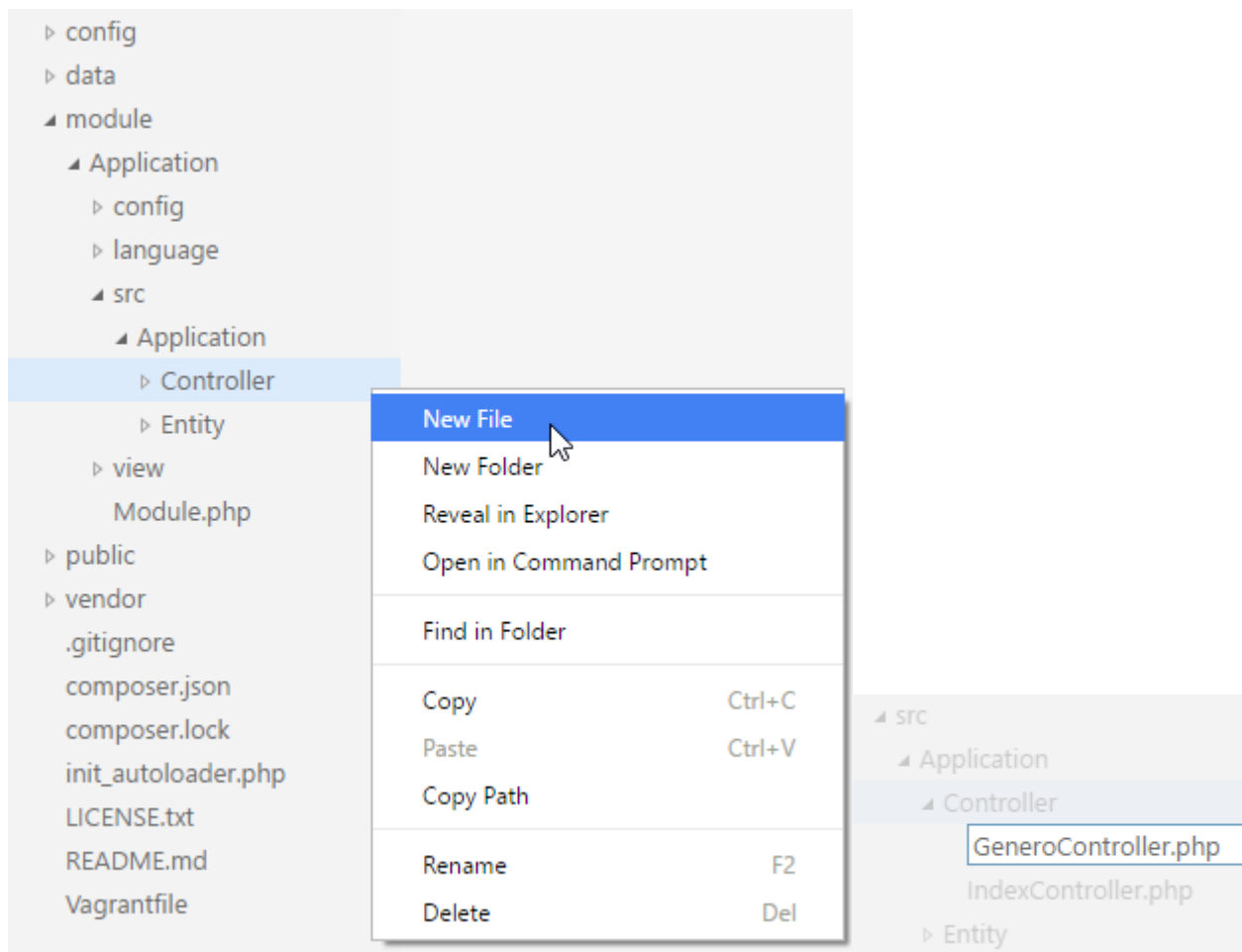


DSII- Projeto Final

Parte 4 – Implementando o CRUD

Criando os Controladores

Crie um arquivo chamado “GeneroController” no diretório “Controller” com o conteúdo a seguir.



Estrutura básica da classe

```
GeneroController.php module\Application\src\Application\Controller
1  <?php
2  namespace Application\Controller;
3
4  use Zend\Mvc\Controller\AbstractActionController;
5  use Zend\View\Model\ViewModel;
6  use Application\Entity\Genero;
7
8  class GeneroController extends AbstractActionController
9  {
10
11 }
```

O uso definido na linha 4 refere-se à classe “AbstractActionController” da qual nossa classe estenderá como declarado na linha 8. A linha 5 declara o uso da classe ViewModel que será utilizada para renderizar as views. Observe que também carregamos as classes de modelos que serão manipuladas pelo controlador.

