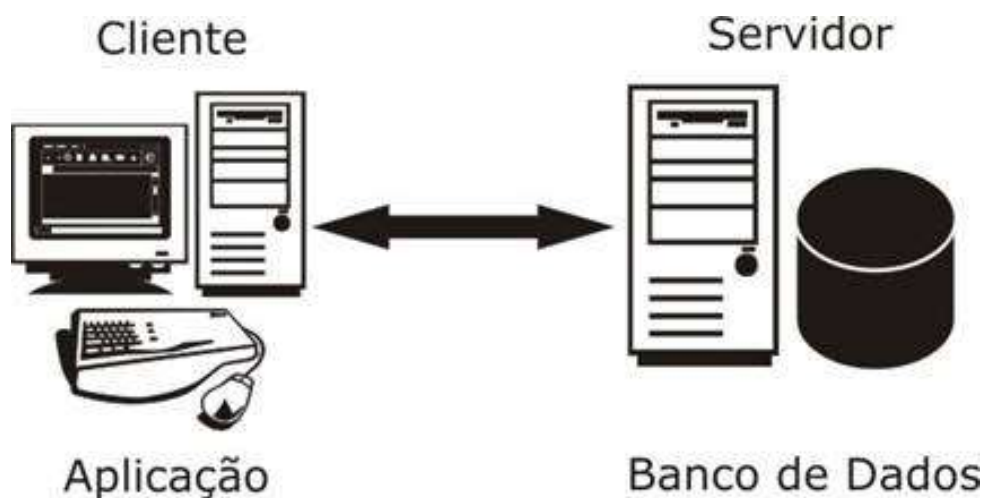
Servidor: O servidor é o computador que hospeda os arquivos de codificação que incluem o aplicativo do site e o banco de dados. Quando você visita www.google.com, por exemplo, seu

navegador *web* solicita a página da *Web* dos servidores do Google, que responde com uma cópia da página da *Web* que você vê no seu navegador.

Aplicação: A aplicação manipula o conteúdo em páginas da *Web* enviadas aos usuários e as alterações feitas no banco de dados. As aplicações são escritas usando linguagens de programação como *Ruby*, *Python* e *PHP*, e só são executadas no servidor. Proficiência em um idioma geralmente é o suficiente.

Banco de dados: O banco de dados armazena os dados do site e do usuário para que ele esteja disponível para futuras sessões de navegação. O banco de dados mais simples é uma planilha do *Excel*, que é apropriada para desenvolvimento *Web*. Bancos de dados como o *Postgres* e o *MongoDB* são otimizados para o uso do site, geralmente um desses bancos de dados é usado por site.

Figura 6 – Representação dos lugares de servidor, aplicação e banco de dados



Fonte: Battisti (2017)

Como um exemplo de desenvolvimento *Web Back-end*, suponha que você visite www.amazon.com usando seu navegador. O seu computador faz uma solicitação para o servidor *Amazon*, que executa um aplicativo para determinar o conteúdo da web para o servidor. O aplicativo cria uma página da *Web* que exibe livros que correspondem aos seus interesses e envia para o seu computador. Você vê um livro sobre trilhas de bicicleta em Nova Iorque e

clica para comprá-lo. Depois de inserir o seu cartão de crédito e os detalhes de envio, o aplicativo armazena as informações em um banco de dados no servidor para fácil *check-out* no futuro.

Para os desenvolvedores *Back-end*, uma parte importante do trabalho é escrever o código para o aplicativo e banco de dados para renderizar páginas da *Web* no navegador. Os empregadores estão interessados em habilidades adicionais como:

Scaling: Os desenvolvedores *Back-end* devem alterar e otimizar o código do aplicativo, os servidores e o banco de dados para responder aos aumentos no tráfego do site. Sem o planejamento certo, a menção do seu site em um programa de entrevistas da manhã ou no jornal pode resultar em uma mensagem de erro "site não disponível" em vez de milhares de novos clientes. A escala envolve equilibrar o custo de otimizar o site e deixar a configuração como está.

