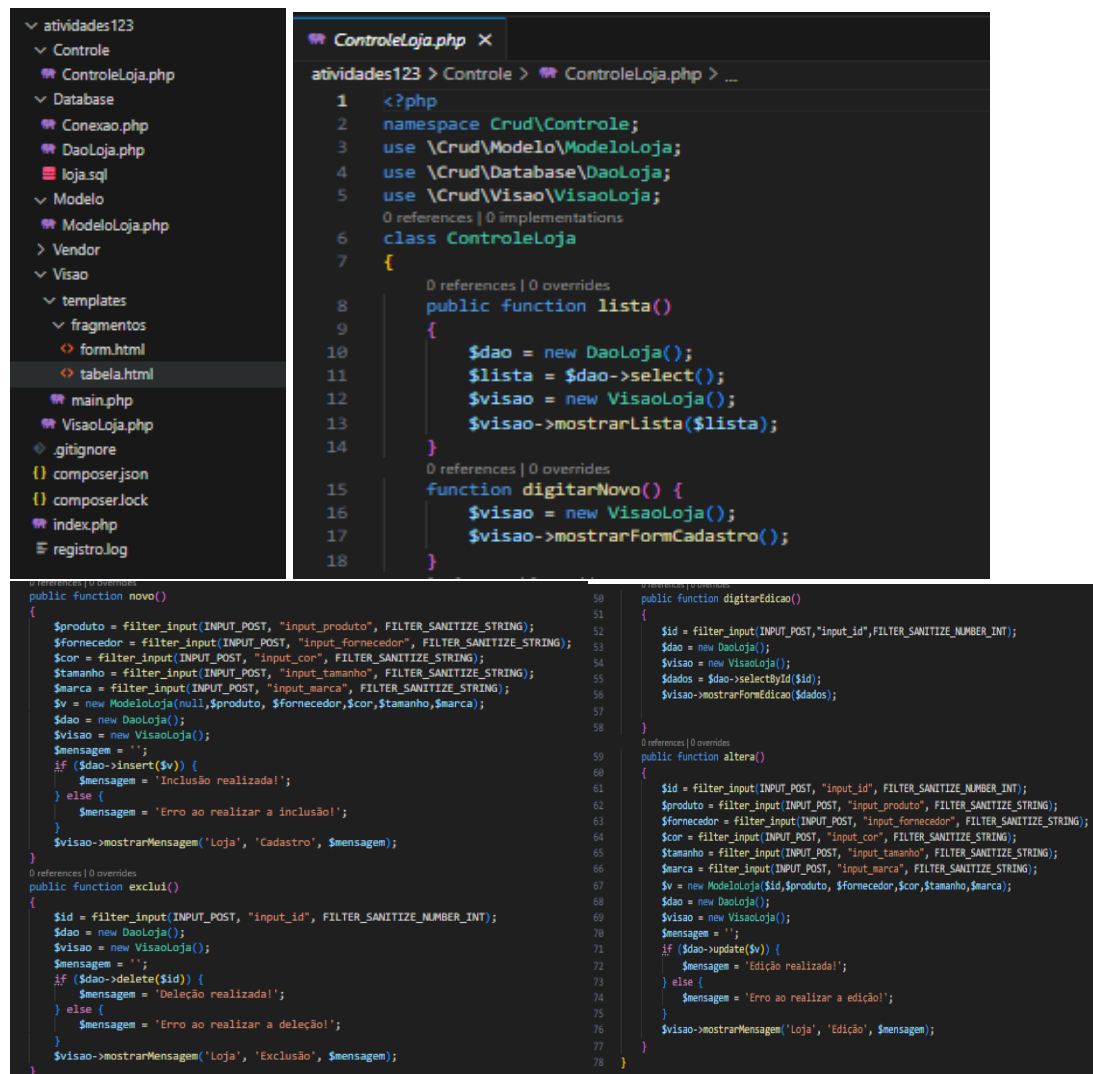


## ATIVIDADES 1,2 E 3

A atividade não foi complicada, embora uma grande dificuldade de lembrar os comandos de PHP após 3 meses. O main foi o responsável por ligar o back ao front, de forma a funcionar juntos. A atividade é sobre uma loja qualquer e informações de seus produtos, contém uma tabela com 5 atributos. Possui as opções de listar os itens, excluir, editar e inserir novos produtos. Faz ligação com o banco de dados sql local.

Segue as capturas das suas funcionalidades.

Contamos com o arquivo de controle da loja, onde estão as funções citadas acima(listar,excluir,adicionar novo e editar).



