

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

Campus São João da Boa Vista

Trabalho Final de Curso

4º ano – Curso Técnico em Informática

Prof. Breno Lisi Romano

**DESENVOLVIMENTO *BACK-END* PARA O CONTROLE DAS
REGRAS DOS ESTABELECIMENTOS REFERENTE AO
PROJETO RECLAME SÃO JOÃO.**

Aluno: Rafael Alves Camillo

Prontuário: 1420593

São João da Boa Vista – SP

2017

Resumo

A motivação pela qual houve a criação do portal de reclamações, denominado Reclame São João surgiu diante da necessidade de estabelecer uma comunicação entre os consumidores e estabelecimentos de São João da Boa vista, com isso, para realizar a criação deste site houve tanto a utilização do desenvolvimento front-end quanto *back-end*, sendo considerados fatores de desenvolvimento fundamentais para qualquer aplicação de um sistema *web*. O objetivo desse trabalho é apresentar todo o desenvolvimento *back-end* do módulo de Estabelecimentos do Projeto Reclame São João, apresentando as informações necessárias para a compreensão da relevância da utilização desse tipo de desenvolvimento em aplicações em geral, com isso está pesquisa possibilitará conhecer melhor o desenvolvimento *back-end*; analisar como funciona o fluxo de dados por trás das interfaces *web*, além de adaptar as linguagens de programação e seus códigos de uma forma compreensível para o leitor. Os conceitos utilizados abordam temas da área da informática, especificamente a de desenvolvimento de *websites* utilizando-se de algumas linguagens de programação como *PHP*, *JavaScript*, *HTML*, *CSS* e *JQuery*. Como procedimentos para realizar os objetivos propostos por este trabalho, foram utilizadas as aplicações *web* do módulo de Estabelecimentos do Projeto Reclame São João para a análise dos códigos, dando ênfase nos códigos utilizados para a construção de um *layout*, no qual o usuário consiga interagir. Os principais resultados obtidos com essa pesquisa foram, a partir da utilização desse tipo de desenvolvimento mostrar ao leitor todo o processo a ser realizado para desenvolver um *website*. Contudo as funcionalidades propostas obtiveram em sua maioria sucesso, além disso os objetivos propostos foram atingidos, e com isso, foram obtidos alguns resultados, tal como as linhas de códigos desenvolvidas, com um total de 11.554 linhas até o momento utilizando a metodologia apresentada nesta pesquisa.

Palavras-chave: *Back-end*; *PHP*; *Front-end*; *Website*; Reclame São João.

Lista de Figuras

Figura 1 - Exemplo de Código <i>PHP</i>	8
Figura 2 - Identificação dos Módulos para Desenvolvimento e Macro-Requisitos.....	11
Figura 3 – Criar um Novo Usuário e Chamar a Função “pesquisarPorId”	15
Figura 4 – Criar um Novo Estabelecimento e Chamar a Função “pesquisarPorId”	16
Figura 5 - Selecionar as Informações do Estabelecimento no Banco de Dados referente a tabela de Usuários	17
Figura 6 - Selecionar as Informações do Estabelecimento no Banco de Dados	17
Figura 7 - Criar a Conexão com o Banco de Dados.....	18
Figura 8 - Código para Estilizar o <i>Layout</i> da Página	19
Figura 9 - Inserir Informações nos métodos get() e set()	20
Figura 10 - Classe Usuário que possui os Métodos get() e set().....	21
Figura 11 - Classe Estabelecimento que possui os Métodos get() e set()	21
Figura 12 - Página Responsável por Alterar os Dados do Estabelecimento na tabela de Usuários no Banco de Dados	22
Figura 13 - Página Responsável por Alterar os Dados do Estabelecimento na tabela de Estabelecimentos no Banco de Dados	23
Figura 14 - Página com <i>Layout</i> Estilizado	24

Sumário

1	Introdução	5
1.1	Tema.....	5
1.2	Problema.....	5
1.3	Justificativa.....	6
1.4	Objetivos	7
1.4.1	Objetivo Geral	7
1.4.2	Objetivos Específicos.....	7
1.5	Fundamentação Teórica	7
1.6	Metodologia.....	9
1.6.1	Materiais.....	9
1.6.2	Procedimentos	9
1.7	Estrutura do Trabalho.....	9
2	Desenvolvimento	10
2.1	Referencial Teórico	12
2.1.1	Desenvolvimento <i>Back-end</i>	12
2.1.2	Desenvolvimento de <i>Websites</i> com <i>PHP</i>	14
2.2	Metodologia.....	15
3	Conclusões e Recomendações	25
4	Referências Bibliográficas	27

1 Introdução

1.1 Tema

O tema dessa pesquisa é sobre o Desenvolvimento *Back-end* para o controle das regras dos estabelecimentos referente ao Projeto Reclame São João, dando ênfase na importância da implementação da regra de negócio, ou seja, do desenvolvimento na parte de trás da interface.

A escolha do tema partiu do interesse de estudar como funciona o tipo de desenvolvimento em questão. Contudo, sempre houve afinidade pessoal relativa ao estudo do fluxo de dados de uma aplicação *web*.

O presente trabalho tem como propósito mostrar ao leitor a importância do desenvolvimento *Back-end* ou server-side, uma vez que a maioria da população não sabe que por trás de todas as interfaces gráficas desenvolvidas pelo ser humano até os dias de hoje sempre encontra-se códigos, denominados como linguagens de programação, do qual não são visíveis ao leitor, para dinamizar o site e dar fluidez aos dados para que a aplicação seja visualizada e compreendida pelo leitor.

Para mostrar a importância desta pesquisa utilizaremos como base, conforme citado acima, todo o processo de desenvolvimento do módulo de reclamações, no qual faz parte do projeto Reclame São João.

1.2 Problema

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 dificuldade em entrar em contato com os estabelecimentos na cidade de São João da Boa Vista situada no interior do Estado de São Paulo.

Para solucionar este problema, o Instituto Federal de Educação, Ciência e Tecnologia de São Paulo juntamente com a Prefeitura de São João da Boa Vista criaram um projeto para desenvolver um Portal de Reclamações, no qual tem como principal característica estreitar e facilitar a relação entre a população e os comerciantes sanjoanenses, para assim o consumidor poder avaliar e reclamar sobre possíveis problemas referente as empresas cadastradas no *web site* e as mesmas resolveriam estes problemas por meio deste, além disso os consumidores poderiam visualizar reclamações de outros usuários e como o estabelecimento esta avaliado.

