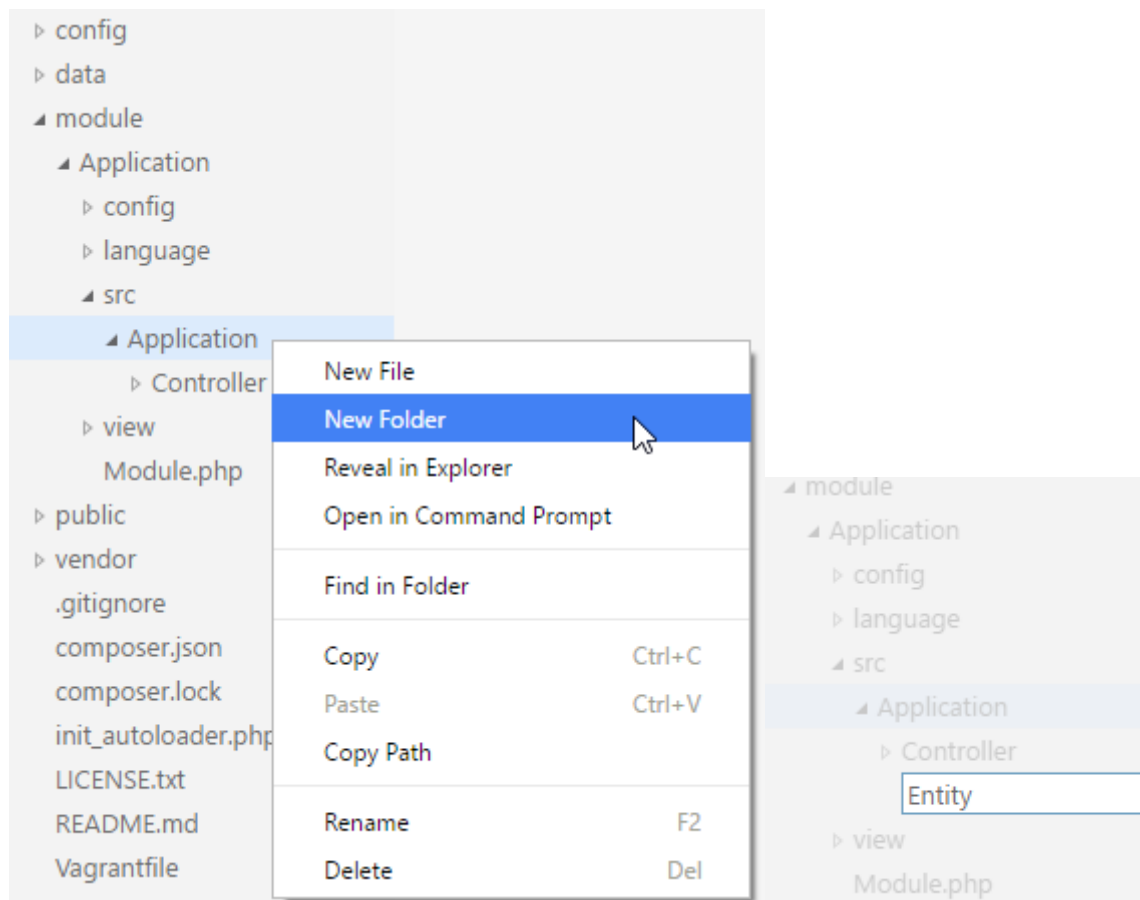


DSII- Projeto Final

Parte 2 – Criação das Entidades

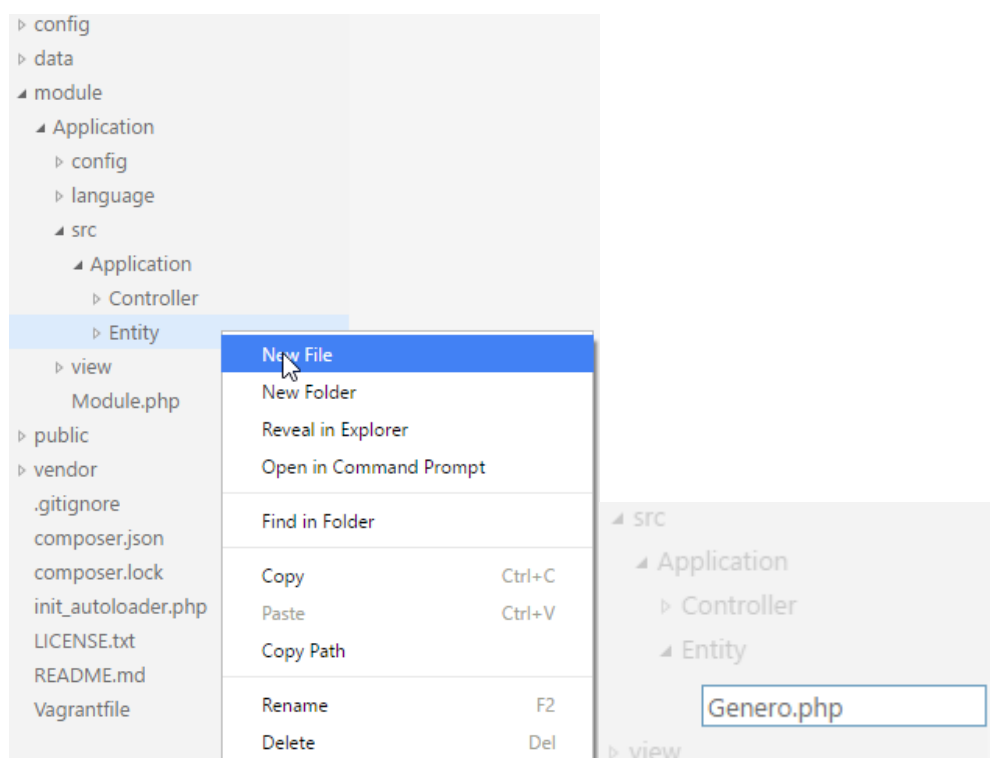
Diretório de Entidades

Crie um diretório chamado “Entity” no módulo “application” (module/Application/src/Application):



Entidade Gênero

Crie um arquivo chamado “Genero.php” no diretório “Entity” recém-criado com o conteúdo listado a seguir.



Genero.php module\Application\src\Application\Entity

```
1  <?php
2  namespace Application\Entity;
3
4  use Doctrine\ORM\Mapping as ORM;
5  use Doctrine\Common\Collections\ArrayCollection;
6
7  /**
8   * @ORM\Entity
9   * @ORM\Table(name="generos")
10  */
11  class Genero
12  {
13      /**
14       * @ORM\Id
15       * @ORM\GeneratedValue(strategy="AUTO")
16       * @ORM\Column(type="integer")
17      */
18      protected $id;
19
20      /**
21       * @ORM\Column(type="string", length=50, unique=TRUE)
22      */
23      protected $descricao;
24
25      /**
26       * @ORM\OneToMany(targetEntity="Livro", mappedBy="genero")
27      */
28      protected $livros;
29
30      // Construtor
31      public function __construct()
32      {
33          $this->livros = new ArrayCollection();
34      }
35
36      // Getters & Setters
37      public function addLivro(Livro $livro)
38      {
39          if (!$this->livros->contains($livro)) {
40              $this->livros->add($livro);
41          }
42          return $this;
43      }
44
45      public function removeLivro($livro)
46      {
47          if ($this->livros->contains($livro)) {
48              $this->livros->removeElement($livro);
49          }
50          return $this;
51      }
52
53      public function __get($name)
54      {
55          return $this->$name;
56      }
57
58      public function __set($name, $value)
59      {
60          $this->$name = $value;
61      }
62  }
```

Detalhamento do Código

Define o namespace da qual a classe “Genero” faz parte

```
2 namespace Application\Entity;
```

