

Controller Editar e Excluir

Criar Function editar e Function excluir

Corrigir a primeira linha de save

'ID\_Conteudo'=>\$this->request->getVar('ID\_Conteudo'),

```
public function editar($id = NULL){
    $model = new ConteudoModel();
    $data['conteudo'] = $model->getConteudoItem($id);
    echo view('templates/Header', $data);
    echo view('pages/ConteudoGravar',$data);
    echo view('templates/Footer');
}
public function excluir($id=NULL){
    $model = new ConteudoModel();
    $model->delete($id);
    return redirect('/');
}
public function gravar(){
    helper('form');
    $model = new ConteudoModel();
    //Validação dos campos
    if($this->validate([
        'Titulo' =>['label'=>'Título', 'rules'=>'required|min_length[3]|max_length[100]'],
        'Texto_curto'=>['label'=>'Resumo', 'rules'=>'required|min_length[3]|max_length[100]'],
        'Texto_completo'=>['label'=>'Conteúdo Completo', 'rules'=>'required|min_length[3]']
    ])){
        //Salva no banco de dados
        $model->save([
            'ID_Conteudo' => $this->request->getVar('ID_Conteudo'),
            'Titulo' => $this->request->getVar('Titulo'),
```

Implementar o value dos campos

Corrigir o name="id" para name="ID\_Conteudo"

```
<form action="<?= '/gravar' ?>" method="post">
  <div class="form-group">
    <label for="Titulo">Título</label>
    <!--isset significa se tiver conteúdo vindo do banco imprime, senão set_value apresenta o conteúdo do campo-->
    <input type="text" class="form-control" name="Titulo" value="<?= isset($conteudo['Titulo']) ? $conteudo['Titulo'] : set_value('Titulo') ?>"/>
  </div>
  <div class="form-group">
    <label for="Texto_curto">Resumo</label>
    <input type="text" class="form-control" name="Texto_curto" value="<?= isset($conteudo['Texto_curto']) ? $conteudo['Texto_curto'] : set_value('Texto_curto') ?>"/>
  </div>
  <div class="form-group">
    <label for="Texto_completo">Conteúdo Completo</label>
    <textarea name="Texto_completo" class="form-control" rows="5"><?= isset($conteudo['Texto_completo']) ? $conteudo['Texto_completo'] : set_value('Texto_completo') ?>
    </textarea>
  </div><!--hidden variável oculta para salvar id-->
  <input type="hidden" name="ID_Conteudo" value="<?= isset($conteudo['ID_Conteudo']) ? $conteudo['ID_Conteudo'] : set_value('ID_Conteudo') ?>"/>
  <div class="card my-3">
    <div class="card-footer">
      <input type="submit" name="submit" class="btn btn-primary" value="Salvar"/>
      <a href="javascript:history.back()" class="btn btn-primary">Voltar</a>
    </div>
  </div>
</form>
```

