

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

```
<?php

namespace Alura\Doctrine\Entity;

abstract class Pessoa
{
    protected $id;
    protected $primeiroNome;
    protected $ultimoNome;
    protected $ultimaAtualizacao;

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

    public function getNome(): string
    {
        return $this->primeiroNome . ' ' . $this->ultimoNome;
    }
}
```

2) Na mesma pasta, crie a classe `Cliente` :

```
<?php

namespace Alura\Doctrine\Entity;

class Cliente extends Pessoa
{
}
```

3) Agora faça com que a classe `Ator` herde de `Pessoa` :

```
<?php

namespace Alura\Doctrine\Entity;
use Doctrine\Common\Collections\ArrayCollection;
```

```

class Ator extends Pessoa
{
    private $filmes;
    public function __construct(
        ?int $id,
        string $primeiroNome,
        string $ultimoNome
    ) {
        parent::__construct($id, $primeiroNome, $ultimoNome);
        $this->filmes = new ArrayCollection();
    }

    public function addFilme(Filme $filme): void
    {
        if ($this->filmes->contains($filme)) {
            return;
        }
        $this->filmes->add($filme);
        $filme->addAtor($this);
    }

    public function quantidadeFilmes(): int
    {
        return $this->filmes->count();
    }
}

```

4) Hora do mapeamento! Crie o arquivo **mapeamentos/Ator.Doctrine.Entity.Cliente.dcm.xml**:

```

<doctrine-mapping xmlns="http://doctrine-project.org/schemas/orm/doctrine-mapping">
    <entity name="Alura\Doctrine\Entity\Cliente" table="cliente">
        <indexes>
            <index columns="ultimo_nome" name="idx_ultimo_nome_cliente" />
        </indexes>

        <unique-constraints>
            <unique-constraint columns="primeiro_nome,ultimo_nome"
                name="idx_nome_completo_cliente_unique" />
        </unique-constraints>

        <id name="id" column="id_cliente" type="integer">
            <generator strategy="IDENTITY" />
        </id>
    </entity>
</doctrine-mapping>

```

5) Altere o mapeamento de Ator , removendo as propriedades que também existem em Cliente :

```

<doctrine-mapping xmlns="http://doctrine-project.org/schemas/orm/doctrine-mapping">
    <entity name="Alura\Doctrine\Entity\Ator"
        table="ator"
        repository-class="Alura\Doctrine\Repository\RepositorioAtores">
        .
        .
    </entity>
</doctrine-mapping>

```

```

<indexes>
  <index columns="ultimo_nome" name="idx_ultimo_nome" />
</indexes>

<unique-constraints>
  <unique-constraint columns="primeiro_nome,ultimo_nome"
    name="idx_nome_completo_unique" />
</unique-constraints>

<id name="id" column="id_ator" type="integer">
  <generator strategy="IDENTITY" />
</id>

<many-to-many field="filmes"
  inversed-by="atores"
  target-entity="Alura\Doctrine\Entity\Filme">
  <cascade>
    <cascade-persist/>
  </cascade>
  <join-table name="ator_filme">
    <join-columns>
      <join-column name="id_ator" referenced-column-name="id_ator" />
    </join-columns>
    <inverse-join-columns>
      <join-column name="id_filme" referenced-column-name="id_filme"/>
    </inverse-join-columns>
  </join-table>
</many-to-many>
</entity>
</doctrine-mapping>

```

6) Finalmente, crie o mapeamento da super-classe Pessoa :

```

<doctrine-mapping xmlns="http://doctrine-project.org/schemas/orm/doctrine-mapping">
  <mapped-superclass name="Alura\Doctrine\Entity\Pessoa">
    <field name="primeiroNome"
      column="primeiro_nome"
      type="string"
      length="45" />

    <field name="ultimoNome"
      column="ultimo_nome"
      type="string"
      length="45" />

    <field name="ultimaAtualizacao"
      column="data_atualizado"
      type="datetime">
      <options>
        <option name="default">NOW()</option>
      </options>
    </field>
  </mapped-superclass>
</doctrine-mapping>

```