Declara que a classe que está sendo criada utilizará componentes dos namespaces “Doctrine\ORM\Mapping”, para realizar o mapeamento objeto relacional e “Doctrine\Common\Collections\ArrayCollection”, para coleções de dados.

```
4 use Doctrine\ORM\Mapping as ORM;
5 use Doctrine\Common\Collections\ArrayCollection;
```

Declara a classe “Genero” definindo (linha 9) que a entidade faz referência à tabela “generos” do banco de dados.

```
7 /**
8  * @ORM\Entity
9  * @ORM\Table(name="generos")
10 */
11 class Genero
12 {
```

Define o atributo “id” anotando-o como chave primária (linha 14), auto incremento (linha 15) e do tipo inteiro (linha 16).

```
13 /**
14  * @ORM\Id
15  * @ORM\GeneratedValue(strategy="AUTO")
16  * @ORM\Column(type="integer")
17 */
18 protected $id;
```

Cria o atributo “descricao” sendo do tipo string com tamanho 50 e único (linha 21).

```
20 /**
21  * @ORM\Column(type="string", Length=50, unique=TRUE)
22 */
23 protected $descricao;
```

Declara o atributo “livros” como sendo uma coleção de elementos do tipo “Livro” cujo relacionamento está mapeado no atributo “generos” da classe “Livro” (Linha26).

```
25 /**
26  * @ORM\ManyToMany(targetEntity="Livro", mappedBy="generos")
27 */
28 protected $livros;
```

Declara o método construtor da classe inicializando o atributo “livros” como sendo um “ArrayCollection”.

```
30 // Construtor
31 public function __construct()
32 {
33     $this->livros = new ArrayCollection();
34 }
```

Define o método “addLivro” responsável por adicionar um objeto do tipo “Livro” à coleção “livros”.

```
37     public function addLivro(Livro $livro)
38     {
39         if (!$this->livros->contains($livro)) {
40             $this->livros->add($livro);
41         }
42         return $this;
43     }
```

Declara o método “removeLivro” responsável por excluir um objeto do tipo “Livro” da coleção “livros”.

```
45     public function removeLivro($livro)
46     {
47         if ($this->livros->contains($livro)) {
48             $this->livros->removeElement($livro);
49         }
50         return $this;
51     }
```

