

# **DESENVOLVIMENTO DA APLICAÇÃO MY.IMG COM HTML, CSS, JAVASCRIPT E PHP**

**Francisco Gabriel de Araújo Chagas<sup>1</sup>**

**Gerson Bhrener Silva Ribeiro de Sá<sup>1</sup>**

**Gianmarco Salvador Daitx<sup>1</sup>**

**Elzimar José de Carvalho Neto<sup>2</sup>**

## **RESUMO**

Esse trabalho tem o objetivo de abordar conceitos e ferramentas do desenvolvimento de uma aplicação web com persistência em banco de dados relacional. A web não é mais a de dez anos atrás, pois, o avanço das tecnologias tem lhe expandido de uma forma nunca vista. É possível desenvolver grandes aplicações web por meio de várias linguagens, como o Java ou C# por exemplo, mas este estudo foca nas linguagens mais conhecidas deste “mundo”, que são: HTML5, CSS3, JavaScript e PHP. Para demonstrar algumas dessas tecnologias, foi desenvolvida a aplicação web My.Img. A My.Img é capaz de cadastrar um usuário e armazenar as imagens por ele escolhidas no banco de dados MySQL, além disso, é possível recuperá-las após o envio, exibindo-as na página inicial do usuário. Em adição às linguagens de programação citadas anteriormente, algumas ferramentas são abordadas, tais como: o Apache Xampp, a biblioteca JQuery e a classe PDO do PHP.

## **1. INTRODUÇÃO**

Hoje em dia, a Internet está repleta de inúmeros sistemas, sejam: websites, redes sociais, aplicativos, jogos etc. “Existem vários tipos de sistemas de software, desde os simples sistemas embutidos até os sistemas de informações complexos, de alcance mundial.” (SOMMERVILLE, 2011, p. 2).

Pressman (2011, p. 34-35) define que hoje há sete grandes categorias de software de acordo com o campo de aplicação. Software de sistema que consiste em um conjunto de programas desenvolvidos para atender a outros programas (software de rede, drivers etc). Software de aplicação: programas que atendem a uma necessidade específica do negócio (ERP, controle de processos de fabricação etc); Software científico/engenharia: são aplicações bem específicas para uma determinada área (simulação, cálculos quânticos etc); Software embutido, são aqueles que residem num produto ou sistema e são usados para implementar e controlar características (Painel de controle de um veículo, painel de controle do microondas etc); Software para linha de produtos: projeta para prover capacidade específica de utilização por muitos clientes diferentes (editor de texto, planilhas eletrônicas etc); Aplicações para a Web: são chamadas de WebApps e em sua forma mais simples são nada mais que um conjunto de arquivos de hipertexto interconectadas (marketplaces, streaming etc); e por fim, os Softwares

<sup>1</sup> Francisco Gabriel de Araújo Chagas, Gerson Bhrener Silva Ribeiro de Sá e Gianmarco Salvador Daitx.

<sup>2</sup> Elzimar José de Carvalho Neto

de inteligência artificial: que faz uso de algoritmos para solucionar problemas complexos (jogos, redes neurais etc).

Quanto às aplicações para a Web, elas nem sempre foram ricas em recursos como vemos nos dias atuais, segundo Sommerville (2011, p. 8):

No início, a Internet era basicamente um armazenamento de informações acessível universalmente e tinha pouco efeito nos sistemas de software. Esses sistemas executavam em computadores locais e eram acessíveis apenas dentro da organização. Por volta do ano 2000, a Internet começou a evoluir, e mais e mais recursos passaram a ser adicionados aos navegadores. Isso significa que sistemas Web poderiam ser desenvolvidos e que, em vez de ter uma interface de usuário específica, poderiam ser acessados por um navegador. Isso levou ao desenvolvimento de uma enorme quantidade de novos produtos de software que ofereciam serviços inovadores e que eram acessados através da Internet. Esses produtos eram frequentemente sustentados pela propaganda exibida na tela do usuário e não exigiam pagamento direto.

Alguns dos sistemas espalhados pela Web passam por um longo processo de software até serem disponibilizados para os usuários finais. Ainda de acordo com Sommerville (2011, p. 18), um processo de software depende do fim deste software não podendo ser seguido da mesma maneira para um outro software, e que esse processo é um grupo de atividades que levam a produção deste; existem vários processos de software, porém, eles devem incluir quatro atividades, tais elas: Especificação de software; Projeto e implementação de software; Validação de software; e Evolução de software. Por este se tratar do estudo das linguagens mais conhecidas da Web, essas etapas serão abstraídas, dando enfoque em algumas funcionalidades dessas linguagens.

Sabendo como os processos de software se dividem, basta entender o que são as aplicações Web, objeto deste estudo. De acordo com Noleto (2020), autor do blog betrybe:

A aplicação web diz respeito a uma solução que é executada diretamente no browser (ou navegador), não sendo preciso realizar uma instalação na máquina do usuário. Pode-se, também, utilizar como definição “tudo aquilo que é processado em um servidor terceiro”. As plataformas de e-commerce e as redes sociais são alguns dos exemplos que se enquadram nesse perfil.

Essas aplicações dependem de outros sistemas para funcionarem, como por exemplo, os protocolos de comunicação. Ou seja, a “[...] aplicação deve permitir que as pessoas usuárias consigam fazer uma solicitação e receber algo em resposta.” (NOLETO, 2020). Por exemplo, “[...] se a pessoa pede para abrir uma foto, é preciso que isso seja devolvido à ela, e não a abertura de uma página aleatória.” (NOLETO, 2020). “Nesse sentido, o servidor tem por função

receber a solicitação do público e devolver uma resposta para a aplicação. A resposta pode ser a abertura de uma nova página, imagens, documentos, entre outros.” (NOLETO, 2020)

O autor ainda aponta que é importante não confundir os sistemas Web com outros sistemas:

É importante deixarmos claro que aplicações web e sistemas tradicionais não são a mesma coisa! Os sistemas tradicionais apresentam o mesmo resultado para todos, ou seja, não há adaptações ou navegações personalizadas de acordo com cada visitante. Agora, quando você abre o Facebook, a sua página não é a mesma encontrada pelos seus amigos — as permissões, a usabilidade e a visualização são diferentes —, algo característico de aplicações web. (NOLETO, 2020)

“Além disso, não é preciso qualquer tipo de instalação nas máquinas quando se trata de aplicações web. O mesmo não ocorre quando falamos em sistemas tradicionais, que precisam ser adaptados para cada sistema operacional e dependem de instalação e atualizações manuais.” (NOLETO, 2020)

Por fim, é importante que se tenha o entendimento da diferença entre site e aplicação. Para Noleto (2020), a aplicação Web é interativa, em outras palavras, ela irá direcionar uma interação de um usuário com um item desta. Por exemplo, a aplicação Trello fornece um serviço de gerenciamento de tarefas, a aplicação Google Drive, dispõe de armazenamento na nuvem para que seus usuários possam persistir os seus documentos e arquivos. Ambas aplicações tem inúmeros recursos que permitem a interação com eles.

Um site geralmente é mais estático. Sendo apenas um conjunto de páginas as quais informam o usuário sobre alguma coisa, ou seja, acaba não complementando muito ao usuário, é como se o site fosse uma vitrine que informa sobre o que os produtos fazem e a aplicação é o produto em si que trará uma utilidade prática para aquele usuário. Mesmo assim, não é possível generalizar os sites, pois alguns fornecem interação, mas em pouca proporção, o usuário podendo fazer poucas ações. Ainda a casos onde sites dependem de aplicações Web que tornam possível a interatividade do usuário, como persistência em banco de dados relacionais, por exemplo.

Além das questões anteriormente citadas, uma aplicação nasce de uma necessidade. O My.Img surgiu por meio da necessidade de criar uma base de dados de imagens em um contexto pessoal, onde os usuários possam alimentá-la não só apenas com imagens, mas outras informações sobre aquelas imagens, tais elas: um título e uma descrição; e quando ele a enviar para o servidor, a sua página pessoal a recupera, sendo que ele pode a qualquer momento excluí-la caso desejar. O título da imagem ajudará o usuário a identificar o que aquela imagem representa. A descrição o irá lembrá-lo do que aquela imagem representa.

Ademais, fica em aberto a seguinte questão:

Será possível desenvolver uma aplicação que persista imagens com as linguagens e ferramentas que serão percorridas?