Análise: Todos os negócios *on-line*, grandes ou pequenos, têm indicadores de desempenho de sites importantes, como novas inscrições de usuários e retenção de usuários existentes. Os desenvolvedores *Back-end* podem implementar e rastrear essas métricas consultando informações do banco de dados do site.

Segurança: *Website* com um número substancial de usuários se tornam um alvo para todos os tipos de riscos de segurança. Os atacantes podem automatizar inscrições, em que perfis falsos postam *spam* que promove produtos não relacionados. Além disso, você pode receber uma enorme quantidade de tráfego em um curto período de tempo, chamado de ataque de negação de serviço, o que impede que clientes legítimos acessem seu site. Ou os atacantes podem tentar detectar fraquezas em seus servidores para obter acesso não autorizado a informações confidenciais, como endereços de *e-mail*, senhas e números de cartão de crédito. Em 2014, grandes brechas de dados foram descobertas em grandes corporações, incluindo Sony, Target e JP Morgan. A prevenção desses ataques repousa, em parte, com desenvolvedores de *Back-end*.

O desenvolvedor de *Back-end* faz parte da equipe de desenvolvimento que trabalha em colaboração com desenvolvedores de *Front-end* e gerentes de produção. Ao contrário dos desen-

volvedores *Front-end*, os desenvolvedores de *Back-end* não interagem com frequência com os designers porque o trabalho não é tão visual ou baseado na aparência do site.¹ [2]

Com base as atuações do desenvolvedor *Back-end*, os responsáveis pelo desenvolvimento do módulo de usuários, deviam controlar e realizar as funções que são destinadas aos usuários sendo elas: o cadastro, o *login*, recuperação de senha, acessar página de boas-vindas após *login*, editar informações de perfil e a disponibilização das funções do *Website*.

¹ Back-end development happens in the following three places: Server: The server is the computer hosting the coding files that include the website application and the database. When you visit www.google.com, for example, your web browser requests the web page from Google servers, which respond with a copy of the web page you see in your browser. Application: The application handles the content in web pages setn to users and the changes made to the database. Applications are written using programming languages like Ruby, Python and PHP, and only run on the server. Proficiency in one language is usually sufficient. Database: The database stores website and user data so it is available for future browsing sessions. The simplest database is an Excel spreadsheet, which is ill suited for web development. Databases such as Postgres and MongoDB are optimized for website use, usually only one of these databases is used per website. As an example of back-end web development, suppose that you visit www.amazon.com using your web browser. Your computer makes a request to the Amazon server, which runs an application to determine what web content to serve you. The application creates a web page that displays books matching your interests, and send it to your computer. You see a book on bike trails in New York and click to purchase it. After you enter your credit card and shipping details, the application stores the information in a database on the server for easy checkout in the future. For back-end developers, one major part of the job is writing code for the application and database to render web pages in the browser. Employers are interested in additional skills such as these: Scaling: Back-end developers must change and optimize application code, servers, and database to respond to increases in website traffic. Without the right planning, a mention of your website on a morning talk show or in the newspaper could result in a "website not available" error message instead of thousands of new customers. Scaling involves balancing the cost of optimizing the website with leaving the configuration as-is. Analytics: Every online business, whether large or small, has key website performance indicators, such as new user signups and retention of existing users. Back-end developers can implement and track these metrics by querying information from the website database. Security: Websites with a substantial number of users become a target for all types of security risks. Attackers may automate signups, in which fake profiles post spam that promotes unrelated products. Additionally, you may receive a massive amount of traffic in a short period of time, called a denial of service attack, which prevents legitimate customers from accessing your website. Or attackers might try to detect weaknesses in your servers to gain unauthorized access to sensitive information such as email addresses, passwords, and credit card numbers. In 2014, major data breaches were uncovered at large corporations including Sony, Target and JP Morgan. Prevention of these attacks rests, in part, with back-end developers. The back-end developer is a part of the product team and works closely with front-end developer and product managers. Unlike front-end developers, back-end developers do not interact frequently with designers because the job is not as visual or based on website appearance, (ABRAHAM; TAYLOR; SMITH, 2015, tradução nossa).