Expõe os atributos da classe para que sejam acessíveis por outros objetos.

```
53     public function __get($name)
54     {
55         return $this->$name;
56     }
57
58     public function __set($name, $value)
59     {
60         $this->$name = $value;
61     }
```

Entidade Autor

Autor.php module\Application\src\Application\Entity

```
1  <?php
2  namespace Application\Entity;
3
4  use Doctrine\ORM\Mapping as ORM;
5  use Doctrine\Common\Collections\ArrayCollection;
6
7  /**
8   * @ORM\Entity
9   * @ORM\Table(name="autores")
10  */
11  class Autor
12  {
13      /**
14       * @ORM\Id
15       * @ORM\GeneratedValue(strategy="AUTO")
16       * @ORM\Column(type="integer")
17       */
18      protected $id;
19
20      /** @ORM\Column(type="string", Length=100) */
21      protected $nome;
22
23      /** @ORM\ManyToMany(targetEntity="Livro", mappedBy="autores") */
24      protected $livros;
25
26      // Construtor
27      public function __construct()
28      {
29          $this->livros = new ArrayCollection();
30      }
31
32      // Getters & Setters
33      public function addLivro(Livro $livro)
34      {
35          if (!$this->livros->contains($livro)) {
36              $this->livros->add($livro);
37          }
38          return $this;
39      }
40
41      public function removeLivro($livro)
42      {
43          if ($this->livros->contains($livro)) {
44              $this->livros->removeElement($livro);
45          }
46          return $this;
47      }
48
49      public function __get($name)
50      {
51          return $this->$name;
52      }
53
54      public function __set($name, $value)
55      {
56          $this->$name = $value;
57      }
58  }
```

Entidade Livro

```
Livro.php module\Application\src\Application\Entity
1  <?php
2  namespace Application\Entity;
3
4  use Doctrine\ORM\Mapping as ORM;
5  use Application\Entity\Genero;
6  use Doctrine\Common\Collections\ArrayCollection;
7
8  /**
9   * @ORM\Entity
10   * @ORM\Table(name="Livros")
11   */
12  class Livro
13  {
14      /**
15       * @ORM\Id
16       * @ORM\GeneratedValue(strategy="AUTO")
17       * @ORM\Column(type="integer")
18       */
19      protected $id;
20
21      /** @ORM\Column(type="string", Length=100) */
22      protected $titulo;
23
24      /**
25       * @ORM\ManyToOne(targetEntity="Genero", inversedBy="Livros")
26       * @ORM\JoinColumn(name="genero_id", referencedColumnName="id")
27       */
28      protected $genero;
29
30      /**
31       * @ORM\ManyToMany(targetEntity="Autor", inversedBy="Livros")
32       * @ORM\JoinTable(name="Livros_posuem_autores")
33       */
34      protected $autores;
35
36      public function __construct()
37      {
38          $this->autores = new ArrayCollection();
39      }
40
```

(.... Continua na próxima página)

```

41 // Getters & Setters
42
43 public function addAutor(Autor $autor)
44 {
45     if (!$this->autores->contains($autor)) {
46         $this->autores->add($autor);
47         $autor->addLivro($this);
48     }
49     return $this;
50 }
51
52 public function removeAutor($autor)
53 {
54     if ($this->autores->contains($autor)) {
55         $this->autores->removeElement($autor);
56         $autor->removeLivro($this);
57     }
58     return $this;
59 }
60
61 public function setGenero($genero)
62 {
63     if($genero === NULL) {
64         if($this->genero !== NULL) {
65             $this->genero->removeLivro($this);
66         }
67         $this->genero = NULL;
68     } else {
69         if($this->genero !== NULL) {
70             $this->genero->removeLivro($this);
71         }
72         $this->genero = $genero;
73         $this->genero->addLivro($this);
74     }
75 }
76
77 public function __get($name)
78 {
79     if(method_exists($this, 'get'.ucfirst($name))) {
80         $method = 'get'.ucfirst($name);
81         return $this->$method();
82     }
83     return $this->$name;
84 }
85
86 public function __set($name, $value)
87 {
88     if(method_exists($this, 'set'.ucfirst($name))) {
89         $method = 'set'.ucfirst($name);
90         $this->$method($value);
91         return $this;
92     }
93     $this->$name = $value;
94     return $this;
95 }
96 }

```

Referências

ZEND. Zend\Form Disponível em < <http://framework.zend.com/manual/current/en/modules/zend.form.quick-start.html>>

DOCTRINE. Doctrine 2 Documentation: Basic Mapping Disponível em < <http://doctrine-orm.readthedocs.io/projects/doctrine-orm/en/latest/reference/basic-mapping.html>>

DOCTRINE. Doctrine 2 Documentation: Association Mapping Disponível em < <http://doctrine-orm.readthedocs.io/projects/doctrine-orm/en/latest/reference/association-mapping.html>>