## 2. FUNDAMENTAÇÃO TEÓRICA

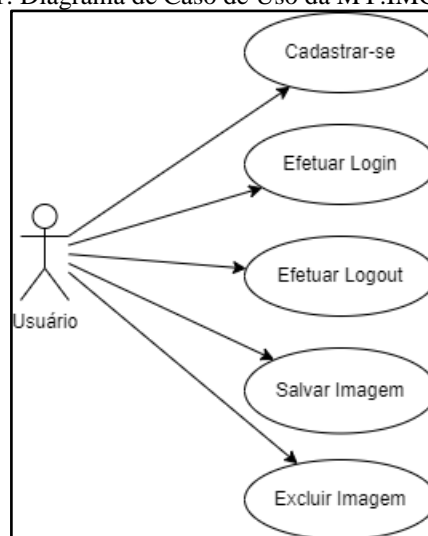
### 2.1. Caso de Uso, Estrutura do My.Img e Ferramentas

Antes de iniciar a construção de um projeto é necessário ter definido o problema a ser solucionado e fazer um estudo de caso em torno dele, isto irá facilitar a visualização do problema. Este estudo abordará apenas uma versão abstraída desta etapa.

A linguagem de modelagem unificada ou UML é sem dúvidas a linguagem mais conhecida e utilizada para a construção de representações visuais de projetos, ela é acessível a qualquer projeto na área de software. “A UML não é uma linguagem de programação, mas existem ferramentas que podem ser usadas para gerar código em várias linguagens por meio de diagramas UML.” (LUCIDCHART, 2022).

Existem vários tipos de diagramas UML: diagramas de classes; diagramas de componentes; diagramas de objetos; diagramas de atividades etc. Este trabalho irá abordar somente o diagrama de caso de uso. O diagrama de caso de uso em UML segundo a Lucidchart (2022), tem como objetivo: “[...] demonstrar as diferentes maneiras que o usuário pode interagir com um sistema.”. Conforme pode ser visto abaixo:

Imagem 1: Diagrama de Caso de Uso da MY.IMG

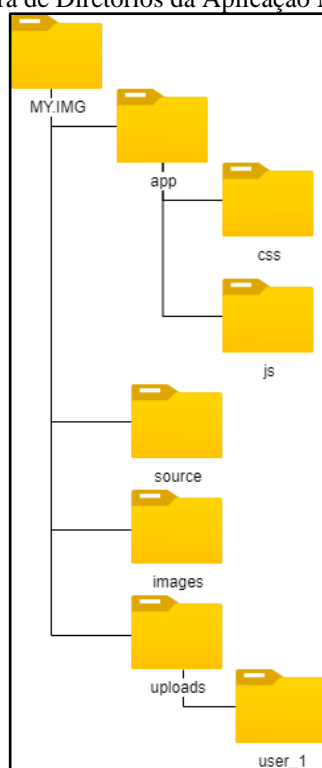


Fonte: O autor.

De acordo com a **Imagem 1**, o usuário pode realizar cinco ações, que são: cadastrar no sistema, efetuar o login; uma vez logado, é possível efetuar o logout; dentro da página do usuário é possível salvar as imagens escolhidas e ainda deletar àquelas que não tenha mais interesse.

A organização dos componentes (pastas, arquivos etc.) do projeto é outro fator importante. De acordo com site Holder Tech (2020), uma estrutura de pasta bem organizada pode trazer muitos benefícios tais como: localizar um arquivo mais rápido, evitar a perda de documentos, deixar o backup mais eficiente, fácil movimentação dos arquivos etc.

Imagem 2: Estrutura de Diretórios da Aplicação MY.IMG



Fonte: O autor.

Na **Imagem 2**, a aplicação MY.IMG está estruturada em um diretório raiz nomeado com o próprio nome da aplicação, logo após as pastas **app** com outras duas **css** e **js**, **source**, **images** e **uploads** com uma pasta **user\_1**. No diretório **app** serão inseridos os arquivos referente ao usuário e suas subpastas controlarão o estilo das páginas e algumas funcionalidades. No diretório **source** conterà os arquivos fonte responsáveis por autenticação, controle de formulários e comunicação com a base de dados. No diretório **images** serão adicionadas as imagens definidas do site, como logotipo, fundo etc. E por fim, no diretório **uploads**, quando os usuários se cadastrarem na aplicação, será criado um diretório separado para cada usuário

com o seu nome, dentro dele estarão as imagens que este usuário subiu para o servidor. Vale ressaltar que essas pastas estão organizadas e dispostas no diretório público da aplicação, este método foi utilizado apenas para fins didáticos.

Existem na internet inúmeras ferramentas para a criação de sistemas web. É possível encontrar ferramentas online que criam sistemas sem a necessidade de instalação na máquina do desenvolvedor, todavia, esta forma traz muitas desvantagens como: a limitação de recursos e a impossibilidade de mudanças críticas. Além destas, há outras, em grande maioria, que fornecem ao desenvolvedor um ambiente onde ele possa desenvolver manualmente o seu código fonte.

Ferramentas como o famoso adobe dreamweaver fizeram muito sucesso anos atrás, mas a grande maioria dos desenvolvedores sempre apreciaram um método mais direto, como é o caso de usar um editor de código fonte e o próprio navegador para visualizar os resultados. E por falar em editores, existem uma gama deles na Internet, cada um com suas especificidades. Exemplos: Notepad++, Visual Studio Code, Sublime Text, Eclipse, Atom e etc.

O estudo presente focou no uso do Visual Studio Code, que segundo Hanashiro (2021), ele é capaz de fazer coisas incríveis para ajudar o fluxo de trabalho, além de possuir uma loja de extensões, as quais podemos adicionar novas funcionalidades ao editor; e ainda ser customizável, ou seja, é possível mudar os elementos visuais ou o comportamento deles conforme a necessidade do usuário.

## **2.2. Front-End com HTML, CSS, Bootstrap e Javascript**

Quando se fala em front-end, é possível imaginar a parte frontal de algo, que de acordo com TOTVS (2021), uma famosa empresa de SGE (Sistema de Gestão Empresarial), “[...] está muito relacionado com a interface gráfica do projeto. Ou seja, é onde se desenvolve a aplicação com a qual o usuário irá interagir diretamente, seja em softwares, sites, aplicativos, etc.”. Souto (2019) agrega afirmando que: “Quem trabalha com Front End é responsável por desenvolver por meio do código uma interface gráfica e, normalmente, com as tecnologias base da Web: HTML, CSS e JavaScript.”

De acordo com a página inicial da seção HTML do site w3schools (2022), HTML é uma linguagem padrão de marcação para criação de páginas da Web; ela descreve a estrutura de página Web; consiste em uma série de elementos; os elementos dizem ao seu navegador como deve mostrar o conteúdo; os elementos rotulam partes do conteúdo como “isto é um cabeçalho”, “isto é um parágrafo”, “isto é um link”, etc.

O famoso site de tecnologia Devmedia (2012), complementa a definição afirmando que, “O **HTML5** é a nova versão do **HTML4** e um dos seus principais objetivos é facilitar a manipulação dos elementos, possibilitando o desenvolvedor modificar as características dos objetos de forma não intrusiva, fazendo com que isso fique transparente para o usuário final.”.

Imagem 3: Formulário de Login

```
29 <form action="Source/auth.php" method="POST">
30 <h4>Entre com suas credenciais para acessar o seu banco de imagens.</h4>
31 <div class="input-username">
32 <i class="fa-solid fa-user"></i>
33 <input id="username-field" type="text" required placeholder="Usuário" name="login">
34 <div id="login-empty-error"></div>
35 </div>
36 <div class="input-username">
37 <i class="fa-solid fa-key"></i>
38 <input id="password-field" type="password" required placeholder="Senha" name="password">
39 <div id="pass-empty-error"></div>
40 </div>
41 <button type="submit" class="login-button" title="Entrar">ENTRAR</button>
42 <div id="reg-user">
43 <label>Não tem conta?</label>
44 <button class="register-button"><a href="app/register_user.php">CADASTRE-SE</a></button>
45 </div>
46 <div id="autherror">
47 <?php
48 <div id="autherror">
49 <div id="autherror">
50 <div id="autherror">
51 <div id="autherror">
52 <div id="autherror">
53 <div id="autherror">
```

Fonte: O autor.