The image shows a code editor with a file explorer on the left and the main code area on the right. The file explorer shows a project structure with folders like 'atividades123', 'Controle', 'Database', 'Modelo', 'Visao', and 'templates'. The main code area displays the 'ControleLoja.php' file, which contains PHP code for a CRUD application. The code includes namespaces, class definitions, and methods for listing, adding, editing, and deleting products. The code is written in PHP and uses a namespace-based architecture.

```

1  <?php
2  namespace Crud\Controle;
3  use \Crud\Modelo\ModeloLoja;
4  use \Crud\Database\DaoLoja;
5  use \Crud\Visao\VisaoLoja;
6  class ControleLoja
7  {
8      public function lista()
9      {
10         $dao = new DaoLoja();
11         $lista = $dao->select();
12         $visao = new VisaoLoja();
13         $visao->mostrarLista($lista);
14     }
15     function digitarNovo() {
16         $visao = new VisaoLoja();
17         $visao->mostrarFormCadastro();
18     }
19     public function novo()
20     {
21         $produto = filter_input(INPUT_POST, "input_produto", FILTER_SANITIZE_STRING);
22         $fornecedor = filter_input(INPUT_POST, "input_fornecedor", FILTER_SANITIZE_STRING);
23         $cor = filter_input(INPUT_POST, "input_cor", FILTER_SANITIZE_STRING);
24         $tamanho = filter_input(INPUT_POST, "input_tamanho", FILTER_SANITIZE_STRING);
25         $marca = filter_input(INPUT_POST, "input_marca", FILTER_SANITIZE_STRING);
26         $v = new ModeloLoja(null, $produto, $fornecedor, $cor, $tamanho, $marca);
27         $dao = new DaoLoja();
28         $visao = new VisaoLoja();
29         $mensagem = '';
30         if ($dao->insert($v)) {
31             $mensagem = 'Inclusão realizada!';
32         } else {
33             $mensagem = 'Erro ao realizar a inclusão!';
34         }
35         $visao->mostrarMensagem('Loja', 'Cadastro', $mensagem);
36     }
37     public function excluir()
38     {
39         $id = filter_input(INPUT_POST, "input_id", FILTER_SANITIZE_NUMBER_INT);
40         $dao = new DaoLoja();
41         $visao = new VisaoLoja();
42         $mensagem = '';
43         if ($dao->delete($id)) {
44             $mensagem = 'Deleção realizada!';
45         } else {
46             $mensagem = 'Erro ao realizar a deleção!';
47         }
48         $visao->mostrarMensagem('Loja', 'Exclusão', $mensagem);
49     }
50     public function digitarEdicao()
51     {
52         $id = filter_input(INPUT_POST, "input_id", FILTER_SANITIZE_NUMBER_INT);
53         $dao = new DaoLoja();
54         $visao = new VisaoLoja();
55         $dados = $dao->selectById($id);
56         $visao->mostrarFormEdicao($dados);
57     }
58     public function altera()
59     {
60         $id = filter_input(INPUT_POST, "input_id", FILTER_SANITIZE_NUMBER_INT);
61         $produto = filter_input(INPUT_POST, "input_produto", FILTER_SANITIZE_STRING);
62         $fornecedor = filter_input(INPUT_POST, "input_fornecedor", FILTER_SANITIZE_STRING);
63         $cor = filter_input(INPUT_POST, "input_cor", FILTER_SANITIZE_STRING);
64         $tamanho = filter_input(INPUT_POST, "input_tamanho", FILTER_SANITIZE_STRING);
65         $marca = filter_input(INPUT_POST, "input_marca", FILTER_SANITIZE_STRING);
66         $v = new ModeloLoja($id, $produto, $fornecedor, $cor, $tamanho, $marca);
67         $dao = new DaoLoja();
68         $visao = new VisaoLoja();
69         $mensagem = '';
70         if ($dao->update($v)) {
71             $mensagem = 'Edição realizada!';
72         } else {
73             $mensagem = 'Erro ao realizar a edição!';
74         }
75         $visao->mostrarMensagem('Loja', 'Edição', $mensagem);
76     }
77 }