2.7 - Instrumentos utilizados

Os instrumentos utilizados na composição do TCC são os programas e linguagens:

2.7.1 - Notepad++

O Notepad++ é um arquivo que permite o livre uso para o programador, sendo facilmente moldável e intuitivo, servindo de editor de texto e editor de códigos fonte em várias linguagens de programação.

Ele é um *software* de uso livre e pode ser baixado nas plataformas *Windows* e *Linux* (via *Wine*). [2]

Figura 7 – Notepad++



Fonte: <https://notepad-plus-plus.org/> (2017)

2.7.2 - HTML

HTML é uma linguagem de programação utilizada para se criar páginas na *Web*, e podem ser interpretadas por quase todos os navegadores que existem.

O HTML é muito utilizado, pois suas funções são simples e objetivas, por isso, pode ser utilizado por programadores menos experientes, e ainda sim, gerar ótimos resultados.

Figura 8 – HTML 5

Fonte: <https://www.brandsoftheworld.com/logo/html5-1> (2017)

2.7.3 - CSS

O CSS é uma linguagem de folhas que definem o padrão da apresentação de um documento ou uma página na *Web* criada, por exemplo, em HTML. Sua principal utilidade é ajudar na separação do conteúdo, do formato do site.

Para utilizar o CSS o programador precisa criar a formatação em uma página separada do código para *Web*, ligado a ele por um link, que vai leva-lo a um portal que permite a modificação do seu arquivo.

Figura 9 – CSS 3

Fonte: <https://www.brandsoftheworld.com/logo/css3> (2017)

2.7.4 - JavaScript

O *JavaScript* é uma linguagem de programação do lado cliente, ou seja, é processada pelo próprio navegador. Com o *JavaScript* podemos criar efeitos especiais para nossas páginas na *Web*, além de podermos proporcionar uma maior interatividade com nossos usuários. O *JavaScript* é uma linguagem orientada a objetos, ou seja, ela trata todos os elementos da página como objetos distintos, facilitando a tarefa da programação. [3]

Figura 10 – JavaScript



Fonte: <https://www.brandsoftheworld.com/logo/javascript> (2017)

2.7.5 – PHP

PHP é uma linguagem que permite criar sites *WEB* dinâmicos, possibilitando uma interação com o usuário através de formulários, parâmetros da URL e *links*. A diferença de PHP com relação a linguagens semelhantes a *Javascript* é que o código PHP é executado no servidor, sendo enviado para o cliente apenas HTML puro. Desta maneira é possível interagir com bancos de dados e aplicações existentes no servidor, com a vantagem de não expor o código fonte para o cliente. [4]

Figura 11 – PHP

Fonte: <https://www.brandsoftheworld.com/logo/php-0> (2017)

2.7.6 - jQuery

jQuery é uma biblioteca em *JavaScript* projetada para simplificar as interações via script com o Modelo de Documento de Objeto (*Document Object Model* – DOM). Ele provê uma Interface de Programação de Aplicativos (*Application Programming Interface* – API) em plataforma cruzada que mascara as diferenças entre as diversas APIs de *runtime* (Tempo de execução) e provê um *framework* fácil de utilizar para o desenvolvimento de *JavaScript* não intrusivo.

Ou seja, o *jQuery* é uma biblioteca de *Javascript* super leve, muito fácil de usar e com uma curva de aprendizagem relativamente curta. Substitui as maiores e maiores complicadas tarefas do *Javascript* por funções mais diretas, rápidas e compatíveis com a generalidade dos browsers. [5]

Figura 12 – jQuery

Fonte: <https://www.brandsoftheworld.com/logo/jquery> (2017)

2.7.7 - Microsoft Word

O *Microsoft Word* é um editor de texto produzido pela *Microsoft*, suas funções variam das mais simples como escrever um texto comum, até produzir grandes documentos e formulários. Ele é um programa muito utilizado, pois é o mais comum editor/criador de textos para as plataformas de computador.

