

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) Acesse <https://www.postgresql.org/download/> (<https://www.postgresql.org/download/>) e siga as instruções para instalar e configurar o **PostgreSQL** em sua máquina.

2) Crie uma nova pasta para o projeto, e dentro dela, execute o seguinte comando:

```
composer require doctrine/orm
```

3) Crie, na pasta **src/Helper** o arquivo `EntityManagerCreator.php`, com o seguinte conteúdo:

```
<?php

namespace Alura\Doctrine\Helper;

use Doctrine\ORM\EntityManager;
use Doctrine\ORM\EntityManagerInterface;
use Doctrine\ORM\Tools\Setup;

class EntityManagerCreator
{
    public function criaEntityManager(): EntityManagerInterface
    {
        $config = Setup::createAnnotationMetadataConfiguration(
            [__DIR__ . '/../Entity']
        );
        $con = [
            'driver' => 'pdo_pgsql',
            'host' => 'localhost',
            'dbname' => 'alura_filmes',
            'user' => 'postgres',
            'password' => 'senhalura',
        ];

        return EntityManager::create($con, $config);
    }
}
```

4) Crie agora, na pasta do projeto, o arquivo `cli-config.php`, com o seguinte conteúdo:

```
<?php

require_once __DIR__ . '/vendor/autoload.php';

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

```
return \Doctrine\ORM\Tools\Console\ConsoleRunner::createHelperSet($entityManager);
```

5) No arquivo `composer.json`, logo após a chave `require`, adicione a chave `autoload`:

```
"autoload": {  
    "psr-4": {  
        "Alura\\Doctrine\\": "src/"  
    }  
}
```

6) Crie, na pasta **src/Entity**, um arquivo chamado `Ator.php`, com o seguinte conteúdo:

```
<?php  
  
namespace Alura\Doctrine\Entity;  
  
/**  
 * @Entity  
 * @Table(name="ator")  
 */  
class Ator  
{  
    /**  
     * @Id  
     * @GeneratedValue  
     * @Column(name="id_ator", type="integer")  
     */  
    private $id;  
    /**  
     * @Column(name="primeiro_nome", type="string")  
     */  
    private $primeiroNome;  
    /**  
     * @Column(name="ultimo_nome", type="string")  
     */  
    private $ultimoNome;  
    /**  
     * @Column(name="data_atualizado", type="datetime")  
     */  
    private $ultimaAtualizacao;  
}
```

7) Execute os comandos `vendor/bin/doctrine orm:info` e `vendor/bin/doctrine orm:mapping:describe Ator` para garantir que todo o mapeamento ocorreu corretamente.