Na a **Imagem 3** é possível perceber um trecho de código HTML referente ao elemento “form”. Por meio deste elemento é possível capturar as entradas do usuário na página de login e fazer uma validação a fim de verificar se os dados inseridos correspondem àqueles armazenados no banco de dados, caso estejam, o usuário é redirecionado para a página inicial da aplicação (Home.php). O método utilizado para enviar os dados foi o “POST” que tem a capacidade de esconder os dados recolhidos no corpo da requisição, ou seja, os dados digitados pelo usuário não serão expostos na barra de endereço do navegador.

Os atributos “class” e “id” são referenciados no arquivo de estilização (CSS) para definir seu formato, cor e posicionamento. Os elementos “i” definem ícones para os inputs. Nos campos input, o atributo “required placeholder” impede que o usuário envie seus dados se caso estes estiverem vazios. O botão “register-button” direciona o usuário para uma tela de cadastro caso ele não tenha uma conta. E por fim, um trecho de código PHP verifica se o retorno da requisição está vazio, ou seja, se o usuário não tiver cadastro; neste caso, será mostrada uma mensagem de usuário ou senha inválida.

Para complementar as funcionalidades do HTML5, como posicionamento de ítems, seu dimensionamento, entre outras, surgiu o CSS (Cascading Style Sheets) ou Folhas de Estilo em Cascata, que segundo a documentação da Mozilla (2021), é uma linguagem de estilo usada para descrever a apresentação de um documento escrito em HTML ou em XML. “Utilizando o CSS é possível alterar a cor do texto e do fundo, fonte e espaçamento entre parágrafos. Também pode criar tabelas, usar variações de layouts, ajustar imagens para suas respectivas telas e assim por diante.” (HOSTINGER, 2022).

Ainda de acordo com a (hostinger), o CSS é uma ferramenta muito poderosa para o desenvolvedor, uma vez que por meio dele é possível criar diversas funcionalidades ao invés de usar o javascript ou outra linguagem. Em outra perspectiva: “O HTML cria a ordem dos elementos e a forma básica, como um esqueleto, e o CSS é o que deixa o website visualmente atraente, como o corpo e as roupas que vestimos.” (NOLETO, 2022).

O CSS usa uma sintaxe simples baseada em inglês com um conjunto de regras que o governam. Como mencionamos anteriormente, o HTML nunca teve a intenção de usar elementos de estilo, apenas a marcação da página. Foi criado para descrever apenas o conteúdo. Por exemplo: <p>Este é um parágrafo.</p>. O seletor aponta para o elemento HTML que você deseja estilizar. O bloco de declaração contém uma ou mais declarações separadas por ponto e vírgula. (NOLETO, 2022)

“Cada declaração inclui um nome de propriedade CSS e um valor, separados por dois pontos. Uma declaração CSS sempre termina com um ponto-e-vírgula e os blocos de declaração são cercados por chaves.”(HOSTINGER, 2022). E o autor continua:

[...] a sintaxe do CSS é muito simples de aprender. O seletor seleciona quais elementos em html receberão a propriedade. Pode ser *p* (parágrafo) ou o *body* (corpo da sua página). Já a propriedade pode ser a cor ou algo mais específico como cor do fundo (*background*). E por último o valor, que determina o valor da propriedade.” (HOSTINGER, 2022).

Imagem 4: Arquivo de Estilização (CSS)

```
26  /* PÁGINA DE LOGIN -----
27
28  .page-login-container {
29      padding-top: 4%;
30      display: flex;
31      flex-direction: column;
32      justify-content: flex-start;
33      align-items: center;
34      gap: 25px;
35  }
36
```

Fonte: O autor.



A **Imagem 4** representa uma pequena parte do arquivo de estilização da aplicação My.Img. na linha 26 há um comentário que documenta a funcionalidade abaixo dele. Na linha 28 é feita a referência ao elemento HTML de classe “page-login-container”. Entre as chaves há algumas funcionalidades que serão aplicadas durante o carregamento das páginas da aplicação. Na linha 29 é definido o espaçamento acima do elemento referenciado, que neste caso é de 4% do tamanho total da área visível da página Web. Na linha 30 a propriedade display setado como flex, assim os elementos serão dispostos em linha, um ao lado do outro. Na linha 31, o flex-direction com valor “column” define que os itens devem ser organizados em forma de colunas iniciando de cima para baixo. Na linha 32 a propriedade justify-content com valor flex-start, alinha os itens a partir do início do eixo principal. Na linha 33, o align-items com o valor center, centraliza os itens no eixo transversal. E por fim, a linha 34 que define o valor do gap em 25 pixels, ajusta o espaçamento entre os elementos internos com esse valor.

Hoje em dia não dá mais para falar sobre layout de sites e não citar o bootstrap como uma das melhores, ou se não, a melhor opção para organizá-los e torná-los responsivos, isto é, configurá-los para que eles se adaptem a todas as dimensões de tela. O bootstrap foi desenvolvido inicialmente para o Twitter, mas por conta da sua usabilidade, se tornou um grande sucesso por toda a Internet.

Na página do bootstrap, ele é conceituado como: “um poderoso kit de ferramentas de front-end repletas de recursos [...]” (GETBOOTSTRAP, 2022, tradução nossa). O site de cursos Alura complementa esse conceito afirmando que ele “[...] é um framework front-end que fornece estruturas de CSS para a criação de sites e aplicações responsivas de forma rápida e simples. Além disso, pode lidar com sites de desktop e páginas de dispositivos móveis da mesma forma.”(LIMA, 2021).

O bootstrap funciona por meio de componentes, e estes são referenciados nas tags HTML pelo atributo “class”. Na prática, basta apenas utilizar este atributo e inserir dentro dele os componentes desejados. Por exemplo: class = “fw-bold”, significa que estou definindo a característica negrito para um dado texto. Dessa forma, não se faz necessário o uso de documentos de estilização (CSS), uma vez que os componentes do bootstrap fornecem o suficiente para estruturar todo o sistema.

```

48 <div class="modal-body">
49   <form action="../Source/img_send.php" method="post" enctype="multipart/form-data">
50     <div class="mb-3">
51       <label for="imgTitleForm" class="form-label">Título</label>
52       <input type="text" class="form-control" name="img_title" id="imgTitleForm"
53         required placeholder="Dê um título a sua imagem">
54     </div>

```

Fonte: O autor.

O trecho de código na **Imagem 5** ilustra parte da funcionalidade responsável por enviar imagens para o banco de dados, chamada de Modal Popup. O Modal Popup é um pequeno menu escondido na interface que pode ser chamado por meio de um botão. Neste caso, o botão adicionar no menu home da aplicação abre um menu que se sobrepõe à tela dele e pode ser fechado a qualquer momento, bastando que o usuário clique no “x”. Este menu não é nada mais do que um formulário de envio que tem um **título**, uma **descrição** e o arquivo de **imagem**. Quanto ao código acima, é possível perceber algumas das classes presentes no bootstrap as quais configuram um formulário. Na linha 50 um elemento div de classe “mb-3” define o espaçamento de baixo da margem. Na próxima linha, 51, a classe “form-label” configura esse elemento como um rótulo de formulário. Nas linhas 52 e 53 há um elemento input, onde o usuário irá inserir o título da sua imagem; aqui, a classe “form-control” irá carregar as configurações de estilo definidas pelo bootstrap para este campo. Para que haja a conexão entre os elementos label e input, a propriedade for é definida no label e chamada no input por meio da propriedade "id".

Para se desenvolver um bom front-end, não basta usar apenas as tecnologias já abordadas, afinal, o que adianta uma interface ser muito atraente, mas não ter interatividade? O que adianta ter um botão, mas ao clicá-lo nada acontecer? Para responder a essas e outras perguntas que envolvem eventos dentro das aplicações web, foi desenvolvida uma linguagem poderosa capaz de lidar com eles e ir muito além.

O Javascript é segundo a documentação da Mozilla (2021),

Uma linguagem leve, interpretada e baseada em objetos com funções de primeira classe, mais conhecida como a linguagem de script para páginas Web, mas usada também em vários outros ambientes sem browser, tais como node.js, Apache CouchDB e Adobe Acrobat. O JavaScript é uma linguagem baseada em protótipos, multi-paradigma e dinâmica, suportando estilos de orientação a objetos, imperativos e declarativos (como por exemplo a programação funcional).

A documentação da Mozilla continua:

O padrão JavaScript é ECMAScript. Desde 2012, todos os navegadores modernos possuem suporte total ao ECMAScript 5.1. Navegadores mais antigos suportam pelo menos ECMAScript 3. Em 17 de Junho de 2015, a ECMA International publicou a

sexta versão do ECMAScript, que é oficialmente chamado de ECMAScript 2015, e foi inicialmente conhecido como ECMAScript 6 ou ES6. Desde então, as especificações do ECMAScript são lançadas anualmente. Essa documentação faz referência à última versão de referência, que atualmente é a ECMAScript 2018.