Facilitando o acesso à dados

```
GeneroController.php module\Application\src\Application\Controller
1  <?php
2  namespace Application\Controller;
3
4  use Zend\Mvc\Controller\AbstractActionController;
5  use Zend\View\Model\ViewModel;
6  use Application\Entity\Genero;
7
8  class GeneroController extends AbstractActionController
9  {
10     protected $entityManager;
11
12     public function getEntityManager()
13     {
14         if (!$this->entityManager) {
15             $this->entityManager = $this->getServiceLocator()->get('Doctrine\ORM\EntityManager');
16         }
17         return $this->entityManager;
18     }
19 }
```

A linha 10 declara um atributo que conterá o gerenciador de entidades responsável por mapear as entidades no banco.

O método contido nas linhas 12 à 18 instancia a Classe EntityManager do doctrine armazenando-a no atributo recém-criado.

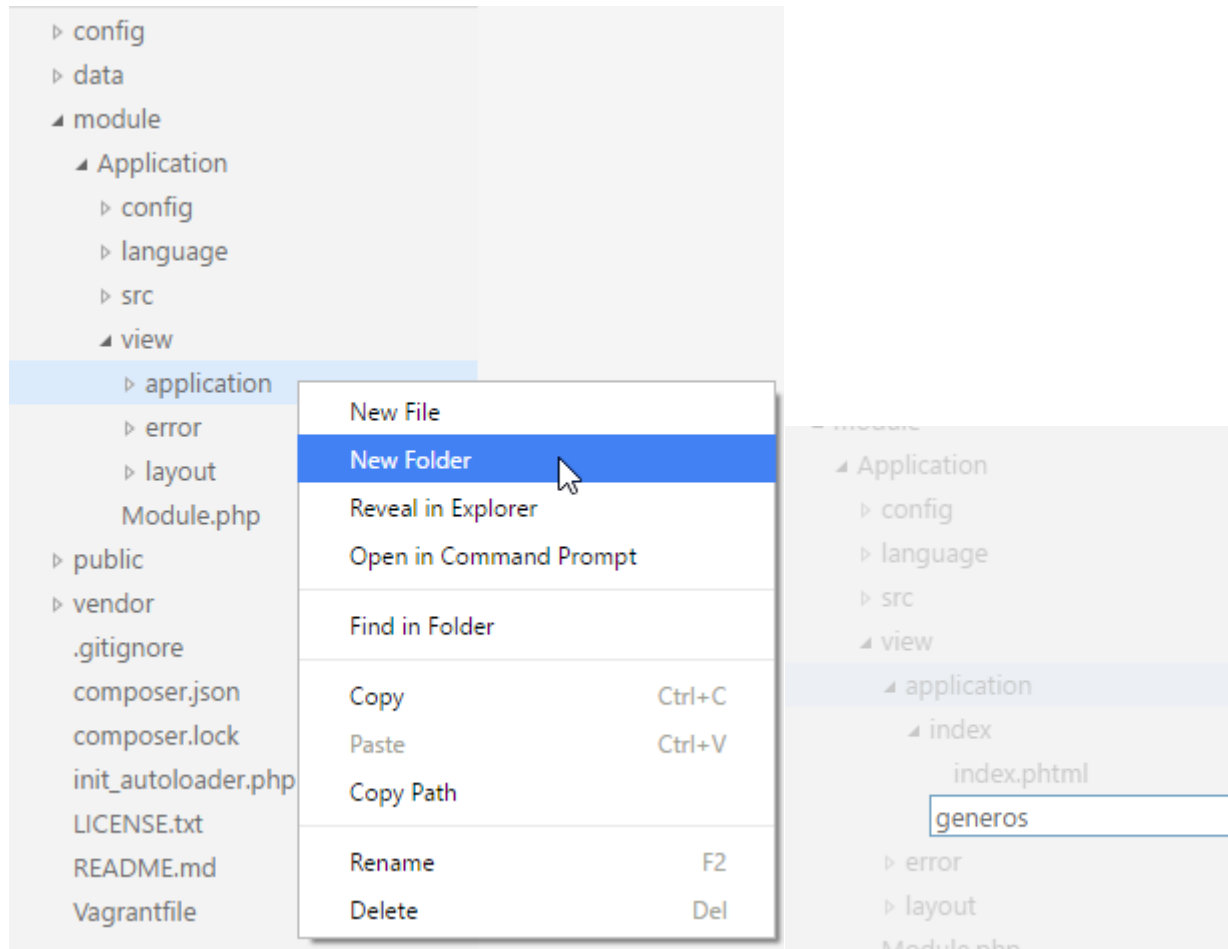
Listando os gêneros

```
GeneroController.php module\Application\src\Application\Controller
4  use Zend\Mvc\Controller\AbstractActionController;
5  use Zend\View\Model\ViewModel;
6  use Application\Entity\Genero;
7
8  class GeneroController extends AbstractActionController
9  {
10     protected $entityManager;
11
12     public function getEntityManager()
13     {
14         if (!$this->entityManager) {
15             $this->entityManager = $this->getServiceLocator()->get('Doctrine\ORM\EntityManager');
16         }
17         return $this->entityManager;
18     }
19
20     public function indexAction()
21     {
22         $generos = $this->getEntityManager()
23             ->getRepository('\Application\Entity\Genero')->findAll();
24         return new ViewModel(array('generos' => $generos));
25     }
26 }
27
28
```