```

Seguimos com os códigos de conexão ao banco de dados e a inserção de itens.

```
Conexao.php X
atividades123 > Database > Conexao.php > ...
5 class Conexao
6 {
7     2 references
8     private static $user = 'root';
9     private static $password = 'root';
10    private static $con = null;
11
12    8 references | 0 overrides
13    public static function getConnection(){
14        if (Conexao::$con == null) {
15            try {
16                Conexao::$con = new PDO(Conexao::$dsn, Conexao::$user, Conexao::$password);
17            } catch (PDOException $e) {
18                echo 'Erro ao conectar com o banco!<br>';
19                // echo $e->getMessage();
20                // phpinfo();
21            }
22        }
23        return Conexao::$con;
24    }
25
26    4 references | 0 overrides
27    public static function getPrepareStatement(string $sql)
28    {
29        if(Conexao::getConnection()){
30            try {
31                return Conexao::$con->prepare($sql);
32            } catch (PDOException $e) {
33                echo 'Erro ao preparar o comando!<br>';
34                //echo $e->getMessage();
35            }
36        }
37        return null;
38    }
39 }

DaoLoja.php X
atividades123 > Database > DaoLoja.php > DaoLoja
1 <?php
2 namespace Crud\Database;
3 use \Crud\Modelo\ModeloLoja;
4 use PDO;
5
6 6 references | 0 implementations
7 class DaoLoja
8 {
9     1 reference | 0 overrides
10    public function select()
11    {
12        $sql = "select * from Loja";
13        $pst = Conexao::getPrepareStatement($sql);
14        if (!$pst) {
15            exit("Erro ao preparar");
16        }
17        $pst->execute();
18        $result = $pst->fetchAll(PDO::FETCH_ASSOC);
19        return $result;
20    }
21
22    1 reference | 0 overrides
23    public function selectById($id)
24    {
25        $sql = "select * from Loja where id = ?";
26        $pst = Conexao::getPrepareStatement($sql);
27        if (!$pst) {
28            exit("Erro ao preparar");
29        }
30        $pst->bindValue(1, $id);
31        $pst->execute();
32        $result = $pst->fetchAll(PDO::FETCH_ASSOC)[0];
33        return $result;
34    }
35 }

1 reference | 0 overrides
public function insert(ModeloLoja $v)
{
    $sql = "insert into Loja (produto, fornecedor, cor, tamanho, marca) values (?, ?, ?, ?, ?)";
    $pst = Conexao::getConnection()->prepare($sql);
    $pst->bindValue(1, $v->getproduto(), PDO::PARAM_STR);
    $pst->bindValue(2, $v->getfornecedor(), PDO::PARAM_STR);
    $pst->bindValue(3, $v->getcor(), PDO::PARAM_STR);
    $pst->bindValue(4, $v->gettamanho(), PDO::PARAM_STR);
    $pst->bindValue(5, $v->getmarca(), PDO::PARAM_STR);
    if ($pst->execute()) {
        // echo $pst->errorInfo();
        return true;
    }
    return false;
}

1 reference | 0 overrides
public function delete($id)
{
    $sql = "delete from Loja where id = ?";
    $pst = Conexao::getConnection()->prepare($sql);
    $pst->bindValue(1, $id, PDO::PARAM_INT);
    if ($pst->execute()) {
        return true;
    }
    return false;
}

1 reference | 0 overrides
public function update(ModeloLoja $v)
{
    $sql = "update Loja set produto = ?, fornecedor = ?, cor = ?, tamanho = ?, marca = ? where id = ?";
    $pst = Conexao::getConnection()->prepare($sql);
    $pst->bindValue(1, $v->getproduto(), PDO::PARAM_STR);
    $pst->bindValue(2, $v->getfornecedor(), PDO::PARAM_STR);
    $pst->bindValue(3, $v->getcor(), PDO::PARAM_STR);
    $pst->bindValue(4, $v->gettamanho(), PDO::PARAM_STR);
    $pst->bindValue(5, $v->getmarca(), PDO::PARAM_STR);
    $pst->bindValue(6, $v->getid(), PDO::PARAM_STR);
    if ($pst->execute()) {
        return true;
    }
    return false;
}
```

O arquivo Modelo Loja conta somente com os gets e sets dos atributos.

```
ModeloLoja.php X
atividades123 > Modelo > ModeloLoja.php > ...
1 <?php
2 namespace Crud\Modelo;
3
4 use PDO;
5 use PDOException;
6
7 12 references | 0 implementations
8 class ModeloLoja
9 {
10     2 references
11     private $id;
12     2 references
13     private $produto;
14     2 references
15     private $fornecedor;
16     1 reference
17     private $cor;
18     2 references
19     private $tamanho;
20     2 references
21     private $marca;
22
23     2 references | 0 overrides
24     public function __construct($id = null, $produto, $fornecedor, $cor, $tamanho, $marca) {
25         $this->id = $id;
26         $this->produto = $produto;
27         $this->fornecedor = $fornecedor;
28         $this->cor = $cor;
29         $this->tamanho = $tamanho;
30         $this->marca = $marca;
31     }
32
33     1 reference | 0 overrides
34     public function getId() {
35         return $this->id;
36     }
37
38     0 references | 0 overrides
39     public function setId($id): self {
40         $this->id = $id;
41         return $this;
42     }
43 }
```

Foi feito para todos os atributos.