Graças a sua grande expansividade no que diz respeito às suas funcionalidades dentro das aplicações Web, o javascript ganhou algumas bibliotecas que buscam expandi-lo ainda mais. Um exemplo bem famoso é o JQuery, uma biblioteca que busca simplificar o código, principalmente quando o assunto são eventos.

No site oficial do JQuery (2022), ele é conceituado como uma biblioteca JavaScript rápida, pequena e rica em recursos. O seu uso torna as coisas como manipulação de documentos HTML, manipulação de eventos, animação e Ajax muito mais simples e fácil de usar, que funciona em vários navegadores.

Assim como o bootstrap, o JQuery precisa que um arquivo seja requerido pela página no elemento HEAD; esse arquivo pode ser buscado por meio de um link online ou o arquivo local. Ambos funcionam da mesma forma, sendo o link mais vantajoso por não ocupar espaço físico no site.

O JQuery possui uma forma bem simples de acessar os elementos HTML, através de seletores. Abaixo temos alguns exemplos deles:

- \$("\*") : seleciona todos os elementos da página;
- \$(this) : seleciona o elemento atual que será operado dentro de uma função;
- \$("p") : seleciona todas as instâncias da tag <p>;
- \$(".example"): seleciona todos os elementos com a classe example aplicada;
- \$("#example"): seleciona uma única instância do único id example;
- \$("[type='text']"): seleciona qualquer elemento com text aplicado ao atributo type;
- \$("p:first-of-type"): seleciona o primeiro <p>.

Imagem 6: Função para visualizar o Modal Popup

```
109 <script>
110     $("#button#add-button").click(function() {
111         |     $("#simpleModal").css("display", "block");
112         | });
113     $("#span#closeBtn").click(function() {
114         |     $("#simpleModal").css("display", "none");
115         | });
116 </script>
```

Fonte: O autor.

Conforme a **Imagem 6**, o código ilustrado resume dezenas de linhas que possivelmente seriam usadas para codificar o mesmo comportamento em javascript no seu formato convencional. Esse código tem a função de abrir e fechar o Modal Popup, citado anteriormente.

Para inserirmos um código javascript dentro de uma página HTML, basta abrir o elemento script e fechá-lo, conforme as linhas 109 e 116. Em vista da melhor compreensão da imagem, o Modal Popup será novamente citado, na sua folha de estilização (CSS) tem a propriedade display com o valor none por padrão, ou seja, quando o usuário entrar na sua página pessoal, este menu estará inativo. Entretanto, caso o usuário clique no botão “adicionar” da página, cujo id é “add-button”, a função definida entre as linhas 110 e 112 será disparada, alterando o valor de display para “block”, o que significa que será exibido o Modal Popup na tela. As linhas 113 a 115 fazem exatamente o contrário. Perceba que ambos os eventos disparados na linha 111 e 114 são quase idênticos, mudando apenas o valor de display, ou seja, será utilizado o mesmo elemento cujo id é “simpleModal”. Esse elemento agrupa todos os campos presentes no formulário de envio das imagens.

### **2.3. Banco de Dados e Servidor WEB com o Apache Xampp**

Banco de dados é uma grande necessidade nos dias de hoje, visto que a demanda por informações cresce exponencialmente seja entre as pessoas, mas principalmente das grandes corporações. Em um mundo movido por softwares, dados é um alimento vital para que estes desempenhem bem o seu papel e forneçam o suporte adequado no cotidiano.

“Bancos de dados e sistemas de banco de dados são um componente essencial da vida na sociedade moderna; a maioria de nós encontra diariamente diversas atividades que envolvem alguma interação com um banco de dados.”(ELMASRI; NAVATHE, 2011, p. 2). Mas como se conceitua um banco de dados? Segundo o mesmo autor:

Um banco de dados é uma coleção de dados relacionados. Com dados, queremos dizer fatos conhecidos que podem ser registrados e possuem significado implícito. Por exemplo, considere os nomes, números de telefone e endereços das pessoas que você conhece. Você pode ter registrado esses dados em uma agenda ou, talvez, os tenha armazenado em um disco rígido, usando um computador pessoal e um software como Microsoft Access ou Excel. Essa coleção de dados relacionados, com um significado implícito, é um banco de dados.(ELMASRI; NAVATHE, 2011, p. 2)

Quanto aos tipos de banco de dados eles podem ser: relacionais ou não relacionais. Para Alexandruk (2018), a estrutura básica dos bancos de dados relacionais é formada por tabelas, que também são conhecidas como relações. Uma tabela é uma estrutura bidimensional formada

por colunas e por linhas. Apresentam, portanto, uma estrutura similar às planilhas eletrônicas como as do Microsoft Excel. Os bancos de dados não relacionais ao invés de usar tabelas para agrupar seus dados, usam outras linguagens de programação e construtos para consultar os dados.

Dentro do escopo dos banco de dados, uma ferramenta se faz extremamente importante. O SGBD (Sistema Gerenciador de Banco de Dados) é um “[...] software utilizado para gerir Bases de Dados, permitindo criar base de dados, modificar Base de dados, eliminar bases de dados, inserir dados na Base de Dados e Eliminar dados da base de dados.”(DEV MEDIA, 2014).

Atualmente existem muitos SGBDs, tais como: PostgreSQL, Oracle, MySQL, MariaDB, SQL-Server entre muitos outros. Mesmo que eles tenham funções parecidas, existem características que ampliam suas funcionalidades, como é o caso do Oracle que têm sua própria linguagem (PL/SQL) que dá suporte ao SQL já existente através de scripts que rodam do lado do servidor.

Não basta entender como os bancos de dados funcionam, é preciso compreender como se manipula os dados. Assim, se faz necessário o conhecimento da linguagem padrão dos SGBDs.

SQL é uma linguagem padrão para trabalhar com bancos de dados relacionais. Ela é uma linguagem declarativa e que não necessita de profundos conhecimentos de programação para que alguém possa começar a escrever queries, as consultas e [sic]pedidos, que trazem resultados de acordo com o que você está buscando. SQL significa Standard Query Language, literalmente a linguagem padrão para realizar queries.(SILVEIRA, 2019)

As principais instruções desta linguagem são SELECT, INSERT, UPDATE e DELETE. “O comando SELECT é utilizado para realização de consultas em tabelas ou visões. Através das consultas podemos obter todos os dados de uma tabela ou apenas aqueles que desejamos utilizando, para isso, filtros.” (ALEXANDRUK, 2018, p. 76). “[...] INSERT é usado para acrescentar uma única tupla a uma relação. Temos de especificar o nome da relação e uma lista de valores para a tupla.” (ELMASRI; NAVATHE, 2011, p. 71). Uma tupla é um outro nome para as linhas da tabela. “O comando UPDATE é usado para modificar valores de atributo de uma ou mais tuplas selecionadas.” (ELMASRI; NAVATHE, 2011, p. 72). “A exclusão de linhas de uma tabela é realizada através do comando DELETE. Podemos excluir uma ou mais linhas de uma tabela utilizando este comando.” (ALEXANDRUK, 2018, p. 80).

Vale ressaltar que para criar um banco de dados existem muitos outros conceitos que devem ser estudados, como os modelos (conceitual, lógico e físico), os tipos de

relacionamentos, cardinalidade etc. Todavia esses outros conceitos fogem do escopo do trabalho, por isso não serão abordados.

Imagem 7: Estrutura do Banco de Dados do My.Img

```
1  create database myimgdatabase;
2
3  create table Users (
4      user_id int not null AUTO_INCREMENT,
5      user_login varchar(50) not null,
6      user_password varchar(32) not null,
7      user_email varchar(255) not null,
8      primary key(user_id)
9  );
10
11 create table Images(
12     image_id int not null AUTO_INCREMENT,
13     image_name varchar(255) not null,
14     image_title varchar(50) not null,
15     image_desc text,
16     user_id int not null,
17     constraint FK_user_id foreign key(user_id)
18     references Users(user_id)
19 );
```

Fonte: O autor.