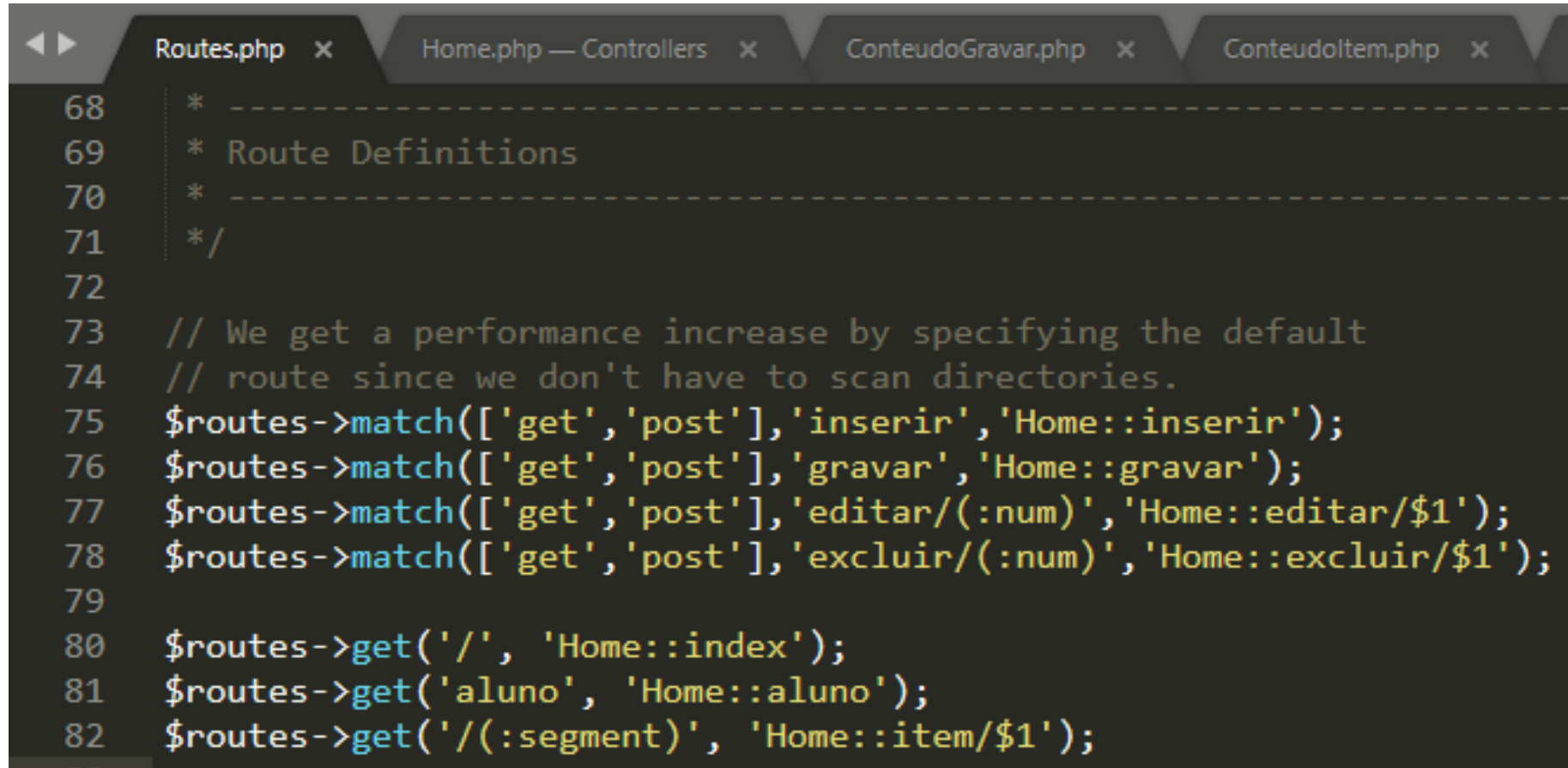
A implementação de value poderia ser no modelo tradicional ou no modelo sintético

```
<input type="text" class="form-control" name="Titulo" value=
"php
    if(!empty($conteudo['Titulo']))
        echo $conteudo['Titulo'];
    else
        echo set_value('Titulo');
/*O código php acima poderia ser escrito como:
&lt;?= isset($conteudo['Titulo']) ? $conteudo['Titulo'] : set_value('Titulo') ?&gt;
isset significa se tiver conteúdo vindo do banco imprime, senão set_value apresenta o conteúdo
do campo no formulário*/
?&gt;" /&gt;</pre
```

# Chamada nos botões editar e excluir

```
Routes.php x Home.php — Controllers x ConteudoGravar.php x ConteudoItem.php x ConteudoModel.php x HomeAluno.php x Home.php — Views\pages x
<div class="container">
  <a href="<?php echo '/inserir' ?>" class="btn btn-primary">Adicionar Conteúdo</a>
  <?php if(!empty($conteudo)):?>
    <?php foreach($conteudo as $conteudo_item):?>
      <div class="card my-5">
        <div class="card-body">
          <h3><?php echo $conteudo_item['Titulo'];?></h3>
          <p><?php echo $conteudo_item['Texto_curto'];?></p>
        </div>
        <div class="card-footer">
          <a href="<?= $conteudo_item['ID_Conteudo']?>" class="btn btn-success">Saiba mais</a>
          <a href="<?= '/editar/' . $conteudo_item['ID_Conteudo']?>" class="btn btn-warning">Editar
          </a>
          <a href="<?= '/excluir/' . $conteudo_item['ID_Conteudo']?>" class="btn btn-danger">
            Excluir</a>
        </div>
      </div>
    </?php>
  </?php>
</div>
```

# Rotas para editar e excluir



The screenshot shows a code editor with four tabs: 'Routes.php', 'Home.php — Controllers', 'ConteudoGravar.php', and 'ConteudoItem.php'. The 'Routes.php' tab is active, displaying PHP code for route definitions. The code includes comments and several route matching and getting statements for a Laravel-style framework.

```
68  * -----
69  * Route Definitions
70  * -----
71  */
72
73  // We get a performance increase by specifying the default
74  // route since we don't have to scan directories.
75  $routes->match(['get', 'post'], 'inserir', 'Home::inserir');
76  $routes->match(['get', 'post'], 'gravar', 'Home::gravar');
77  $routes->match(['get', 'post'], 'editar/(:num)', 'Home::editar/$1');
78  $routes->match(['get', 'post'], 'excluir/(:num)', 'Home::excluir/$1');
79
80  $routes->get('/', 'Home::index');
81  $routes->get('aluno', 'Home::aluno');
82  $routes->get('/:segment)', 'Home::item/$1');
```