Código do front, construção dos formulários.

form.html X

atividades123 > Visão > templates > fragmentos > form.html > form.html

```

1 <form action="{fct}" method="post">
2 <input type="hidden" name="input_id" id="input_id" value="{id}"><br>
3 <label for="nome">Produto</label><br>
4 <input type="text" name="input_produto" id="input_produto" value="{produto}"><br>
5 <label for="fornecedor">Fornecedor</label><br>
6 <input type="text" name="input_fornecedor" id="input_fornecedor" value="{fornecedor}"><br>
7 <label for="cor">Cor</label><br>
8 <input type="text" name="input_cor" id="input_cor" value="{cor}"><br>
9 <label for="tamanho">Tamanho</label><br>
10 <input type="text" name="input_tamanho" id="input_tamanho" value="{tamanho}"><br>
11 <label for="marca">Marca</label><br>
12 <input type="text" name="input_marca" id="input_marca" value="{marca}"><br>
13 <button>Salvar</button>
14 </form>
15

```

tabela.html X

atividades123 > Visão > templates > fragmentos > tabela.html > table

```

1 <table border="1" style="text-align: center;">
2 <thead>
3 <tr>
4 <th>ID</th>
5 <th>Produto</th>
6 <th>Fornecedor</th>
7 <th>Cor</th>
8 <th>Tamanho</th>
9 <th>Marca</th>
10 <th colspan="2">Ação</th>
11 </tr>
12 </thead>
13 <tbody>
14 {tbody}
15 </tbody>
16 </table>
17

```

main.php X

atividades123 > Visão > templates > main.php

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 <title>Document</title>
7 </head>
8 <body>
9 <header>
10 <h1><?php echo $titulo; ?></h1>
11 <a href="/loja/lista">Listar</a>
12 <a href="/loja/digitarnovo">Novo</a>
13 </header>
14 <main>
15 <h2><?php echo $subtitulo; ?></h2>
16 <div>
17 <?php echo $conteudo; ?>
18 </div>
19 </main>
20 <footer>
21 <p>Rodapé!</p>
22 </footer>
23 </body>
24 </html>

```

Chama as funções de listar e adicionar, com as rotas; (localhost:8000/loja/lista ou localhost:8000/loja/digitarnovo)

O Visão Loja mostra o usuário o formulário e permite que ele visualize dados e as mudanças.

VisaoLoja.php X

atividades123 > Visão > VisaoLoja.php > VisaoLoja > mostrarMensagem()

```

1 <?php
2 namespace Crud\Visao;
3
4 final class VisaoLoja
5 {
6     function mostrarLista($lista) {
7         $titulo = 'Loja';
8         $subtitulo = 'Listagem';
9         $dadosPraTabela = '';
10         foreach ($lista as $linha) {
11             $dadosPraTabela .= '<tr>';
12             $dadosPraTabela .= '<td> . $linha['id'] . '</td>';
13             $dadosPraTabela .= '<td> . $linha['produto'] . '</td>';
14             $dadosPraTabela .= '<td> . $linha['fornecedor'] . '</td>';
15             $dadosPraTabela .= '<td> . $linha['cor'] . '</td>';
16             $dadosPraTabela .= '<td> . $linha['tamanho'] . '</td>';
17             $dadosPraTabela .= '<td> . $linha['marca'] . '</td>';
18
19             $dadosPraTabela .= '<td>';
20             $dadosPraTabela .= '<form action="/loja/excluir" metodo="post">';
21             $dadosPraTabela .= '<input type="hidden" name="input_id" value=" . $linha['id'] . ">';
22             $dadosPraTabela .= '<button>Excluir</button>';
23             $dadosPraTabela .= '</form>';
24             $dadosPraTabela .= '</td>';
25
26             $dadosPraTabela .= '<td>';
27             $dadosPraTabela .= '<form action="/loja/digitarEdicao" metodo="post">';
28             $dadosPraTabela .= '<input type="hidden" name="input_id" value=" . $linha['id'] . ">';
29             $dadosPraTabela .= '<button>Digitar</button>';
30             $dadosPraTabela .= '</form>';
31             $dadosPraTabela .= '</td>';
32
33             $dadosPraTabela .= '</tr>';
34
35             $fragmento = file_get_contents(__DIR__ . '/templates/fragmentos/tabela.html');
36             $conteudo = str_replace('{tbody}', $dadosPraTabela, $fragmento);
37             require_once __DIR__ . '/templates/main.php';
38
39 }

```

VisaoLoja.php X

atividades123 > Visão > VisaoLoja.php > VisaoLoja > VisaoLoja