Conforme ilustrado na **Imagem 7**, algumas instruções foram usadas para montar a base de dados, vale a pena frisar que os dados ainda não estão sendo inseridos. Na linha 1, a instrução `create database` adicionará uma nova base de dados dentro do SGBD chamada de “myimgdatabase”. Entre as linhas 3 e 9 será criada a tabela “User” onde serão inseridos os dados dos usuários da aplicação. Dentro do seu corpo há algumas colunas. A coluna “user\_id” está setada como int, não nulo e será gerado automaticamente após um novo dado ser inserido na tabela. Essa coluna ajudará a identificar os usuários por meio de um número. As linhas 5, 6 e 7 configuram dados do tipo varchar, que são um conjunto de caracteres com tamanho definido, sendo o “user\_login” com no máximo 50 caracteres, “user\_password” com no máximo 32 e o user\_email com até 255 caracteres. Por fim, o “user\_id” foi escolhido para ser o campo que representará o usuário, este não pode ser repetido entres os registros. Quando feitas consultas avançadas, este campo será usado como um recurso para referenciar um usuário, assim, será possível recuperar qualquer outro dado que esteja na faixa de registros deste.

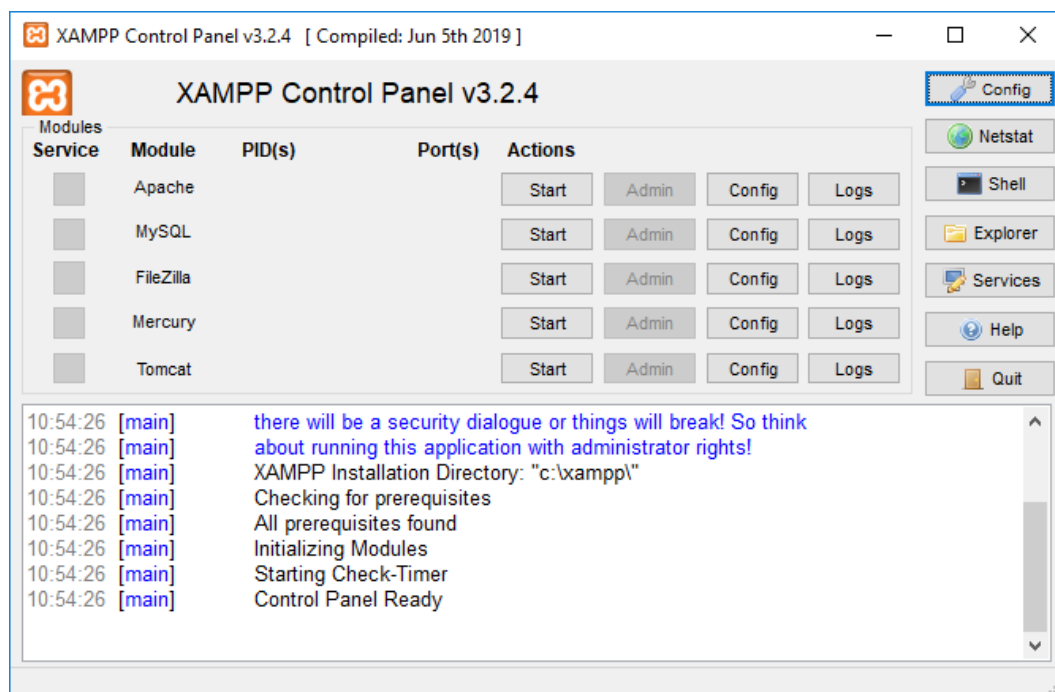
Em sequência, entre as linhas 11 e 19, será criada a tabela “Images” que armazenará os registros das imagens. As colunas dentro desse intervalo, são praticamente iguais às citadas anteriormente, dando destaque para a “image\_desc” do tipo “text” onde o usuário irá armazenar a descrição da imagem. Não foi definido um limite para este campo, mas é sempre uma boa prática definir. É possível perceber a presença do “user\_id” novamente; desta vez ele não será

incrementado automaticamente, pois para isso já se tem o “image\_id”, portanto ele servirá apenas para criar um vínculo com a tabela Users. Nas linhas 17 e 18 este vínculo é definido, onde uma chave estrangeira é criada, fazendo com que o “user\_id” da tabela “Images” se conecte ao “user\_id” da tabela Users. Isso significa que no futuro, ao ser adicionada uma imagem no banco de dados, esta estará ligada automaticamente ao usuário que a enviou. Se isso não for feito, qualquer usuário poderá recuperar qualquer imagem enviada, o que causaria uma quebra no princípio da privacidade da aplicação.

Uma aplicação Web não funciona somente por meio de código HTML, CSS, Javascript etc. É necessário que ela esteja hospedada em um servidor o qual permite inúmeras requisições. De acordo com Vasconcelos (2021), uma referência no ramo, os Servidores Web são simples programas de computador que conseguem entregar uma página Web solicitada pelo cliente sob demanda. Ele pode responder a solicitações de diversos protocolos, tais como HTTP, HTTPS, FTP, sFTP, etc.

O My.Img foi hospedado localmente para fins didáticos, e para isso, foi utilizada uma ferramenta muito famosa, o Apache XAMPP. “O XAMPP é um pacote com os principais servidores de código aberto do mercado, incluindo FTP, banco de dados MySQL e Apache com suporte [sic] às linguagens PHP e Perl.” (HIGA, 2012). Vale ressaltar que o banco de dados foi substituído pelo MariaDB, uma versão open source já que o MySQL foi comprado pela Oracle. No windows, para usar um destes servidores ou o banco de dados, basta abrir o painel do XAMPP e clicar em habilitar de acordo com a necessidade. No linux, essa opção pode ser acionada pelo terminal com o comando **sudo /opt/lampp/lampp start**, que habilita todas o servidores e banco de dados, ou **sudo /opt/lampp/lampp start<server>** para habilitar componentes individuais.

Imagem 8: Painel de Controle do XAMPP



Fonte: <<https://oraculoti.com.br/2019/07/01/o-que-e-e-para-que-serve-o-xampp/>>.

Na **Imagem 8**, é possível ver o painel de controle do XAMPP sendo executado no ambiente Windows. Na parte superior, no lado esquerdo, são listadas as funcionalidades presentes na ferramenta. Ao centro, na coluna “actions”, podem ser habilitados, configurados ou abertos os logs dos recursos. A direita há uma lista de outras funcionalidades, dando destaque a “Config” onde estão presentes várias configurações da ferramenta, o “Shell” que abre um terminal caso o usuário necessite usá-lo; e o “Explorer”, que localiza e abre a pasta “htdocs” onde serão adicionadas as pastas e arquivos do My.Img.

## 2.4. Conectando a Aplicação com o PHP, PDO e DAO

Segundo Dall’Oglio (2018, p. 22), a linguagem de programação PHP que a primeiro momento significava Personal Home Page Tools, foi criada por Rasmus Lerdorf em outubro de 1994. Era formada por um conjunto de scripts escritos em linguagem C, que serviam para criar páginas dinâmicas que seu criador usava para monitorar o acesso ao seu currículo na internet. Com o passar do tempo, o PHP foi se popularizando e mais pessoas passaram a usá-lo, e a ele foram adicionados novos recursos, como a capacidade de se conectar a uma base de dados.

Após várias atualizações, foram adicionadas inúmeras funcionalidades ao PHP. Atualmente, ele se encontra na versão 8.0. Para se utilizar o PHP é necessário instalá-lo e executá-lo por meio de um servidor. “O PHP é focado principalmente nos scripts do lado do servidor, portanto, você pode fazer qualquer coisa que outro programa CGI pode fazer, como



coletar dados de formulários, gerar páginas com conteúdo dinâmico ou enviar e receber cookies.”(PHP, 2022). O XAMPP, citado anteriormente, já vem com a última versão do PHP, ou seja, ao instalar a ferramenta, o PHP será instalado também.

O PHP normalmente utiliza a extensão .php e caso o desenvolvedor queira programar uma página da web, ao invés de usar a extensão html convencional, deve ser usada php. Dessa forma a página será processada e os trechos php serão lidos e interpretados corretamente. E por falar em trechos, assim como o javascript, o PHP pode ser codificado dentro da estrutura HTML (mesmo sendo esta uma prática não recomendada para certas situações) por meio de elementos php da seguinte forma: <?php [código-fonte] ?> . Se for necessário apenas imprimir um texto, o PHP implementa uma outra forma: <?= [texto] ?>.