Figura 13 – *Microsoft Word*



Fonte: <https://www.brandsoftheworld.com/logo/microsoft-word-0> (2017)

2.8 – Página de Cadastro de Consumidor

Figura 14 – Página de cadastro do consumidor com *layout* estilizado

A imagem mostra a interface de usuário para o cadastro no Portal Reclame São João. No topo, há uma barra de navegação com o logo do portal e links para 'CADASTRAR', 'HOME' e 'LOGAR'. O título principal do formulário é 'Cadastre-se', seguido pelo subtítulo 'Para ter acesso a todos os recursos oferecidos por nós!'. O formulário contém sete campos de entrada:

- Nome Completo
- CPF
- Data de Nascimento
- Telefone Fixo (DDD) 0000-0000
- Celular (DDD) 00000-0000
- CEP
- Estado

Fonte: Autoria própria (2017)

Como pode-se visualizar a página de cadastro do consumidor é a página na qual permite o consumidor se cadastrar no Portal Reclame São João, ou seja, nesta página o usuário insere as informações dele que são relevantes e importante para o Portal, e posteriormente o consumidor consegue usufruir do site.

Entretanto, as informações são inseridas em um banco de dados do Portal, no qual cada dado é redirecionado para uma tabela referente aquela informação.

Para isso, primeiramente é realizado a conexão com o banco de dados, como pode-se ver abaixo.

Figura 15 - Criar a Conexão com o Banco de Dados

```

<?php
class Conexao
{
    private $servidor;
    private $usuario;
    private $senha;
    private $database;
    private $conexao;

    public function __construct()
    {
        $this->servidor = "localhost";
        $this->usuario = "root";
        $this->senha = "user";
        $this->database = "reclame_sao_joao";

        try
        {
            $this->conexao = new PDO("mysql:host=$this->servidor;dbname=$this->database", $
                $this->usuario, $this->senha);
            $this->conexao->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
        }
        catch (PDOException $e)
        {
            echo "Erro banco de dados: " . $e->getMessage() . "<br/>";
        }
    }

    public function getConexao()
    {
        return $this->conexao;
    }
}
?>

```

Fonte: Autoria própria (2017)

Figura 16 – Criar novo Usuário, Consumidor e Localização

```

< > usu_cadastroUsuario.php x
1 <!DOCTYPE html>
2 <?php
3 include 'controle/UsuarioDAO.php';
4 $dao = new UsuarioDAO();
5
6 include 'controle/LocalizacaoDAO.php';
7 $local = new LocalizacaoDAO();
8
9 include 'controle/ConsumidorDAO.php';
10 $consum = new ConsumidorDAO();
11 //$registro = $dao;
12

```

Fonte: Autoria própria (2017)

O código acima é responsável por criar um novo usuário, localização e consumidor, referente a cada tabela do banco de dados, para que posteriormente os dados de cada tabela sejam devidamente selecionados e inseridos

Figura 17 – Código para Estilizar o *Layout* da Página

```

120 <section id="contact">
121 <!--wrapper-->
122 <div id="contact-us" class="parallax">
123 <div class="container">
124 <div class="row">
125 <div class="heading text-center col-sm-6 col-sm-offset-2 wow fadeInUp" data-wow-duration="1000ms" data-wow-delay="300ms">
126 <h2>Cadastre-se</h2>
127 <p>Para ter acesso a todos os recursos oferecidos por nós!</p>
128 </div>
129 </div>
130
131 <div class="row">
132 <div class="col-md-8 col-md-offset-2">
133 <form action="php/mod91/cadastrarUsuario.php" method="POST" id="cadastro">
134 <div class="row wow fadeInUp" data-wow-duration="1000ms" data-wow-delay="300ms">
135 <div style="padding-left: 20px; padding-right: 20px">
136 <div class="form-group">
137 <input type="text" name="USU_NOME" class="form-control" placeholder="Nome Completo" required="required">
138 </div>
139 </div>
140 </div>
141
142 <div class="row wow fadeInUp" data-wow-duration="1000ms" data-wow-delay="300ms">
143 <div class="col-sm-6" style="padding-left: 20px; padding-right: 20px">
144 <div class="form-group">
145 <input type="text" name="CON_CPF" id="CON_CPF" class="form-control" placeholder="CPF" maxLength="14"
146 onkeypress="formatar('###.###.###-##', this);" required="required" oninvalid="setCustomValidity('Por
147 favor, digite um CPF válido');" onchange="try{setCustomValidity('')}catch(e){">
148 </div>
149 </div>
150 <div class="col-sm-6" style="padding-right: 20px; padding-left: 20px">
151 <div class="form-group">
152 <input type="date" id="CON_DATA_NASCIMENTO" name="CON_DATA_NASCIMENTO" class="form-control" placeholder="Data de
153 Nascimento" required="required">
154 </div>
155 </div>
156 </div>
157 <div class="row wow fadeInUp" data-wow-duration="1000ms" data-wow-delay="300ms">
158 <div class="col-sm-6" style="padding-left: 20px; padding-right: 20px">
159 <div class="form-group">
160 <input type="text" name="USU_TELEFONE" id="USU_TELEFONE" class="form-control" maxLength="14" placeholder="
161 Telefone fixo: (DDD) 0000-0000" required="required" oninvalid="setCustomValidity('Por favor, digite um
162 telefone válido');" onchange="try{setCustomValidity('')}catch(e){">
163 </div>

```

Fonte: Autoria própria (2017)

Figura 18 – Selecionar dados inseridos na página de cadastro

```

1  <?php
2  date_default_timezone_set('America/Sao_Paulo');
3
4  require_once '../..//controle/phpMailer/PHPMailerAutoload.php';
5
6  include '../..//modelo/Localizacao.php';
7  include '../..//controle/LocalizacaoDAO.php';
8
9  include '../..//modelo/Usuario.php';
10 include '../..//controle/UsuarioDAO.php';
11
12 include '../..//modelo/Consumidor.php';
13 include '../..//controle/ConsumidorDAO.php';
14
15
16
17     $localizacao = new Localizacao();
18     $localizacaoDAO = new LocalizacaoDAO();
19
20     $localizacao->setLOC_ESTADO($_POST['LOC_ESTADO']);
21     $localizacao->setLOC_RUA($_POST['LOC_RUA']);
22     $localizacao->setLOC_BAIRRO($_POST['LOC_BAIRRO']);
23     $localizacao->setLOC_CEP($_POST['LOC_CEP']);
24     $localizacao->setLOC_CIDADE($_POST['LOC_CIDADE']);
25     $idLocalizacao = $localizacaoDAO->inserirLocalizacao($localizacao);
26
27     $localizacao->setLOC_ID($idLocalizacao);
28
29
30
31     $usuario = new Usuario();
32     $usuarioDAO = new UsuarioDAO();
33
34     $usuario->setUSU_NOME($_POST['USU_NOME']);
35     $usuario->setUSU_EMAIL($_POST['USU_EMAIL']);
36     $usuario->setUSU_FOTO_PERFIL($_POST['USU_FOTO_PERFIL']);
37     $usuario->setUSU_TELEFONE($_POST['USU_TELEFONE']);
38     $usuario->setUSU_SENHA($_POST['USU_SENHA']);
39     $idUserio = $usuarioDAO->inserirUsuario($usuario);
40
41     $usuario->setUSU_ID($idUserio);
42
43
44
45

```

Fonte: Autoria própria (2017)

Acima estão os métodos *POST* responsáveis por selecionar cada dado que foi inserido na página de cadastro pelo consumidor. Primeiramente, é iniciado uma nova função e posteriormente, os dados são selecionados para que assim seja possível ser inseridos ao banco através de uma função. Por exemplo a função “*\$idUserio = \$usuarioDAO->inserirUsuario(\$usuario)*” referente aos dados da tabela “*usuarios*”;

A página de *Usuario.php* possui os métodos *get()* e *set()* que são responsáveis por selecionar as informações do método *POST* do formulário digitado pelo usuário e armazena via *set()* em uma variável privada.

As demais páginas (*Localizacao.php* e *Consumidor.php*), realizam as mesmas funções e portanto, selecionam as informações do método *POST* do formulário digitado pelo usuário e armazenam via *set()* em uma variável privada.

Figura 19 – Classe que possui os Métodos *get()* e *set()*

```
<?php
class Usuario
{
    private $USU_NOME;

    public function __construct()
    {

    }

    public function getUSU_NOME()
    {
        return $this->USU_NOME;
    }

    public function setUSU_NOME($USU_NOME)
    {
        $this->USU_NOME = $USU_NOME;
    }
}
?>
```

Fonte: Autoria própria (2017)

Posteriormente a função para inserir usuário é chamada na página *cadastrarUsuario.php* para assim inserir as informações no banco de dados, esta função se encontra dentro de *Usuario-DAO.php*.

Figura 20 – Página *UsuárioDAO* e função de inserir Usuário

```

1  <?php
2
3  require_once 'Conexao.php';
4  // require_once('.../modelo/Usuario.php');
5  require_once($_SERVER['DOCUMENT_ROOT'] . '\RECLAME_SAO_JOAO-INTEGRAR\modelo\Usuario.php');
6
7
8  class UsuarioDAO
9  {
10
11     private $conexao;
12     private $sql;
13     private $usuario;
14     private $resultado;
15     private $tabela;
16
17
18     public function __construct()
19     {
20         $conn = new Conexao();
21         $this->conexao = $conn->getConexao();
22         $this->tabela = "usuarios";
23     }
24
25     public function inserirUsuario($usuario)
26     {
27         $this->usuario = $usuario;
28         $this->sql="insert into $this->tabela (USU_FOTO_PERFIL, USU_DATA_CADASTRO, USU_EMAIL, USU_TELEFONE, USU_SENHA,
29             USU_NOME, USU_CADASTRO_VALIDADO) values (:USU_FOTO_PERFIL, :USU_DATA_CADASTRO, :USU_EMAIL, :USU_TELEFONE,
30             :USU_SENHA, :USU_NOME, 0)";
31         $this->resultado = $this->conexao->prepare($this->sql);
32
33         $this->resultado->bindValue(':USU_FOTO_PERFIL','adasd');
34         $this->resultado->bindValue(':USU_DATA_CADASTRO',date("Y-m-d",time()));
35         $this->resultado->bindValue(':USU_EMAIL',$this->usuario->getUSU_EMAIL());
36         $this->resultado->bindValue(':USU_TELEFONE',$this->usuario->getUSU_TELEFONE());
37         $this->resultado->bindValue(':USU_SENHA',$this->usuario->getUSU_SENHA());
38         $this->resultado->bindValue(':USU_NOME',$this->usuario->getUSU_NOME());
39
40         $this->resultado->execute();
41         return $this->conexao->lastInsertId();
42     }
43

```

Fonte: Autoria própria (2017)

E por fim, dentro da página *UsuarioDAO.php* será estabelecido a conexão por causa do código “*require_once ‘Conexao.php’;*” com o banco novamente e dentro da função “*public function inserirUsuario(\$usuario)*” ocorrerá a preparação para inserir as informações no banco de dados.

Logo, as funções designadas ao módulo de usuário foram desenvolvidas com base na metodologia descrita anteriormente, posteriormente, tendo sua implementação dada no Portal Reclame São João ao longo do ano de 2017.

Conclusões e Recomendações

Conforme apresentado em todos os capítulos anteriores, teve-se como finalidade apresentar a definição de desenvolvimento *Back-end* e a utilização do mesmo no projeto Reclame São João, mais precisamente referente ao módulo 1, no qual é responsável pelos usuários do *Web-Site*.