```

1 final class VisaoLoja
2 {
3     function mostrarFormCadastro() {
4         $titulo = 'Produtos';
5         $subtitulo = 'Cadastro';
6         $form = file_get_contents(__DIR__ . '/templates/fragmentos/form.html');
7         $form = str_replace(
8             ['{fct}', '{id}', '{produto}', '{fornecedor}', '{cor}', '{tamanho}', '{marca}'],
9             ['/loja/novo'],
10             $form
11         );
12         $conteudo = $form;
13         require_once __DIR__ . '/templates/main.php';
14     }
15
16     function mostrarFormEdicao($dados) {
17         $titulo = 'Produtos';
18         $subtitulo = 'Edição';
19         $form = file_get_contents(__DIR__ . '/templates/fragmentos/form.html');
20         $form = str_replace(
21             ['{fct}', '{id}', '{produto}', '{fornecedor}', '{cor}', '{tamanho}', '{marca}'],
22             ['/loja/altera',
23                 $dados['id'],
24                 $dados['produto'],
25                 $dados['fornecedor'],
26                 $dados['cor'],
27                 $dados['tamanho'],
28                 $dados['marca'],
29             ],
30             $form
31         );
32         $conteudo = $form;
33         require_once __DIR__ . '/templates/main.php';
34     }
35
36     function mostrarMensagem($tit, $sub, $msg) {
37         $titulo = $tit;
38         $subtitulo = $sub;
39         $conteudo = $msg;
40         require_once __DIR__ . '/templates/main.php';
41     }
42 }

```

```

index.php X
atividades123 > index.php
1 <?php
2 header('Access-Control-Allow-Origin: *');
3
4 // require_once './modelo/ModeloLoja.php';
5 // require_once './database/Conexao.php';
6 // require_once './database/DaoLoja.php';
7 // require_once './visao/VisaoLoja.php';
8 // require_once './controle/ControleLoja.php';
9
10 require __DIR__ . '/vendor/autoload.php';
11
12 $router = new \Bramus\Router\Router();
13
14
15 $router->all('/{mod}/{acao}', function ($mod, $acao) {
16     $modulo = '\Crud\Controle\Controle' . ucfirst($mod);
17
18     if (class_exists($modulo)) {
19         $obj = new $modulo();
20         if (method_exists($modulo, $acao)) {
21             echo $obj->$acao();
22         } else {
23             echo 'No action...';
24         }
25     } else {
26         echo 'No module...';
27     }
28 });
29
30 $router->run();
31

```

o index usa a biblioteca Bramus Router para criar um roteador simples que pode gerenciar requisições HTTP e encaminhá-las para módulos e ações específicas.

## APLICAÇÃO

### Loja

[Listar Novo](#)

#### Listagem

ID	Produto	Fornecedor	Cor	Tamanho	Marca	Ação
1	Blusa	Magazine	azul	M	M	Excluir Editar
2	calça	renner	jeans	m	gucci	Excluir Editar

Rodapé!

### Produtos

[Listar Novo](#)

#### Cadastro

Produto   
 Fornecedor   
 Cor   
 Tamanho   
 Marca

Rodapé!

### Loja

[Listar Novo](#)

#### Listagem

ID	Produto	Fornecedor	Cor	Tamanho	Marca	Ação
1	Blusa	Magazine	azul	M	M	Excluir Editar
2	calça	renner	jeans	m	gucci	Excluir Editar
3	JAQUETA	RENNER	PRETA	G	MARISA	Excluir Editar

Rodapé!

### Loja

[Listar Novo](#)

#### Exclusão

Deleção realizada!

Rodapé!

### Loja

[Listar Novo](#)

#### Listagem

ID	Produto	Fornecedor	Cor	Tamanho	Marca	Ação
1	Blusa	Magazine	azul	M	M	Excluir Editar
3	JAQUETA	RENNER	PRETA	G	MARISA	Excluir Editar

Rodapé!

### Loja

[Listar Novo](#)

#### Edição

Edição realizada!

Rodapé!

### Loja

[Listar Novo](#)

#### Listagem

ID	Produto	Fornecedor	Cor	Tamanho	Marca	Ação
1	Blusa	Magazine	azul	M	M	Excluir Editar
3	JAQUETA	RENNER	branca	G	G	Excluir Editar

Rodapé!