A linguagem PHP é fracamente tipada e suporta praticamente todos os tipos de variáveis encontradas nas outras linguagens, sendo que estas devem obrigatoriamente iniciarem com um cifrão (\$), por exemplo, <?php \$idade = 25 ?>. É declarada a variável idade, como ela recebe um valor inteiro, o interpretador do PHP rapidamente vai defini-la como do tipo inteiro (int). Além disso, o PHP dispõe de funções (function), bibliotecas, estruturas de repetição, estruturas condicionais, tratamento de exceções e muito mais. Vale destacar um dos recursos mais utilizados no PHP: estrutura de dados; que são usadas muito frequentemente para armazenar um conjunto de dados trazidos de uma base de dados.

Imagem 9: img\_send.php (Arquivo para enviar imagens para o bd)

```
21 //verifica se a imagem foi selecionada.
22 if(!empty($_FILES['picture'])
23    && !empty($_POST['img_title'])
24    && !empty($_POST['img_desc'])
25    && !empty($_SESSION['login'])) {
26
27     //Guarda a extensão do arquivo.
28     $fileExtension = strtolower(substr($_FILES['picture']['name'], -4));
29     $newName = md5(time()) . $fileExtension;
30
31     //move a imagem selecionada da pasta temp para a pasta do usuário.
32     move_uploaded_file($_FILES['picture']['tmp_name'], $userdir . $newName);
```

Fonte: O autor.

Conforme ilustrado na **Imagem 9**, entre as linhas 21 e 32, o código resulta no envio de uma imagem para o banco de dados. Basicamente, foi utilizada a programação estruturada para desenvolver a solução, mas vale lembrar que esta é apenas uma parte do código, o restante foi abstraído. Da linha 22 a linha 25 uma função condicional verifica se quatro valores não estão vazios, que são o picture, img\_title, img\_desc e login. Estes são basicamente os elementos do

formulário de envio de uma imagem, explicado anteriormente. `$_FILES` é a superglobal responsável pelo envio de arquivos para o banco de dados, enquanto o `$_POST` faz exatamente a mesma coisa, porém, com o tipo string. Ambos escondem os dados no corpo da requisição. Para que o arquivo seja persistido no Maria DB, neste caso, precisamos informar o caminho completo da pasta junto ao nome deste. Na linha 28, uma variável armazena os quatro últimos caracteres do nome da imagem por meio da função “substr”, e ainda converte todos os caracteres para caixa baixa (minúsculas) com a função “strtolower”.

Na linha 29 a variável “newName” recebe a função md5 (que converte um valor para uma string de 32 caracteres) com outra função dentro dela, a função time (recupera a hora atual em segundos desde 1 janeiro de 1970 às 00:00:00 GMT). Qual o objetivo dessa função? A cada segundo, um novo valor de time é gerado, quando a função md5 recebe esse valor, ele o transforma em uma cadeia de 32 caracteres; em outras palavras, nunca será gerado o mesmo valor. O ponto representa concatenação de strings, ou seja, o valor aleatório do md5 vai ser unido ao valor contido na variável fileExtension (vai depender da extensão da imagem: .jpg, .png, .gif etc).

Quando um arquivo é enviado para o servidor, ele será armazenado temporariamente em uma pasta (tmp\_name). Dessa forma, devemos movê-lo para o seu destino via código. Na linha 32, a função move\_uploaded\_file() faz exatamente isso; recebe o `$_FILES` com o elemento e a pasta temporária, logo após o diretório do usuário no servidor concatenado com nome da imagem, que está sendo renomeada para o valor contido na variável newName, discutida no parágrafo anterior.

Para que os dados sejam persistidos em uma base de dados através do PHP, ele dispõe de algumas funcionalidades, a mais abordada é o PDO.

A extensão PHP Data Objects (PDO) define uma interface leve e consistente para acessar bancos de dados em PHP. Cada driver de banco de dados que implementa a interface PDO pode expor recursos específicos do banco de dados como funções de extensão regulares. Observe que você não pode executar nenhuma função de banco de dados usando a extensão PDO sozinha; você deve usar um driver PDO específico do banco de dados para acessar um servidor de banco de dados.(PHP, 2022)

E continua:

O PDO fornece uma camada de abstração de acesso a dados, o que significa que, independentemente de qual banco de dados você está usando, você usa as mesmas funções para emitir consultas e buscar dados. O PDO não fornece uma abstração de banco de dados; ele não reescreve o SQL ou emula recursos ausentes. Você deve usar uma camada de abstração completa se precisar desse recurso. (PHP, 2022)

Imagem 10: Arquivo de Configuração do My.Img

```
1  <?php
2  include 'DAO.php';
3
4  //Conexão com o banco de dados-----
5  $dsn = 'mysql:host=localhost;dbname=myimgdatabase;';
6  $user = 'root';
7  $pass = '';
8  $myimgconnect = new DAO($dsn,$user,$pass);
9  $publicDir = "../uploads/";
10 //-----
11
12 ?>
```

Fonte: O autor.

A **Imagem 10** ilustra o código do arquivo de configuração da aplicação My.Img, nele são definidos dados como, conexão do banco de dados, diretórios etc. A linha 5 define o dsn, ou seja, as informações sobre o banco de dados que será feita a conexão, neste caso, será usado o mariaDB cujo host é localhost (127.0.0.1). A base de dados é “myimgdatabase”. Por conseguinte, nas linhas 6 e 7, foram declaradas as variáveis user, que definirá qual o usuário, e pass que definirá qual a senha. Para fins didáticos, foram deixadas no seu formato padrão (user: root, pass: “ ” ). Por fim, é declarada a variável “myimgconnect”, que será usada por toda a aplicação para fazer operações diversas com o banco de dados. Essa variável recebe um objeto DAO que implementa as variáveis anteriormente definidas na sua interface. DAO não é PDO, DAO que será abordado mais à frente, implementa um objeto PDO, por isso que ele recebe os dados referentes ao banco de dados.

Nem sempre uma abordagem “crua” das ferramentas de uma linguagem é suficiente. Às vezes é necessário adaptar suas funcionalidades para uma melhor utilização destas. O PDO, descrito anteriormente, implementa todas as funcionalidades para a conexão e comunicação com uma base de dados, todavia, o seu uso demanda, em muitas ocasiões, de repetição de código, o que acaba poluindo as páginas da aplicação, além de misturar as operações SQL com a lógica de negócio.

Para amenizar essa situação, foi utilizado o padrão DAO de uma forma bem simples. De acordo com a Devmedia (2014), o padrão de projeto DAO surgiu com a necessidade de separarmos a lógica de negócios da lógica de persistência de dados. Este padrão permite que possamos mudar a forma de persistência sem que isso influencie em nada na lógica de negócio, além de tornar nossas classes mais legíveis. Esse padrão faz uso de Orientação a Objetos, sendo organizado em classes.

Classes DAO são responsáveis por trocar informações com o SGBD e fornecer operações CRUD e de pesquisas, elas devem ser capazes de buscar dados no banco e transformar esses em objetos ou lista de objetos, fazendo uso de listas genéricas [...], também deverão receber os objetos, converter em instruções SQL e mandar para o banco de dados.(DEV MEDIA, 2014 )

O devmedia ainda afirma que caso saibamos aplicar corretamente este padrão, ele será capaz de abstrair completamente o modo de busca e gravação de dados, tornando para a aplicação uma abordagem mais transparente, e facilitando a manutenção ou na hora de migrar de banco de dados.

Imagem 12: DAO.php

```
69 //Faz um select no banco de dados
70 public function select($query, $params=array()){
71     $this->setParams($query, "select" ?? "SELECT", $params);
72 }
73
74 //Faz um insert no banco de dados
75 public function insert($query, $params=array()){
76     //verifica se os dados a serem cadastrado já existem
77     $this->setParams($query, "insert" ?? "INSERT", $params);
78 }
79
80 //Faz um update no banco de dados
81 public function update($query, $params=array()){
82     $this->setParams($query, "update" ?? "UPDATE", $params);
83 }
84
85 //Faz um delete no banco de dados
86 public function delete($query, $params=array()){
87     $this->setParams($query, "delete" ?? "DELETE", $params);
88 }
```

Fonte: O autor.

