

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) Atualize o arquivo **teste.php**, para que ele tenha o seguinte conteúdo:

```
<?php

use Alura\Doctrine\Entity\Ator;
use Alura\Doctrine\Entity\ClassificacaoEnum;
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');
$atriz = new Ator(null, 'Maria', 'da Silva');

$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, null, null, ClassificacaoEnum::FILME);
$filme2 = new Filme(null, 'As longas tranças do careca', $portugues, $ingles, null, null, ClassificacaoEnum::FILME);

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

$filme1->addAtor($atriz);

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

$em->flush();
```

2) Execute os comandos `vendor/bin/doctrine orm:schema-tool:drop --force` e `vendor/bin/doctrine orm:schema-tool:create`, para recriar o banco de dados do zero.

3) Execute `php teste.php` para inserir os filmes e atores no banco.

4) Crie (e mapeie no arquivo de configuração do mapeamento da entidade) um novo repositório de atores, com o seguinte conteúdo:

```
<?php

namespace Alura\Doctrine\Repository;

use Alura\Doctrine\Entity\Ator;
use Doctrine\ORM\EntityRepository;
```

```

use Doctrine\ORM\Query\ResultSetMapping;

class RepositorioAtores extends EntityRepository
{
    public function buscaAtoresMaisAtuantesOld()
    {
        return $this->createQueryBuilder('a')
            ->join('a.filmes', 'f')
            ->addSelect('f')
            ->getQuery()
            ->getResult();
    }

    public function buscaAtoresMaisAtuantes()
    {
        $sql = 'SELECT CONCAT(ator.primeiro_nome, \' \' , ator.ultimo_nome) AS nome,
                        COUNT(filme.id_filme) qtd_filmes
                FROM ator
                JOIN ator_filme ON ator_filme.id_ator = ator.id_ator
                JOIN filme ON filme.id_filme = ator_filme.id_filme
                GROUP BY ator.id_ator
                LIMIT 2;';
        $rsm = new ResultSetMapping();
        $rsm->addScalarResult('nome', 'nome');
        $rsm->addScalarResult('qtd_filmes', 'qtdFilmes');

        return $this->getEntityManager()
            ->createNativeQuery($sql, $rsm)
            ->getResult();
    }

    public function buscaTodosAtores()
    {
        $sql = 'SELECT id_ator, primeiro_nome, ultimo_nome FROM ator;';
        $rsm = new ResultSetMapping();
        $rsm->addEntityResult(Ator::class, 'ator');
        $rsm->addFieldResult('ator', 'id_ator', 'id');
        $rsm->addFieldResult('ator', 'primeiro_nome', 'primeiroNome');
        $rsm->addFieldResult('ator', 'ultimo_nome', 'ultimoNome');

        return $this->getEntityManager()
            ->createNativeQuery($sql, $rsm)
            ->getResult();
    }
}

```

5) Altere o arquivo **busca.php**, para ter o seguinte conteúdo:

```

<?php

use Alura\Doctrine\Entity\Ator;
use Alura\Doctrine\Helper\EntityManagerCreator;
use Alura\Doctrine\Repository\RepositorioAtores;
use Doctrine\DBAL\Logging\DebugStack;

require_once 'vendor/autoload.php';

```

```

$debug = new DebugStack();

$em = (new EntityManagerCreator())->criaEntityManager();
$em->getConfiguration()->setSQLLogger($debug);

/** @var RepositorioAtores $repositorioAtores */
$repositorioAtores = $em->getRepository(Ator::class);

$atoresMaisAtuantes = $repositorioAtores->buscaAtoresMaisAtuantes();

foreach ($atoresMaisAtuantes as $ator) {
    echo "O(a) ator/atriz {$ator['nome']} atuou em {$ator['qtdFilmes']} filmes" . PHP_EOL;
}

foreach ($debug->queries as $query) {
    var_dump($query);
}

```

6) Execute `php busca.php` e verifique que o resultado está correto. Verifique também que a *query* executada foi a desejada.

7) Crie agora o arquivo **native.php**, com o seguinte conteúdo:

```

<?php

use Alura\Doctrine\Helper\EntityManagerCreator;
use Doctrine\DBAL\Logging\DebugStack;

require_once 'vendor/autoload.php';

$em = (new EntityManagerCreator())->criaEntityManager();
$debug = new DebugStack();
$em->getConfiguration()->setSQLLogger($debug);

/** @var \Alura\Doctrine\Repository\RepositorioAtores $repositorio */
$repositorio = $em->getRepository(\Alura\Doctrine\Entity\Ator::class);

$atores = $repositorio->buscaTodosAtores();

foreach ($atores as $ator) {
    echo "Ator {$ator->getNome()}" . PHP_EOL;
}

foreach ($debug->queries as $query) {
    var_dump($query);
}

```

8) Execute `php native.php` e analise o resultado.

9) Altere a conexão (no arquivo **EntityManagerCreator.php**) para se conectar ao banco real (`alura_filmes`).

10) Troque o SQL do método `buscaAtoresMaisAtuantes` para `SELECT * FROM atores_mais_atuantes` .

11) Execute novamente `php busca.php` e garanta que tudo continua funcionando.