

Consolidando o seu conhecimento

Chegou a hora de você pôr em prática o que foi visto na aula. Para isso, execute os passos listados abaixo.

1) No arquivo de mapeamento da classe `Ator`, adicione a seguinte *tag* em seu relacionamento (*many-to-many*):

```
<cascade>
  <cascade-persist />
</cascade>
```

2) No arquivo `teste.php`, remova as seguintes linhas:

```
$em->persist($filme1);
$em->persist($filme2);
```

3) Execute os comandos `vendor\bin\doctrine orm:schema-tool:drop --force` e `vendor\bin\doctrine orm:schema-tool:create` para excluir e criar o *schema* do banco de dados novamente.

4) Execute mais uma vez `php teste.php` e garanta que os filmes continuam sendo salvos no banco de dados.

5) Na pasta **src/Entity**, crie o arquivo `Idioma.php`, com o seguinte conteúdo:

```
<?php

namespace Alura\Doctrine\Entity;

class Idioma
{
    private $id;
    private $nome;
    private $ultimaAtualizacao;

    public function __construct(?int $id, string $nome)
    {
        $this->id = $id;
        $this->nome = $nome;
        $this->ultimaAtualizacao = new \DateTimeImmutable();
    }

    public function __toString(): string
    {
        return $this->nome;
    }
}
```

6) Na pasta **mapeamentos**, crie o arquivo `Alura\Doctrine\Entity\Idioma.dcm.xml`, com o seguinte conteúdo:

```
<doctrine-mapping>
  <entity name="Alura\Doctrine\Entity\Idioma" table="idioma">
```

```

<entity name="Alura\Doctrine\Entity\Idioma" table="idioma">
    <id name="id" column="id_idioma" type="smallint">
        <generator strategy="IDENTITY" />
    </id>
    <field name="nome" type="string" length="20">
        <options>
            <option name="fixed">true</option>
        </options>
    </field>
    <field name="ultimaAtualizacao" column="data_atualizado" type="datetime">
        <options>
            <option name="default">NOW()</option>
        </options>
    </field>
</entity>
</doctrine-mapping>

```

7) Edite o construtor da class `Filme` para receber os idiomas:

```

// ...
private $idiomaAudio;
private $idiomaOriginal;
// ...

public function __construct(
    ?int $id,
    string $titulo,
    Idioma $idiomaAudio,
    Idioma $idiomaOriginal,
    ?string $sinopse = null,
    ?string $anoLancamento = null
) {
    $this->id = $id;
    $this->titulo = $titulo;
    $this->idiomaAudio = $idiomaAudio;
    $this->idiomaOriginal = $idiomaOriginal;
    $this->sinopse = $sinopse;
    $this->anoLancamento = $anoLancamento;
    $this->ultimaAtualizacao = new \DateTimeImmutable();
    $this->atores = new ArrayCollection();
}

```

8) No mapeamento da classe `Filme`, adicione o seguinte relacionamento:

```

<many-to-one field="idiomaOriginal"
    target-entity="Alura\Doctrine\Entity\Idioma">
    <cascade>
        <cascade-persist/>
    </cascade>

    <join-column name="id_idioma_original" nullable="false" referenced-column-name="id_idioma" />
</many-to-one>

<many-to-one field="idiomaAudio"
    target-entity="Alura\Doctrine\Entity\Idioma">

```

```

        <target-entity="Alura\Doctrine\Entity\Idioma" />
    </cascade>
    <cascade-persist/>
</cascade>

    <join-column name="id_idioma_audio" nullable="false" referenced-column-name="id_idioma" />
</many-to-one>

```

9) Edite o arquivo `teste.php` para ter o seguinte conteúdo:

```

use Alura\Doctrine\Entity\Ator;
use Alura\Doctrine\Entity\Filme;
use Alura\Doctrine\Entity\Idioma;

require_once 'vendor/autoload.php';

$em = (new \Alura\Doctrine\Helper\EntityManagerCreator())->criaEntityManager();

$ator = new Ator(null, 'Vinicius', 'Dias');

$portugues = new Idioma(null, 'Português');
$salemao = new Idioma(null, 'Alemão');
$ingles = new Idioma(null, 'Inglês');

$filme1 = new Filme(null, 'A volta dos que não foram', $portugues, $salemao);
$filme2 = new Filme(null, 'As longas tranças do careca', $portugues, $ingles);

$ator->addFilme($filme1);
$ator->addFilme($filme2);

$em->persist($ator);
$em->flush();

```

10) Execute os comandos `vendor/bin/doctrine orm:schema-tool:drop --force` e `vendor/bin/doctrine orm:schema-tool:create` para excluir e criar o *schema* do banco de dados novamente.

11) Após criar o *schema* do banco, execute `php teste.php`. Garanta que os filmes agora têm seus idiomas corretamente relacionados no banco de dados.

12) Adicione os seguintes métodos na classe `Titulo`:

```

public function getTitulo(): string
{
    return $this->titulo;
}

public function getIdiomaAudio(): Idioma
{
    return $this->idiomaAudio;
}

public function getAtores(): array
{
    return $this->atores->map(function (Ator $ator) {

```

```
        return $ator->getNome();  
    })->toArray();  
}
```

13) Crie e execute o arquivo `busca.php`, com o seguinte conteúdo:

```
<?php  
use Alura\Doctrine\Entity\Filme;  
  
require_once 'vendor/autoload.php';  
  
$em = (new \Alura\Doctrine\Helper\EntityManagerCreator())->criaEntityManager();  
  
/** @var Filme[] $filmes */  
$filmes = $em->getRepository(Filme::class)->findAll();  
foreach ($filmes as $filme) {  
    echo $filme->getTitulo() . PHP_EOL . 'Idioma: ' . $filme->getIdiomaAudio();  
    echo PHP_EOL;  
    echo PHP_EOL;  
    echo implode(', ', $filme->getAtores());  
}
```