Na **Imagem 11**, é representada uma parte do arquivo DAO.php responsável por reduzir e simplificar o acesso e manipulação dos dados do banco de dados. Entre as linhas 70 e 72, a função select recebe dois parâmetros, “query” e “params” (um arranjo de valores). No parâmetro “query” será inserida uma string contendo o comando sql, e no “params”, serão inseridos os parâmetros de acordo com a query escrita. Na linha 71, é chamada a função “setParams”; esta não será abordada por conta da sua complexidade. Porém ela irá validar e executar a instrução SQL inserida no parâmetro “query”, além de verificar se o comando foi realmente um SELECT e ainda relacionar a “query” com o parâmetro “params”. Essa relação se dará por meio do “bindParam”, uma função do PDO que reduz a possibilidade de um SQL

Injection; uma instrução maliciosa que pode acarretar em terríveis danos à base de dados da aplicação. O restante das linhas faz a mesma coisa, mudando apenas a instrução almejada, seja ela um INSERT, ou UPDATE ou DELETE.

Imagem 12: register.php

```
17 //Seleciona os dados do DB baseados no login inserido pelo usuário.
18 $db->select("select user_login from Users where user_login = ?", array($login));
19
20 if(empty($db->getUserData())){
21     if($login === '' && $email === '' && $pass === ''){
22     } else {
23         $db->insert("insert into Users (user_login, user_password, user_email)
24                 values (?, ?, ?) ",
25                 array($login, $pass, $email));
26         header("location: ../index.php");
27     }
28
29 } else {
30     header("location: ../app/register_user.php?=error");
31 }
```

Fonte: O autor.

Acima, a **Imagem 12** ilustra a abordagem do padrão DAO utilizada na aplicação My.Img para realizar operações CRUD. No exemplo, foi ilustrado um trecho de código do arquivo register.php, responsável por validar os dados inseridos pelo usuário no formulário antes de enviá-los para o banco de dados. Na linha 18, é feito um SELECT usando um objeto DAO nomeado como “db”. No seu corpo foi definida uma instrução que irá buscar a coluna “user\_login” na tabela “Users” onde esta coluna é igual a “?”. A interrogação serve para abstrair o valor a ser recebido pela coluna “user\_login”, fazendo com que este campo seja relacionado com o primeiro registro do array mais a frente, ou seja, o valor da coluna será igual ao valor contido no parâmetro “login”. Vale ressaltar que apenas esta linha de código abstraiu várias outras contidas dentro arquivo DAO.php, além é claro de várias verificações que estão sendo feitas por ele.

Na linha 20 um if é declarado, ele verifica se o valor recuperado do SELECT feito pelo objeto “db” está vazio; caso esteja, o programa continua para a inserção dos dados na base de dados; se não estiver, o usuário será redirecionado para a página de registro com um parâmetro “error” no corpo da requisição. Este parâmetro será tratado e utilizado para mostrar uma mensagem na tela de que este usuário já está cadastrado. Na linha 21 são feitas outras verificações se os dados inseridos pelo usuário estão vazios, se não estiverem, as linhas 23 à 26 são executadas. Nestas, o objeto “db” novamente entra em cena, inserindo o usuário na base de

dados. Note que desta vez é utilizado o comando INSERT o qual recebe três valores, o “user” (login), “pass” (senha) e “email” (email). Após a inserção o usuário é direcionado para a página inicial onde ele poderá fazer seu primeiro login na aplicação My.Img.

### **3. METODOLOGIA**

Para a realização da pesquisa, foram utilizados muitos sites conhecidos que dispõem de artigos sobre diversas áreas da computação. Além de bibliografias sobre os Sistemas Gerenciadores de Banco de Dados, Desenvolvimento Web, Orientação a Objetos em PHP e tecnologias mais específicas como o CSS e Javascript.

Quanto ao tipo de pesquisa, foi realizada a pesquisa descritiva para demonstrar alguns conceitos, características e como as linguagens e ferramentas se relacionam. Dessa forma, o desenvolvimento de uma aplicação com estas foi imprescindível para o desenvolvimento do estudo como um todo.

A aplicação desenvolvida foi a My.Img, um banco de imagens capaz de registrar um usuário, bem como permitir que eles possam persistir suas imagens e informações no banco de dados MariaDB. O usuário pode interagir com ele por meio do navegador. Para a sua concepção, foi projetado um caso de uso; nele foram definidas as principais funcionalidades desejadas além das citadas anteriormente.

Para simplificar ao máximo a aplicação e torná-la didática, não foi utilizado nenhum tipo de padrão de projeto (a não ser o DAO) ou mecanismo de injeção de dependência, por exemplo. Por se tratar de uma perspectiva mais simplista, foi adotada uma metodologia “pura”, ou seja, uma metodologia que foque diretamente na linguagem e nas principais funcionalidades.

As páginas da aplicação foram programadas em HTML e estilizadas com CSS e Bootstrap. As funcionalidades em sua maioria foram desenvolvidas em PHP; uma delas é a SESSÃO, pela qual os usuários persistem dados temporários no servidor e ao fazerem o logout (saírem) da página, este é limpo impedindo que o mesmo usuário consiga recarregar a página uma vez já feita a sua saída. Algumas outras funcionalidades ficaram a cargo do javascript por meio da sua biblioteca JQuery, como para verificar se os campos usuário e senha estão preenchidos, por exemplo.

Foi utilizado o Apache Xampp para hospedar a My.Img e para persistir os dados no Mariadb, visto que esta aplicação disponibiliza os recursos necessários para este estudo. Inicialmente, instruções SQL foram desenvolvidas para criar o banco e as tabelas diretamente no PHPMyAdmin, o sistema de gerenciamento visual do banco de dados.

Os arquivos PHP foram separados de acordo com a sua funcionalidade, visando a melhor estruturação da aplicação. Eles foram dispostos na pasta “Source” e seu carregamento centralizado no arquivo de configuração da aplicação “Config.php”.

Os resultados deste estudo serão representados em forma de análise qualitativa dos recursos utilizados para a produção da aplicação e como eles se desempenharam durante o seu uso.

#### **4. RESULTADOS E DISCUSSÕES**

De acordo com os resultados do estudo realizado e da construção e persistência da aplicação My.Img, foi possível observar os seguintes resultados:

Uma aplicação depende de uma boa estruturação de diretórios, mesmo que isso pareça ao óbvio. Organizar os seus componentes, os separando em diretórios específicos, melhorou bastante o controle deles, bem como a sua localização ao necessitar de algo específico. Por exemplo, na pasta “app” foram dispostas as pastas “js” para os arquivos javascript e “css” para os arquivos de estilização; na sua raiz foi inserido o arquivo “home.php” que fornece um template para a página inicial do usuário.

O Bootstrap facilitou a organização dos componentes HTML, principalmente os do formulário de envio de imagens. A sua “facilidade” contribuiu para um ganho significativo no tempo de construção e estilização dos componentes, além é óbvio, da qualidade visual deixada por ele.

O PHP solidificou e separou a camada de comunicação com o banco de dados. A sua biblioteca PDO melhorou a qualidade e segurança da comunicação com o MariaDB, mas usá-lo avulsamente não é um bom caminho, por isso, foi muito útil criar uma classe (DAO) que organizasse as instruções SQL, os parâmetros e outras verificações. Portanto, foi percebido um grande salto na redução de código ao centralizar as funcionalidades.

Ao organizar os arquivos PHP de acordo com sua funcionalidade, foi percebida uma maior facilidade de fazer manutenção do código, além da facilidade de implementar novas funcionalidades e um leve ganho na performance. O arquivo de configuração “Config.php”, o qual foi responsável por reunir os outros, teve grande importância, uma vez que ele centralizou todos os componentes (constantes como o diretório do usuário) que ao invés de serem buscadas em diferentes arquivos foram concentrados em apenas um. Este arquivo ficou exposto no diretório público da aplicação para fins de estudo, mas em uma abordagem profissional ele deve

ser encapsulado fora deste diretório; sendo que outro arquivo PHP teria a função de chamá-lo nas páginas.

Ainda em relação ao PHP, o seu mecanismo de sessão (\$\_SESSION) possibilitou que os usuários pudessem fechar a página da aplicação My.Img e abri-la novamente sem a necessidade de refazer o login. Ainda, essa superglobal persistiu temporariamente, o que facilitou seu uso em outros mecanismos. Todavia, não dá para persistir qualquer dado nela; de acordo o que foi visto, armazenar dados sensíveis, como senhas ou dados pessoais, pode ser muito arriscado.

O uso do JQuery reduziu drasticamente o volume de código a ser programado se ao invés dele fosse utilizada a linguagem ECMAScript pura. Por exemplo, para programar o menu “Modal Popup”, descrito em tópicos anteriores, no JQuery foram utilizadas em torno de dez linhas de código, enquanto a abordagem direta, utilizou quase trinta linhas de código. Para esse exemplo, variáveis e funções foram separadas por linha, mesmo assim, é óbvia a diferença.