Para cumprir o objetivo, em primeiro momento foi apresentado a definição de desenvolvimento *Back-end*, comparando-o com o desenvolvimento *Front-end*, com o intuito de esclarecer e exemplificá-lo. Em seguida, foi apresentado a utilização deste desenvolvimento no módulo 1, ou seja, as regras de controle referentes aos usuários do *Website*. Além da exemplificação de uma parte do código referente a página de cadastro do consumidor e as ferramentas que foram utilizadas.

Com o objetivo do desenvolvimento *Back-end* referente as regras de controle do módulo de usuários, reportado no capítulo 2, até o presente momento os resultados obtiveram sucesso parciais, pois houveram algumas falhas em algumas regras que foram desenvolvidas, havendo também regras que obtiveram sucesso.

Contudo, os resultados obtidos foram relevantes, devido ao fato de que as regras desenvolvidas obtiveram em sua maioria sucesso, e com isso, permitiu que o projeto Reclame São João estivesse em crescente desenvolvimento no módulo 1, referente aos usuários do sistema *Web*.

As principais dificuldades desenvolvidas ao longo do projeto pelo grupo escolar, foram: comunicação entre os módulos e seus integrantes, divisão das responsabilidades e cooperação entre as funções de cada integrante. Além disso houveram demais dificuldade, sendo elas: desenvolvimento para ativação de cadastro de cada consumidor, recuperação de senha do consumidor, validação da troca de CPF (Certificado de Pessoa Física) e o desenvolvimento responsivo das páginas do *Website*.

Logo, com o desenvolvimento *Back-end* das regras de controle dos usuários referente ao projeto do *Website* Reclame São João, vê-se a necessidade da utilização de um *framework* para facilitar e visar produtividade de qualquer projeto que seja designado aos alunos da intuição de ensino.

Referências Bibliográficas

[1] VIANA, Daniel. O que é front-end e back-end?. Disponível em:

<<https://www.treinaweb.com.br/blog/o-que-e-front-end-e-back-end/>>. Acesso em: 11 set. 2017.

[2] ABRAHAM, Nikhil; TAYLOR, Kathleen; E. SMITH, Bud. **Getting a Coding Job For Dummies**. 1. ed. New York, Estados Unidos: For Dummies, 2015. 288 p. Disponível em:

<https://books.google.com.br/books?id=thgWCgAAQBAJ&pg=PA28&dq=Development+Website+and+code+and+Back-end&hl=pt-BR&sa=X&redir_esc=y#v=onepage&q=Development%20Website%20and%20code%20and%20Back-end&f=false>. Acesso em: 08 out. 2017.

[3] O QUE é o JavaScript. Disponível em: <[https://developer.mozilla.org/pt-](https://developer.mozilla.org/pt-PT/docs/Web/JavaScript/O_que_%C3%A9_o_JavaScript)

[PT/docs/Web/JavaScript/O_que_%C3%A9_o_JavaScript](https://developer.mozilla.org/pt-PT/docs/Web/JavaScript/O_que_%C3%A9_o_JavaScript)>. Acesso em: 09 out. 2017.

[4] O QUE é PHP. Disponível em:

<https://www.oficinadanet.com.br/artigo/659/o_que_e_php>. Acesso em: 09 out. 2017.

[5] O QUE é o jQuery? Como funciona o jQuery?. Disponível em:

<<https://www.redcodestudio.com/o-que-e-o-jquery/>>. Acesso em: 09 out. 2017.

CONCEITOS básicos sobre o jQuery. Disponível em: <<https://software.intel.com/pt-br/xdk/article/jquery-basics>>. Acesso em: 09 out. 2017.

O QUE é API?. Disponível em: <<https://www.tecmundo.com.br/programacao/1807-o-que-e-api-.htm>>. Acesso em: 09 out. 2017.

MODELO de Objeto de Documento (DOM). Disponível em:

<https://developer.mozilla.org/pt-BR/docs/DOM/Referencia_do_DOM>. Acesso em: 09 out. 2017.