O desenvolvimento deste portal denominado “Reclame São João” fora realizado pelos estudantes do 4º ano do Ensino Médio integrado em Informática do Instituto Federal por meio da disciplina Prática de Desenvolvimento de Sistemas (PDS), uma vez que os alunos inscritos nesta matéria têm como principal objetivo solucionar e desenvolver projetos, todos os anos, voltados para o benefício da sociedade.

Para potencializar o desempenho dos alunos houve o planejamento, no qual dividiu-se o projeto em cinco módulos, cada módulo representa uma parcela do projeto final, sendo eles: Usuários, Reclamações, Estabelecimentos, Relatórios e Administrativo.

No entanto, será utilizado o módulo de Estabelecimentos para dar embasamento a solução do problema inicial desta pesquisa: Como mostrar de forma compreensível aos leitores não cultos sobre o assunto a extrema influência que o desenvolvimento *Back-end* tem sobre a tecnologia, e consequentemente sobre a vida das pessoas.

1.3 Justificativa

A população do mundo Contemporâneo, especialmente a do século XXI encontra-se completamente situada no mundo tecnológico que, por sua vez, a cada ano ganha inúmeros integrantes. No ano de 2000, eram apenas 400 milhões de pessoas conectadas a *internet*, porém, atualmente o número de pessoas conectadas é de 3,2 bilhões segundo dados corroborados pela União Internacional das Telecomunicações (UIT). [1]

Por outro lado, apesar da tecnologia estar atualmente presente na vida da maioria dos brasileiros, por exemplo, em aparelhos eletrônicos como: a televisão, o computador, o telefone; a maioria dos cidadãos não compreendem como estes produtos funcionam, como foram feitos, quais linguagens de programação foram utilizadas.

No entanto, essa pesquisa é de extrema relevância, uma vez que passará um pouco de conhecimento desta área não somente para as pessoas que são cultas, mas especificamente para aquelas que não são, tentando passar as informações necessárias para o leitor compreender a dimensão da utilização do desenvolvimento *Back-end* e sua real importância tanto na vida dos profissionais da informática, quanto para os leigos afim de obter informações sobre este tema.

1.4 Objetivos

1.4.1 Objetivo Geral

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

Pretendemos também passar informações necessárias para a compreensão da relevância da utilização do desenvolvimento *Back-end* em aplicações em geral, através do portal de reclamações, Reclame São João.

Almejamos fazer com o que os leitores, após a leitura, saibam olhar de outra maneira todos os tipos de tecnologia que temos hoje. Sendo assim, esperando que os mesmos saiam satisfeitos com o aprendizado.

1.4.2 Objetivos Específicos

Então para cumprir o objetivo geral desta pesquisa será necessário realizar os seguintes objetivos específicos:

- Conhecer melhor o desenvolvimento *Back-end*;
- Analisar como funciona o fluxo de dados por trás das interfaces *web*;
- Criar uma “identidade” sobre esse tipo de desenvolvimento para com os leitores
- Adaptar as linguagens de programação e seus códigos de uma forma compreensível para o leitor.

1.5 Fundamentação Teórica

Os meios tradicionais de comunicação (TV e Computadores) já ocupam lugar significativo em nossas vidas há várias décadas. Como ferramentas de transmissão de informação e entretenimento e, sobretudo, como veículos de divulgação de produtos e serviços. Mas, no mundo contemporâneo, estamos também expostos a novos modelos de comunicação, tal como à *internet*.

A *internet* nada mais é do que rede de computadores dispersas por todo o planeta e que trocam mensagens entre si, também possibilitando aos usuários interagirem com as interfaces gráficas como, por exemplo, aplicações *web*.

As páginas *webs* são desenvolvidas em duas vertentes: a parte do *layout*, ou seja, aquilo que o usuário vê; e a parte do controle do fluxo de informações, ou seja, toda a parte de armazenamento e exibição de dados na tela, denominado desenvolvimento *Back-end*.

Nesta pesquisa, será utilizado alguns termos que por vezes não está no conhecimento de mundo de muitos leitores, pois não estão habituados com a área de programação de *websites*, no entanto, a seguir será abordado, de forma explicativa, esses termos para que com o decorrer da leitura o usuário esteja situado sobre o que é *back-end*, *front-end* e a linguagem de programação *PHP*.

A definição mais comum para os desenvolvimentos *front-end* e *back-end* é que o desenvolvimento *front-end* é o site, ou seja, a interface de que interage com o usuário. No entanto, o desenvolvimento *back-end* é um sistema responsável pela regra de negócios, *webservices* e *APIs* de uma aplicação, ou seja, a parte do lado do servidor que irá gerenciar o conteúdo. [2]

A linguagem de programação *PHP* é uma linguagem de *script open source* de uso geral, muito utilizada, e especialmente adequada para o desenvolvimento *web* e que pode ser embutida dentro do *HTML*. Em vez de muitos comandos para mostrar *HTML* as páginas *PHP* contêm *HTML* em código mesclado que realiza algo, como neste caso da figura 1, no qual mostra "Olá, eu sou um *script PHP*!".