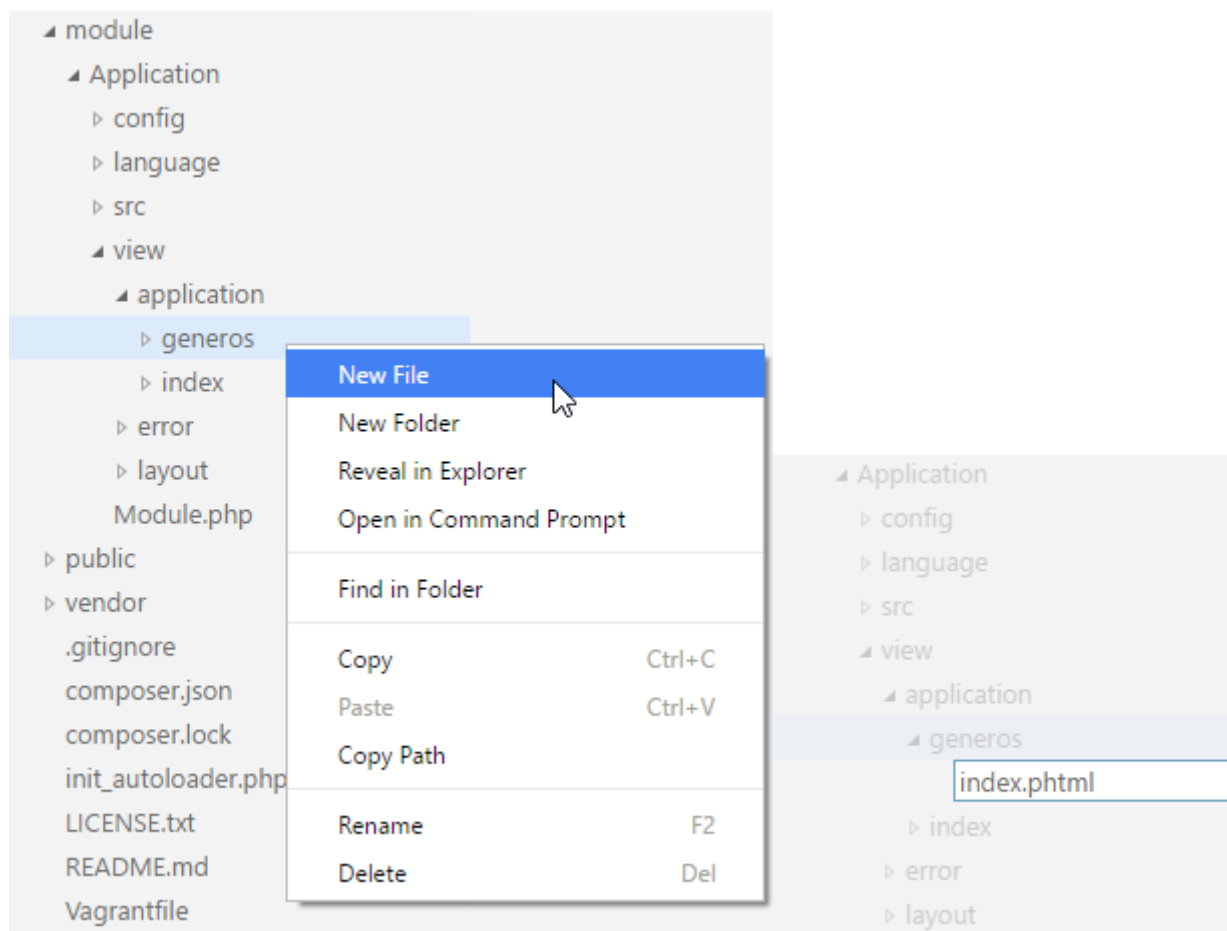
Observe que utilizamos o método “findAll” do repositório retornado pelo “entityManager” para guardar na variável “generos” uma coleção contendo todos os registros da tabela “generos” (linha 22). Na linha 24 chamamos a view passando a variável “generos” como parâmetro.