O Apache Xampp viabilizou a hospedagem e persistência dos dados. Por meio da sua plataforma gráfica PHPMyAdmin foi possível criar a base de dados inteira sem precisar programar uma linha de código. Suas funcionalidades são tão abrangentes, que através de alguns poucos cliques foi possível até mesmo criar as relações entre as tabelas. Contudo, o Xampp reduziu a complexidade da construção da base de dados, ampliou as suas possibilidades e reduziu o seu tempo de desenvolvimento.

## **5. CONCLUSÃO**

Há quem diga que desenvolver software é uma tarefa fácil; o que se prova o contrário quando levado para a prática. Seguir um cronograma e definir bem os requisitos de um software é uma atividade de extrema importância e deveria ser uma prioridade para a maioria destes, porém, ainda está longe de ser uma realidade, logo, pois, a demanda atualmente é muita alta e as exigências por parte dos usuários são quase surreais.

Uma abordagem direta, ou seja, pura, demanda mais tempo e recursos para serem disponibilizados do que outras formas, como por exemplo, o MVC (Model-View-Controller) que pode ser implementado ao PHP via Composer e outros recursos, ou ainda, ASP NET e Java Web. Entretanto, se a aplicação for bem projetada, dependendo do seu objetivo, a abordagem direta é uma ótima opção.

Desenvolver uma aplicação é uma arte, se o artista (desenvolvedor) é perfeccionista e procura cumprir todos os prazos e metas estabelecidas, o seu índice de assertividade será bem



alto. Mas se o artista é desleixado ou não abre a sua mente para outros horizontes (frameworks, bibliotecas, boas práticas) mesmo que sua arte fique agradável de se ver (design), ela conterà pontos defeituosos (falhas, vulnerabilidades) que quando vistos por outras perspectivas serão chamarizes para pessoas mal-intencionadas que não medirão esforços até destruí-la por completo.

Portanto, é muito importante entender o objetivo da aplicação, bem como a necessidade que a trouxe à tona, para que através disto se possa mitigar a melhor forma de criá-la e as ferramentas mais adequadas, sempre visando evitar prejuízos, perdas ou futuras complicações.

## REFERÊNCIAS

ALEXANDRUK, Marcos. **SQL – Structured Query Language**: rápido e fácil. São Paulo : Universidade Nove de Julho, UNINOVE, 2018. 232 p.

CSS. **MDN Web Docs**, 2021. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/CSS>>. Acesso em: 7 jun. 2022.

DALL’OGLIO, Pablo. **PHP: Programando com Orientação a Objetos**. 4. ed. São Paulo: Novatec, 2018. 568 p.

Diagrama de caso de uso UML: O que é, como fazer e exemplos. **Lucidchart**, 2022. Disponível em: <<https://www.lucidchart.com/pages/pt/diagrama-de-caso-de-uso-uml>>. Acesso em: 14 jun. 2022.

ELMASRI, Ramez; NAVATHE, Shamkant B. **Sistemas de banco de dados**. 6. ed. rev. São Paulo: Pearson Addison Wesley, 2011. 808 p.

Front end: O que é, como funciona e qual a importância. **Totvs**, 2021. Disponível em: <<https://www.totvs.com/blog/developers/front-end/>>. Acesso em: 16 jun. 2022.

Gerenciamento de Banco de Dados: Análise Comparativa de SGBD’S. **Devmedia**, 2014. Disponível em: <<https://www.devmedia.com.br/gerenciamento-de-banco-de-dados-analise-comparativa-de-sgbd-s/30788#:~:text=O%20Sistema%20de%20Gest%C3%A3o%20de,dados%20da%20base%20de%20dados>>. Acesso em: 7 jun. 2022.

Get started with Bootstrap. **Getbootstrap**, 2022. Disponível em: <<https://getbootstrap.com/docs/5.2/getting-started/introduction/>>. Acesso em: 7 jun. 2022.

HANASHIRO, Akira. VS Code: O que é e por que você deve usar?. **Treinaweb**, 2021. Disponível em: <<https://www.treinaweb.com.br/blog/vs-code-o-que-e-e-por-que-voce-deve-usar>>. Acesso em: 16 jun. 2022.

HIGA, Paulo. O que é XAMPP e para que serve. **Techtudo**, 2012. Disponível em: <<https://www.techtudo.com.br/noticias/2012/02/o-que-e-xampp-e-para-que-serve.ghml>>. Acesso em: 7 jun. 2022.

HTML Introduction. **W3SCHOOLS**, 2022. Disponível em: <[https://www.w3schools.com/html/html\\_intro.asp](https://www.w3schools.com/html/html_intro.asp)>. Acesso em: 31 mai. 2022.

Introdução PDO. **PHP**, 2022. Disponível em: <[https://www.php.net/manual/pt\\_BR/intro.pdo.php](https://www.php.net/manual/pt_BR/intro.pdo.php)>. Acesso em: 7 jun. 2022.

JavaScript. **MDN Web Docs**, 2021. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>>. Acesso em: 16 jun. 2022.

LIMA, Guilherme. Bootstrap: O que é, como e quando usar?. **Alura**, 2021. Disponível em: <<https://www.alura.com.br/artigos/bootstrap>>. Acesso em: 7 jun. 2022.

NOLETO, Cairo. Aplicações web: entenda o que são e como funcionam!. **Betrybe**, 2020. Disponível em: <<https://blog.betrybe.com/desenvolvimento-web/aplicacoes-web/>>. Acesso em: 27 jun. 2022.

NOLETO, Cairo. CSS: o que é, guia sobre como usar e vantagens!. **Betrybe**, 2022. Disponível em: <<https://blog.betrybe.com/css/>>. Acesso em: 7 jun. 2022.

OPENJS FOUNDATION. **JQuery**, 2022. Página inicial. Disponível em: <<https://jquery.com/>>. Acesso em: 16 jun. 2022.

O QUE é um diagrama UML?: Por que usar um diagrama UML?. **Lucidchart**, 2022. Disponível em: <<https://www.lucidchart.com/pages/pt/o-que-e-uml>>. Acesso em: 14 jun. 2022.

O que é CSS?: Guia Básico para Iniciantes. **Hostinger**, 2022. Disponível em: <<https://www.hostinger.com.br/tutoriais/o-que-e-css-guia-basico-de-css>>. Acesso em: 7 jun. 2022.

O que é HTML5. **Devmedia**, 2012. Disponível em: <<https://www.devmedia.com.br/o-que-e-o-html5/25820>>. Acesso em: 31 mai. 2022.

O que o PHP pode fazer?. **PHP**, 2022. Disponível em: <[https://www.php.net/manual/pt\\_BR/intro-whatcando.php](https://www.php.net/manual/pt_BR/intro-whatcando.php)>. Acesso em: 7 jun. 2022.

O que é um Servidor Web e como funciona?. **Devmedia**, 2014. Disponível em: <<https://www.devmedia.com.br/dao-pattern-persistencia-de-dados-utilizando-o-padrao-dao/30999>>. Acesso em: 7 jun. 2022.

Organização de Arquivos. **Holder Tech**, 2020. Disponível em: <<https://holdertech.com.br/organizacao-de-arquivos/>>. Acesso em: 16 jun. 2022.

PRESSMAN, Roger S. **Engenharia de Software: Uma abordagem profissional**. 7. ed. rev. Porto Alegre: AMGH, 2011. 779 p.

SOMMERVILLE, Ian. **Engenharia de Software**. 9. ed. rev. São Paulo: Pearson, 2011. 545 p.

SOUTO, Mario. O que é front-end e back-end?. **Alura**, 2019. Disponível em: <<https://www.alura.com.br/artigos/o-que-e-front-end-e-back-end>>. Acesso em: 16 jun. 2022.

VASCONCELOS, Luiz Arthur. O que é um Servidor Web e como funciona?. **Hostgator**, 2021. Disponível em: <<https://www.hostgator.com.br/blog/o-que-e-um-servidor-web-e-como-funciona/>>. Acesso em: 7 jun. 2022.

SILVEIRA, Paulo. O que é SQL?. **Alura**, 2019. Disponível em: <<https://www.alura.com.br/artigos/o-que-e-sql#:~:text=SQL%20significa%20Standard%20Query%20Language,SQL%20Server%2C%20entre%20muitos%20outros>>. Acesso em: 7 jun. 2022.