O que distingue o *PHP* de algo como o *Javascript* no lado do cliente é que o código é executado no servidor, gerando o *HTML* que é então enviado para o navegador. O navegador recebe os resultados da execução desse *script*, mas não sabe qual era o código fonte. [3]

Figura 1 - Exemplo de Código *PHP*

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Exemplo</title>
  </head>
  <body>
    <?php echo "Olá, eu sou um script PHP!"; ?>
  </body>
</html>
```

Fonte: https://secure.php.net/manual/pt_BR/intro-what-is.php

1.6 Metodologia

1.6.1 Materiais

Os materiais que serão utilizados para a realização desse trabalho são as páginas de desenvolvimento do módulo de Estabelecimentos do portal Reclame São João, além de projetos similares ao mesmo e artigos científicos, no qual possibilitarão um maior entendimento sobre o tema delimitado.

Outrossim, utilizaremos algumas linguagens de programação, tal como *HTML*, *PHP* e *Java Script*, além de alguns programas como *MySQL* e *Notpad++*, estruturas de *Framework MVC*.

1.6.2 Procedimentos

Como procedimentos para a pesquisa, serão utilizadas as aplicações *web* do módulo de Estabelecimentos do Projeto Reclame São João para a análise dos códigos, dando ênfase nos códigos utilizados para a construção de um *layout*, no qual o usuário consiga interagir.

1.7 Estrutura do Trabalho

Este trabalho será constituído por uma introdução, no qual apresentará o tema da pesquisa, o problema que levou a realização desse *software*, a justificativa, o objetivo e específicos, a fundamentação teórica e a metodologia (materiais e procedimentos).

Além disso apresentaremos todo um desenvolvimento e por fim, os resultados finais do trabalho, ressaltando a importância do desenvolvimento *Back-end* para a contribuição com o mundo tecnológico que temos hoje.

2 Desenvolvimento

Conforme dito anteriormente o site Reclame São João surgiu diante da necessidade de comunicação entre os estabelecimentos e seus respectivos consumidores, pois havia o surgimento de muitas reclamações não solucionadas, uma vez que os consumidores tinham imensa dificuldade em entrar em contato com os estabelecimentos na cidade de São João da Boa Vista situada no interior do Estado de São Paulo.

Para sanar esta necessidade, o Instituto Federal de Educação, Ciência e Tecnologia de São Paulo juntamente com a Prefeitura de São João da Boa Vista criaram um projeto para desenvolver um Portal de Reclamações, no qual tem como principal característica estreitar e facilitar a relação entre a população e os comerciantes sanjoanenses, para assim o consumidor poder avaliar e reclamar sobre possíveis problemas referente as empresas cadastradas no *website* e as mesmas resolveriam estes problemas por meio deste, além disso os consumidores poderiam visualizar reclamações de outros usuários e como o estabelecimento está avaliado.

O desenvolvimento deste portal denominado “Reclame São João” fora realizado pelos estudantes do 4º ano do Ensino Médio integrado em Informática do Instituto Federal por meio da disciplina Prática de Desenvolvimento de Sistemas (PDS), uma vez que os alunos inscritos nesta matéria têm como principal objetivo solucionar e desenvolver projetos, todos os anos, voltados para o benefício da sociedade.

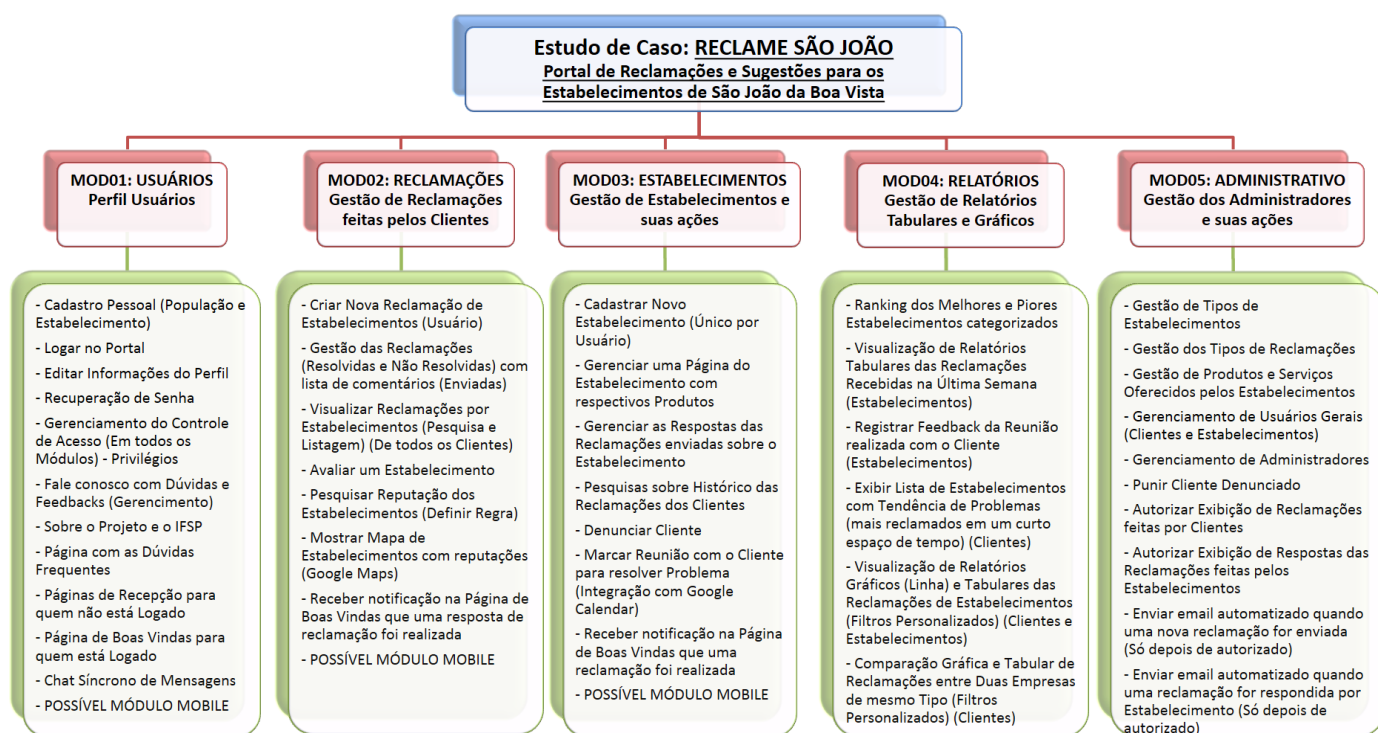
Para potencializar o desempenho dos alunos houve o planejamento, no qual dividiu-se o projeto em cinco módulos, cada módulo representa uma parcela do projeto final, sendo eles: Usuários, Reclamações, Estabelecimentos, Relatórios e Administrativo. Cada módulo tem por finalidade cumprir os requisitos que foram definidos no início do projeto.

No entanto, será utilizado o módulo de Estabelecimentos (módulo 3) para dar embasamento a solução do problema inicial desta pesquisa: Como mostrar de forma compreensível aos leitores não cultos sobre o assunto a extrema influência que o desenvolvimento tem sobre a tecnologia, e consequentemente sobre a vida das pessoas.

A figura 2 trata-se sobre a identificação dos módulos para o desenvolvimento e Macro Requisitos. Contudo, o módulo 3, responsável pela gestão de estabelecimentos e suas ações, tem como principal finalidade desenvolver tanto a parte *front-end* quanto *back-end* da parte do portal destinada as empresas, ou seja, desenvolver a parte visual do *site* (parte que interage com o usuário) e a parte *back-end* responsável por armazenar, exibir e editar informações.

Fora desenvolvido as páginas de cadastro de estabelecimento; gerenciamento das informações com seus respectivos produtos e serviços; gerenciamento das reclamações, possibilitando responder uma reclamação, caso necessário podendo ainda excluir ou editar a resposta. Além disso, a empresa pode agendar uma reunião com um cliente, uma vez que possui uma funcionalidade impar que o integra com o *Google Calendar* ou até mesmo denunciá-lo. A empresa pode ainda pesquisar seu histórico ou receber notificações quando uma reclamação for realizada.

Figura 2 - Identificação dos Módulos para Desenvolvimento e Macro-Requisitos



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

Portanto, é evidente a utilização dos desenvolvimentos *front-end* e *back-end*, em especial para esta pesquisa, enfatizaremos o desenvolvimento *back-end*, do qual se trata de tudo que não está visível no *site*, mas é importante para auxiliar o desenvolvimento *front-end* e vice-versa, um dependendo do outro de maneira equitativa.

Para o desenvolvimento do portal Reclame São João, os desenvolvedores adotaram a linguagem de programação *PHP*, pois é a linguagem que possuímos mais conhecimento, uma vez que aprendemos sobre a mesma nas disciplinas durante o decorrer do Curso de Informática integrado ao Ensino Médio. Contudo, para dar embasamento teórico para a pesquisa, a seguir, realizaremos um referencial teórico para explicar sob alguns pontos de vista, de autores distintos, o desenvolvimento *back-end*, além da linguagem de programação *PHP*.

2.1 Referencial Teórico

2.1.1 Desenvolvimento *Back-end*

Durante os próximos parágrafos inclusos neste tópico sobre desenvolvimento *Back-end* todo o conteúdo explicado tem como embasamento a editora de conteúdo do site *UpWork* Carey Wodehouse, que contribui com artigos e ebooks regulares sobre tendências no desenvolvimento *web*. Especializou-se em escrever assuntos da área da informática, complexos e técnicos, como programação, *APIs*, *internet*, *design* entre outros, de forma que as pessoas que não são da área possam compreender. [4]

O desenvolvimento *back-end* compreende três partes: o servidor, o banco de dados, todas as *APIs* e uma aplicação *web back-end*, *software* escrito via linguagem do lado do servidor.

O servidor pode ser definido, de forma simples e de fácil entendimento, como um computador poderoso que executa o *software back-end*, o banco de dados armazena os dados do seu site e o *software* se comunica entre os dois.

Por exemplo, se um usuário estiver atualizando um perfil em um site de rede, os *scripts* do lado do servidor reunirão as informações que o usuário digitar, o aplicativo irá processá-lo no servidor e, em seguida, interagir com o banco de dados para atualizar essas informações.

Os *scripts* (códigos) do lado do servidor são usados por desenvolvedores *web* para criar o *software back-end* de um site, ou seja, parte que é imperceptível ao usuário, mas que tornam a usabilidade e a funcionalidade de um site possíveis. Esses idiomas criam o canal de comunicação entre usuário, servidor e banco de dados.

Todos os dados que um usuário solicita no navegador (por exemplo, os campos em menus suspensos, fotos ou perfis de usuário) são entregues através de scripts do lado do servidor, que criam um canal entre o servidor e o usuário final, possibilitando ao usuário solicitar, editar e excluir informações no banco de dados. No navegador, os *scripts front-end* disponibilizam essa informação ao usuário.

As noções básicas do desenvolvimento *back-end* são que o código é executado do lado de um servidor, incorporado no código do *site*; projetado para sempre interagir com o armazenamento, como bancos de dados e processar essas informações do servidor para acessar o banco de dados, como uma linha direta do usuário para o banco de dados.

Além disso, facilita a transferência de dados do servidor para o navegador, trazendo vida às páginas no navegador, por exemplo, entrega um campo que um usuário solicita; as páginas ficam dinâmicas, uma vez que pode haver validação de usuários, permitindo ao mesmo salvar e recuperar dados, e navegar entre outras páginas. O *back-end* reproduz um grande papel na forma como um

banco de dados é construído desde o início e gerenciado posteriormente, isto pode ser considerado um exemplo de como os papéis se sobrepõem frequentemente em todos os aspectos do desenvolvimento. Resumidamente, o código de *back-end* adiciona utilidade a tudo o que o designer de *front-end* cria. [5]

Existem inúmeros idiomas populares do lado do servidor, porém o idioma do lado do servidor mais popular na *web* é a linguagem *php*. O *php* foi projetado para armazenar e editar informações no banco de dados. É geralmente incluído em bancos de dados escritos no idioma *SQL*. O *PHP* foi projetado estritamente para a *web* e continua sendo um dos idiomas mais utilizados. É fácil de instalar e implantar. Muitos sites são desenvolvidos em *PHP* como, por exemplo, alguns sites famosos e utilizados por milhares de pessoas ao redor do mundo, dentre eles, *WordPress*, *Wikipedia*, *Facebook*. ¹ [6]

¹ The back end comprises three parts: the server, your database, any APIs, and a back-end web application, software written via server-side languages. The server is a powerful computer that runs the back-end software, the database houses your site's data, and the software communicates between the two. For example, if a user is updating a profile on a networking site, the server-side scripts will gather the information the user enters, the application will process it on the server, then interact with the database to update that information there.

Server-side scripts are used by back-end web developers to build the back-end software of a website—the mechanics we don't see, but that make a site's usability and functionality possible. These languages create the communication channel between user, server, and database. Anything that isn't explicitly written into the text markup of a site is front-end or back-end software. Any data that a user requests in the browser (e.g., the fields in drop-down menus, photos, or user profiles) is delivered via server-side scripts, which create a channel between server and end user that requests, edits, and deletes things in the database. In the browser, front-end scripts make that information available to the user.

PHP: The most popular server-side language on the web, PHP is designed to pull and edit information in the database. It's most commonly bundled with databases written in the SQL language. PHP was designed strictly for the web and remains one of the most widely used languages around. It's easy to install and deploy, is staying competitive with lots of modern frameworks, and is the foundation for a number of content-management systems. PHP-powered sites: WordPress, Wikipedia, Facebook. (WODEHOUSE, 2017, tradução nossa).

2.1.2 Desenvolvimento de *Websites* com *PHP*

Todo o conteúdo explicado durante os próximos parágrafos inclusos neste tópico sobre desenvolvimento de *Websites* com *PHP* tem como embasamento no documento escrito por Juliano Niederauer, publicado na editora Novatec.

A principal diferença do *PHP* em relação às outras linguagens é a capacidade de interagir com o mundo *web*, transformando totalmente os websites que possuem páginas estáticas. Por exemplo, um *site web* precisa exibir milhares de notícias diferentes a todo momento, com isso, seria inviável fazer isso utilizando apenas *HTML*, uma vez que as páginas seriam estáticas, e a cada notícia nova a página deveria ser alterada manualmente, e logo após enviada ao servidor por *FTP* (*File Transfer Protocol*) para que as novas notícias fossem mostradas no site.

Para automatizar este processo, uma das formas é a utilização da linguagem de programação *PHP*, na qual precisaria criar um banco de dados onde ficariam armazenadas as notícias e criar uma página que mostrasse essas notícias, selecionando-as do banco de dados. Nesta página, por exemplo, haveria um único menu para fazer todas as cem páginas acessarem esse arquivo e mostrá-lo em sua parte da esquerda. Quando alguma alteração for necessária, basta alterar um único arquivo, e as cem páginas serão alteradas automaticamente, já que todas acessam o mesmo menu.

O *PHP* tem como principais características o dinamismo e a praticidade, possuindo inúmeras vantagens tais como a automatização de tarefas, economia de tempo e de mão de obra. Além disso, a linguagem é um *software* com código-fonte aberto, ou seja, o arquivo de instalação pode ser obtido gratuitamente no site <http://www.php.net>. O código-fonte do *PHP* e sua documentação detalhada também estão disponíveis no *site* oficial.

O *PHP* possui códigos que executa funções de muitos bancos de dados, dentre eles, temos *MySQL*, *PostgreSQL*, *Sybase*, *Oracle*, *SQL Server* e muitos outros, possuindo uma série de funções para aproveitar todos os recursos do banco de dados a ser utilizado.

Essa linguagem é embutida no *HTML*, uma página que contém programação *PHP* normalmente possui extensão “*.php*”. Porém, os códigos *HTML* e *PHP* estão misturados, pois começamos a escrever em *PHP*, de repente escrevemos um trecho em *HTML*, e assim por diante.

A linguagem *JavaScript* consiste em *scripts* que também são colocados nas páginas *web*, no meio do *HTML*, mas essa é uma programação que é executada no lado do cliente. Você abre seu *browser* (navegador) e acessa uma página que possui *JavaScript*. Além disso, a programação escrita em *JavaScript* pode ser vista e copiada por qualquer pessoa. Para isso, basta escolher Exibir > Código-fonte no menu do navegador.

O *PHP* é exatamente o contrário, pois é executado no lado do servidor (*back-end*). Quando você acessa uma página *PHP* por meio de seu navegador, todo o código *PHP* é executado no servidor, e os resultados são enviados para seu navegador. As linhas de programação *PHP* não podem ser vistas por ninguém, uma vez que são executadas no próprio servidor, sendo retornado apenas o resultado do código. [7]

2.2 Metodologia

Esta pesquisa é classificada como aplicada. Tem como método a forma descritiva e explicativa, utilizando-se tanto da visualização e seus respectivos códigos do portal Reclame São João quanto de artigos científicos correlatos a este trabalho disponibilizados na *internet*.

Através dos pontos de vista dos autores citados no referencial teórico, pode-se chegar ao resultado que todos eles consideram o desenvolvimento *back-end* como parte essencial na construção de um *site web*, atuando lado a lado com o desenvolvimento *front-end*, dando maior dinâmica, funcionalidade e funcionamento.

Para correlacionar o estudo teórico desses autores com o projeto realizado pelos alunos da disciplina de PDS, a seguir, será mostrado uma análise a partir do lado do servidor até o lado do cliente, pretendendo assim alcançar o resultado esperado.

Para demonstrar o funcionamento de um desenvolvimento *back-end* utilizaremos a página de visualizar e editar perfil referente ao estabelecimento no portal Reclame São João, possibilitando ao estabelecimento alterar suas informações se caso necessário. Inicialmente para aparecer as respectivas informações do estabelecimento que está acessando o portal no momento é necessário criar a conexão com o banco, conforme descrito nas figuras 3 e 4, abaixo.

Figura 3 – Criar um Novo Usuário e Chamar a Função “pesquisarPorId”

```
<?php
//Criar a conexão com a tabela Usuários
include 'controle/UsuariosDAO.php';
//Criar um novo Usuário
$USUARIOdao = new UsuarioDAO();
// Chamar a função para pesquisar as
// informações referentes a este estabelecimento
$registro_usuario = $USUARIOdao->pesquisarPorId($id);
?>
```

Fonte: Autoria própria (2017)

Figura 4 – Criar um Novo Estabelecimento e Chamar a Função “pesquisarPorId”

```
<?php
//Criar a conexão com a tabela Usuários
include 'controle/EstabelecimentosDAO.php';
//Criar um novo Usuário
$EstabelecimentosDAO = new EstabelecimentosDAO();
// Chamar a função para pesquisar as
// informações referentes a este estabelecimento
$registro_estabelecimento = $EstabelecimentosDAO->
pesquisarPorIdUsr($id);
?>
```

Fonte: Autoria própria (2017)

Depois de criar um novo usuário e um novo estabelecimento, chamamos as funções para obter todas essas informações do estabelecimento, uma vez que as tabelas de usuários e estabelecimentos estão interligas, pois, os mesmos, em alguns casos, possuem informações iguais que podem ser armazenadas na mesma tabela como é o caso do nome.

A função “pesquisarPorId(\$id)” está situada na pasta de controle do projeto, utilizando-se da estrutura *MVC* (estrutura de *framework* adotado para o desenvolvimento do portal) dentro da página *UsuarioDAO.php* e *EstabelecimentosDAO.php*, estas páginas são as responsáveis por criarem ou estabelecerem um canal de comunicação entre usuário, servidor e banco de dados, com isso, podemos relacionar este fato com o que citamos sobre desenvolvimento *back-end* no referencial teórico, segundo Wodehouse, 2017.

Dentro das páginas *UsuarioDAO.php* e *EstabelecimentosDAO.php* será estabelecido a conexão com o banco de dados novamente por causa do código “*require_once* ‘Conexao.php’;” dentro da função *pesquisarPorId* ocorrerá a preparação para realizar selecionar as informações no banco de dados de acordo com o que fora solicitado. Por exemplo, na página de gerenciamento de perfil o usuário tem que ter acesso a todas as informações que ele inseriu quando realizou o cadastro no portal, algumas informações relevantes são o nome, o CNPJ e o telefone da empresa.

Além disso, as páginas *UsuarioDAO.php* e *EstabelecimentosDAO.php* possuem extensão “.php”, representadas pelas figuras 5 e 6 respectivamente, pois são páginas que contém programação *PHP*, conforme observado por Niederauer, 2011 no referencial teórico desta pesquisa. Depois de ter selecionado as informações do estabelecimento em questão, a partir de códigos *HTML* e *PHP* as informações serão mostradas ao usuário, uma vez que o *PHP* é uma linguagem embutida em *HTML*, os códigos ficarão misturados, o que também foi dito por Niederauer.

Figura 5 - Selecionar as Informações do Estabelecimento no Banco de Dados referente a tabela de Usuários

```
<?php
    require_once 'Conexao.php';

    class UsuarioDAO
    {
        private $conexao;
        private $sql;
        private $usuario;
        private $resultado;
        private $tabela;

        public function __construct()
        {
            $conn = new Conexao();
            $this->conexao = $conn->getConexao();
            $this->tabela = "usuarios";
        }

        public function pesquisarPorId($id)
        {
            $this->sql="select * from $this->tabela where USU_ID=$id";
            $this->resultado= $this->conexao->prepare($this->sql);
            $this->resultado->execute();
            return $this->resultado->fetchAll();
        }
    }
?>
```

Fonte: Autoria própria (2017)

Figura 6 - Selecionar as Informações do Estabelecimento no Banco de Dados

```
<?php
    require_once 'Conexao.php';

    class EstabelecimentosDAO
    {
        private $conexao;
        private $sql;
        private $post;
        private $resultado;
        private $tabela;

        public function __construct()
        {
            $conn = new Conexao();
            $this->conexao = $conn->getConexao();
            $this->tabela = "ESTABELECEMENTOS";
        }

        public function pesquisarPorId($id)
        {
            $this->sql = "select * from $this->tabela where EST_ID=$id";
            $this->resultado = $this->conexao->prepare($this->sql);
            $this->resultado->execute();
            return $this->resultado->fetchAll();
        }
    }
?>
```

Fonte: Autoria própria (2017)

Na figura 7, na página *Conexao.php* que possui uma classe com este nome é onde acontece a conexão com o banco de dados do portal Reclame São João, passando como parâmetros o servidor, o usuário, a senha e o nome do banco de dados.

Ao tentar conectar-se com a *database* (banco de dados), se caso houver algum erro uma mensagem será mostrada, caso contrário, a conexão fora realizada com sucesso.

Figura 7 - 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
            ; charset=utf8", $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)

Na figura 8, contém códigos *HTML*, o qual é responsável pelo *layout* e estilização da página e também contém a parte em *php*, que será responsável por mostrar as informações ao estabelecimento, caso o mesmo queira editá-las a tag *<form>* *</form>* enviará os dados alterados para a página responsável por armazenar as informações antes de armazená-la no banco de dados, através do método *POST*, neste método a *URI* não é retornável ao cliente, o que torna o método *POST* mais seguro, pois não expõe os dados enviados no navegador.

Figura 8 - Código para Estilizar o Layout da Página

```
<body>
  <div class="container">
    <!-- INICIO - INFORMAÇÕES PESSOAIS -->
    <div class="row" id="todas-reclamacoes">
      <div class="col-sm-12">
        <div class="panel panel-default">
          <div class="panel-heading"><span class="fa fa-id-card-o" aria-hidden="true"></span> Informações Pessoais <
            span onclick='informacoes_pessoais()' id="habilitar_informacoes_pessoais" class="fa fa-pencil-square-o
            pull-right" aria-hidden="true"></span></div>
          <div class="panel-body">
            <form method="POST" id="form_informacoes_pessoais" action="php/mod03/editarInformacoesPessoais.php"
            enctype="multipart/form-data">
              <div class="col-sm-12">
                <div class="panel panel-default" style="border-left: thick double #64594f; border-right:
                thick double #64594f;">
                  <div class="panel-body">
                    <div class="row">
                      <div class="col-md-12 col-lg-12">
                        <table class="table table-user-information">
                          <tbody>
                            <?php foreach($registro_usuario as $linha_usuario) { ?>
                              <td>
                                <div class="form-group">
                                  <label>Nome da Empresa</label>
                                  <input type="text" name="USU_NOME" id="USU_NOME" class="
                                  form-control" disabled value="<?php echo $linha_usuario['
                                  USU_NOME'];?>">
                                </div>
                              </td>
                              <?php } ?>
                            <?php foreach($registro_estabelecimento as $linha_estabelecimento) { ?>
                              <td>
                                <div class="form-group">
                                  <label>Nome da Empresa</label>
                                  <input type="text" name="EST_NOME_FANTASIA" id="
                                  EST_NOME_FANTASIA" class="form-control" disabled value="
                                  <?php echo $linha_estabelecimento['EST_NOME_FANTASIA'];?>">
                                </div>
                              </td>
                              <?php } ?>
                            </tbody>
                          </table>
                        </div>
                      </div>
                    </div>
                  </div>
                </div>
              </div>
            </form>
          </div>
        </div>
      </div>
    </div>
    <div class="panel-group">
      <div class="panel panel-default">
        <div class="panel-heading">
          <ul class="nav nav-tabs nav-justified">
            <li>
              <a href="javascript:{}" onclick="document.getElementById('
              form_informacoes_pessoais').submit();" id="
              botao_atualizar_informacoes_pessoais" style="display:none"><span class="fa
              fa-tachometer" aria-hidden="true"></span> Editar Informações Pessoais da
              Empresa</a>
            </li>
          </ul>
        </div>
      </div>
    </div>
  </div>
  <!-- FIM - INFORMAÇÕES PESSOAIS -->
</div>
</body>
```

Fonte: Autoria própria (2017)

A página `editarInformacoesPessoais.php` irá pegar os dados também por *post* e inserir nos métodos *get()* e *set()* nas páginas `Usuario.php` e `Estabelecimentos.php` dentro da pasta modelo do projeto, que são parte de um conceito muito utilizado chamado encapsulamento. O intuito do

encapsulamento é tornar certas variáveis acessíveis através de métodos específicos, proporcionando controle e segurança na manipulação destas variáveis, conforme a figura 9.

Figura 9 - Inserir Informações nos métodos *get()* e *set()*

```
<?php
include '../..//modelo/Usuario.php';

include '../..//controle/UsuarioDAO.php';

$usuario= new Usuario();

$usuarioDAO = new UsuarioDAO();

$usuario->setUSU_NOME($_POST['USU_NOME']);

$usuario->setUSU_ID($_POST['USU_ID']);

$usuarioDAO->editarInformacoesPessoais($usuario);

include '../..//modelo/Estabelecimentos.php';

include '../..//controle/EstabelecimentosDAO.php';

$estabelecimentos = new Estabelecimentos();

$estabelecimentosDAO = new EstabelecimentosDAO();

$estabelecimentos->setEST_NOME_FANTASIA($_POST['EST_NOME_FANTASIA']);

$estabelecimentos->setEST_ID($_POST['EST_ID']);

$estabelecimentosDAO->editarInformacoesPessoais($estabelecimentos);

header("location:../..//est_perfil.php");

?>
```

Fonte: Autoria própria (2017)

As páginas de Usuario.php e Estabelecimentos.php representadas pelas figuras 10 e 11, possuem os métodos *get()* e *set()* como dito anteriormente pegando as informações do método *POST* do formulário digitado pelo usuário e armazenando via *set()* em uma variável privada. Posteriormente através do método *get()* é possível pegar essas informações inseridas pelo usuário, para assim, armazená-las no banco de dados.

Figura 10 - Classe Usuário 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)

Figura 11 - Classe Estabelecimento que possui os Métodos *get()* e *set()*

```
<?php
class Estabelecimentos
{
    private $EST_NOME_FANTASIA;

    function __construct()
    {

    }

    function getEST_NOME_FANTASIA()
    {
        return $this->EST_NOME_FANTASIA;
    }

    function setEST_NOME_FANTASIA($EST_NOME_FANTASIA)
    {
        $this->EST_NOME_FANTASIA = $EST_NOME_FANTASIA;
    }
}
?>
```

Fonte: Autoria própria (2017)

Posteriormente a função para editar as informações pessoais do estabelecimento é chamada na página *editarInformacoesPessoais.php* para assim editar as informações no banco de dados, esta função encontra-se tanto dentro de *UsuarioDAO.php* quanto em *EstabelecimentosDAO.php*.

Nas figuras 12 e 13, estas funções são responsáveis por alterar tudo aquilo que o usuário modificou na página que interage com ele (lado do usuário – *front-end*) no banco de dados (lado do servidor - *back-end*).

Figura 12 - Página Responsável por Alterar os Dados do Estabelecimento na tabela de Usuários no Banco de Dados

```
<?php
require_once 'Conexao.php';

class UsuarioDAO
{
    private $conexao;
    private $sql;
    private $usuario;
    private $resultado;
    private $tabela;

    public function __construct()
    {
        $conn = new Conexao();
        $this->conexao = $conn->getConexao();
        $this->tabela = "usuarios";
    }

    public function editarInformacoesPessoais($post)
    {
        $this->post = $post;
        $this->sql = "update $this->tabela set USU_NOME =:USU_NOME, USU_TELEFONE
            =:USU_TELEFONE where USU_ID =:USU_ID";

        $this->resultado = $this->conexao->prepare($this->sql);

        $this->resultado->bindValue(':USU_NOME', $this->post->getUSU_NOME());
        $this->resultado->bindValue(':USU_TELEFONE', $this->post->getUSU_TELEFONE());
        $this->resultado->bindValue(':USU_ID', $this->post->getUSU_ID());

        $this->resultado->execute();

        return $this->resultado;
    }
}
```

Fonte: Autoria própria (2017)

Figura 13 - Página Responsável por Alterar os Dados do Estabelecimento na tabela de Estabelecimentos no Banco de Dados

```
<?php
    require_once 'Conexao.php';

    class EstabelecimentosDAO
    {
        private $conexao;
        private $sql;
        private $post;
        private $resultado;
        private $tabela;

        public function __construct()
        {
            $conn = new Conexao();
            $this->conexao = $conn->getConexao();
            $this->tabela = "ESTABELECIMENTOS";
        }

        public function editarInformacoesPessoais($post)
        {
            $this->post = $post;
            $this->sql = "update $this->tabela set EST_NOME_FANTASIA=:EST_NOME_FANTASIA,
                where EST_ID =:EST_ID";

            $this->resultado = $this->conexao->prepare($this->sql);

            $this->resultado->bindValue(':EST_NOME_FANTASIA', $this->post->
                getEST_NOME_FANTASIA());
            $this->resultado->bindValue(':EST_ID', $this->post->getEST_ID());

            $this->resultado->execute();

            return $this->resultado;
        }
    }
?>
```

Fonte: Autoria própria (2017)

Depois que as informações forem atualizadas no banco, a página em que o usuário estava é atualizada e imediatamente as informações que foram alteradas pelo estabelecimento são mostradas a ele. No entanto, a única parte que o usuário vê é a do *layout*, da parte que interage com o mesmo, com isso, é notório a dificuldade e os inúmeros fluxos a serem realizados para que o estabelecimento edite uma informação em seu perfil.

A parte que interage com o usuário está demonstrada na figura 14, na qual, mostra as informações do estabelecimento, além de possibilitar o mesmo a editar / alterar suas informações.

Isto só é possível por causa do desenvolvimento back-end, mas é claro que o desenvolvimento front-end tem uma grande parcela para isso tornar-se possível.

Figura 14 - Página com Layout Estilizado

The screenshot displays the 'Gerenciamento do Perfil do Estabelecimento' (Establishment Profile Management) page. The header includes the 'RECLAME SÃO JOÃO' logo and the user 'SUL BRASIL QUIMICA LTDA' with a notification bell showing 2 alerts. The main content area is titled 'Informações Pessoais' and features a 'Habilitar Edição' toggle set to 'Off'. The form contains the following fields:

Informações Pessoais	
Nome da Empresa	Nome Fantasia
Rafael Chaveirinho	SUL BRASIL QUIMICA LTDA
Responsável	CNPJ
Rafael Alves Camillo	44.404.200/8500-00
Telefone	Facebook
(19) 3642-1989	www.facebook.com
Site da Empresa	Email
https://www.youtube.com	rafael.a.camillo15@gmail.com
Tipo de Estabelecimento	Senha

Fonte: Autoria própria (2017)

Todas as metodologias descritas ao longo desta pesquisa foram aplicadas para o desenvolvimento do *layout* e a parte do lado do servidor, *front-end* e *back-end* respectivamente, do projeto Reclame São João. O desenvolvimento de todo o portal fora com base nessa metodologia, o módulo de estabelecimentos obteve 10 funcionalidades a serem desenvolvidas.

3 Conclusões e Recomendações

O objetivo geral deste trabalho era apresentar todo o desenvolvimento *Back-end* do módulo de Estabelecimentos do Projeto Reclame São João, apresentando as informações necessárias para a compreensão da relevância da utilização do desenvolvimento *Back-end* em aplicações em geral, através do portal de reclamações.

Para cumprir o objetivo geral desta pesquisa foi necessário realizar os seguintes objetivos específicos: conhecer melhor o desenvolvimento *Back-end*; analisar como funciona o fluxo de dados por trás das interfaces *web*; criar uma “identidade” sobre esse tipo de desenvolvimento para com os leitores; adaptar as linguagens de programação e seus códigos de uma forma compreensível para o leitor.

Como reportado no capítulo 2, o desenvolvimento de oito funcionalidades foram finalizadas até o momento, restando duas a serem desenvolvidas possuindo apenas o rascunho da ideia inicial., uma vez que o prazo final para o desenvolvimento das funcionalidades do projeto Reclame São João ser posterior a data de entrega desta pesquisa, por esse motivo considera-se que o objetivo foi parcialmente cumprido com sucesso.

As principais dificuldades encontradas foram a gestão do tempo, tanto para o desenvolvimento do projeto uma vez que não foram desenvolvidos todos os casos de uso de acordo com os requisitos definidos por conta de que o prazo final para o desenvolvimento dos casos de uso ser posterior a data de entrega desta pesquisa quanto para a realização deste trabalho, além disso realizar essa pesquisa de maneira que um leitor não culto no assunto pudesse compreender todo o conteúdo apresentado é outra dificuldade a ser considerada.

Como procedimentos para realizar os objetivos propostos por este trabalho, foram utilizadas as aplicações web do módulo de Estabelecimentos do Projeto Reclame São João para a análise dos códigos, dando ênfase nos códigos utilizados para a construção de um *layout*, no qual o usuário consiga interagir.

Contudo, os resultados obtidos foram relevantes, pois as funcionalidades propostas obtiveram em sua maioria sucesso, além disso os objetivos propostos foram atingidos, e com isso, foram obtidos alguns resultados, tal como as linhas de códigos desenvolvidas, com um total de 11.554 linhas até o momento utilizando a metodologia apresentada nesta pesquisa.

Como trabalho futuro, é esperado finalizar o desenvolvimento *Back-end* das regras de controle dos estabelecimentos referente ao projeto Reclame São João das funcionalidades que ainda faltam, apresentando os resultados obtidos. Mas, é importante salientar a necessidade da utilização de um *framework* para facilitar o desenvolvimento, visando a produtividade de qualquer projeto

designado aos alunos dos próximos anos da disciplina de PDS do Instituto Federal de Educação, Ciência e Tecnologia de São João da Boa Vista. Além disso, espera-se que o *website* Reclame São João seja implantado para facilitar a comunicação entre os consumidores e estabelecimentos.

4 Referências Bibliográficas

[1] GLOBO, G1. Mundo tem 3,2 bilhões de pessoas conectadas à internet, diz UIT: População Online. Disponível em: <<http://g1.globo.com/tecnologia/noticia/2015/05/mundo-tem-32-bilhoes-de-pessoas-conectadas-internet-diz-uit.html>>. Acesso em: 20 out. 2017.

[2] LAMIM, Jonathan. Afinal, o que é Frontend e o que é Backend?. Disponível em: <<https://www.oficinadanet.com.br/post/13541-afinal-o-que-e-frontend-e-o-que-e-backend->>. Acesso em: 22 out. 2017.

[3] PHP.NET. O que é o PHP?. Disponível em: <https://secure.php.net/manual/pt_BR/intro-whatis.php>. Acesso em: 23 out. 2017.

[4] WODEHOUSE, Carey. Author Details. Disponível em: <<https://www.upwork.com/hiring/author/careywodehouse/>>. Acesso em: 24 out. 2017.

[5] WODEHOUSE, Carey. Server-Side Scripting: Back-End Web Development Technology. Disponível em: <<https://www.upwork.com/hiring/development/server-side-scripting-back-end-web-development-technology/>>. Acesso em: 23 out. 2017.

[6] WODEHOUSE, Carey. Back-End Technology: The Role of the Back-End Web Developer. Disponível em: <<https://www.upwork.com/hiring/development/back-end-web-developer/>>. Acesso em: 24 out. 2017.

[7] NIEDERAUER, Juliano. Desenvolvendo Websites com PHP. 2ª Edição. Disponível em: <<http://www.martinsfontespaulista.com.br/anexos/produtos/capitulos/650595.pdf>>. Acesso em: 26 out. 2017.