Criando as visualizações

Crie um diretório com o nome do controlador na pasta view/application:



Crie um arquivo phtml com o nome da action na pasta recém criada:



```
index.phtml module\Application\view\application\generos
1  <?php
2  $title = "Gêneros";
3  $this->headTitle($title);
4  ?>
5  <h1><?= $this->escapeHtml($title) ?></h1>
6
7  <table class="table">
8      <tr>
9          <th>Id</th>
10         <th>Descrição</th>
11     </tr>
12     <?php foreach ($generos as $genero): ?>
13         <tr>
14             <td><?= $this->escapeHtml($genero->Id) ?></td>
15             <td><?= $this->escapeHtml($genero->Descricao) ?></td>
16         </tr>
17     <?php endforeach; ?>
18 </table>
```

Torne o controlador “Genero” acessível via url acrescentando a alinha destacada abaixo no arquivo “module\Application\config\module.config.php”

```

module.config.php module\Application\config
134         ),
135     ),
136     'controllers' => array(
137         'invokables' => array(
138             'Application\Controller\Index' => Controller\IndexController::class,
139             'Application\Controller\Genero' => Controller\GeneroController::class
140         ),
141     ),

```

Execute a aplicação.

```
..\php\php -S localhost:80 -t public
```

```

C:\> Selecionar Prompt de Comando - ..\php\php -S localhost:80 -t public

C:\Users\marco\Desktop\Trabalho\DSII-Workspace\marcocarvalho>..\php\php -S localhost:80 -t public
PHP 7.0.6 Development Server started at Thu Jun  9 16:46:38 2016
Listening on http://localhost:80
Document root is C:\Users\marco\Desktop\Trabalho\DSII-Workspace\marcocarvalho\public
Press Ctrl-C to quit.

```

Acesse “localhost/application/genero” no navegador para visualizar a exibição dos gêneros.



Gêneros

Id	Descrição

© 2005 - 2016 by Zend Technologies Ltd. All rights reserved.

No MySQL Command Line Client, adicione manualmente 3 registros à tabela gêneros como mostrado abaixo:

```
Selecionar MySQL 5.7 Command Line Client
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 5.7.12-log MySQL Community Server (GPL)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SHOW TABLES FROM marcocarvalho_db;
+-----+
| Tables_in_marcocarvalho_db |
+-----+
| autores                     |
| editoras                   |
| generos                     |
| livros                     |
| livros_possuem_autores      |
| livros_possuem_categorias   |
+-----+
6 rows in set (0.00 sec)

mysql> INSERT INTO marcocarvalho_db.generos(descricao) VALUES('Ação'),('Aventura'),('Terror');
```

Recarregue a página de gêneros no navegador e veja que a consulta na base de dados já está sendo realizada.

localhost/application/genero

Skeleton Application

Home

Gêneros

Id	Descrição
1	Ação
2	Aventura
3	Terror

© 2005 - 2016 by Zend Technologies Ltd. All rights reserved.

Preparando a rota para os gêneros

module.config.php module\Application\config

```
11
12 return array(
13     'doctrine' => array(
14         'driver' => array(
15             __NAMESPACE__ . '_driver' => array(
16                 'class' => 'Doctrine\ORM\Mapping\Driver\AnnotationDriver',
17                 'cache' => 'array',
18                 'paths' => array(__DIR__ . '/../src/' . __NAMESPACE__ . '/Entity')
19             ),
20             'orm_default' => array(
21                 'drivers' => array(
22                     __NAMESPACE__ . '\Entity' => __NAMESPACE__ . '_driver'
23                 )
24             )
25         )
26     ),
27     'router' => array(
28         'routes' => array(
29             'genero' => array(
30                 'type' => 'segment',
31                 'options' => array(
32                     'route' => '/genero[/:action][/id]',
33                     'constraints' => array(
34                         'action' => '[a-zA-Z][a-zA-Z0-9_-]*',
35                         'id' => '[0-9]+',
36                     ),
37                     'defaults' => array(
38                         'controller' => 'Application\Controller\Genero',
39                         'action' => 'index',
40                     ),
41                 ),
42             ),
43         ),
44     ),
45     'home' => array(
46         'type' => 'Zend\Mvc\Router\Http\Literal',
47         'options' => array(
```

Agora pode acessar os gêneros digitando somente “localhost/genero” na barra de endereços do navegador: