08

Consolidando o seu conhecimento

Chegou a hora de você seguir todos os passos realizados por mim durante esta aula:

- Dentro da pasta **scr/Entity**, crie a classe `Telefone`, com o `namespace Alura\Doctrine\Entity` e anotada com `@Entity`
  - Nesta classe, crie os atributos privados `id` (crie também seu *getter*), `numero` (crie também seu *getter* e *setter*) e `aluno` (crie também seu *getter* e *setter*)
  - Anote o atributo `id` com `@Id`, `@GeneratedValue` e `@Column`, dizendo que a coluna será do tipo `integer`
  - Anote o atributo `numero` com `@Column`, dizendo que a coluna será do tipo `string`
  - Anote o atributo `aluno` com `@ManyToOne`, dizendo que a entidade alvo será `Aluno`
- Na classe `Aluno`, adicione o atributo `telefones`. Anote-o com `@OneToMany`, configurando que a entidade alvo será `Telefone`, qual propriedade de `Telefone` se refere à classe e que as operações `remove` e `persist` aconteçam em cascata:

```
/**
* @OneToMany(targetEntity="Telefone", mappedBy="aluno", cascade={"remove", "persist"})
*/
private $telefones;
```

  - No construtor da classe, instancie o atributo `telefones` como um `ArrayCollection`:

```
public function __construct()
{
$this->telefones = new ArrayCollection();
}
```

  - Implemente os métodos `getTelefones`, que retorna a coleção de telefones, e `addTelefone`, que recebe um `Telefone` por parâmetro, o adiciona na coleção de telefones e atribui o seu `Aluno`:

```
public function addTelefone(Telefone $telefone)
{
    $this->telefones->add($telefone);
    $telefone->setAluno($this);

    return $this;
}

public function getTelefones(): Collection
{
    return $this->telefones;
}
```

- Para configurar as ***migrations***, na pasta do seu projeto, crie o arquivo **migrations.php**, com o seguinte conteúdo:

```
<?php
```

```
return [
    'name' => 'Fundamentos Doctrine',
    'migrations_namespace' => 'Alura\\Doctrine\\Migrations',
    'table_name' => 'doctrine_migration_versions',
    'column_name' => 'version',
    'column_length' => 14,
    'executed_at_column_name' => 'executed_at',
    'migrations_directory' => 'src/Migrations',
    'all_or_nothing' => true,
];
```

- Além disso, crie a pasta **Migrations**, dentro de **src**, e exclua a base de dados **banco.sqlite**, dentro de **var/data**
- Através da linha de comando, dentro da pasta do seu projeto, instale o pacote **doctrine/migrations**:

```
composer require doctrine/migrations
```

- Ainda na linha de comando, gere uma *migration*, comparando o seu banco de dados atual com as informações de mapeamento:

```
vendor\bin\doctrine-migrations migrations:diff
```

- Em seguida, execute as *migrations*:

```
vendor\bin\doctrine-migrations migrations:migrate
```

- No arquivo **criar-aluno.php**, antes do `persist`, faça um `for` para pegar o segundo argumento em diante da linha de comando, que serão o(s) telefone(s) do aluno. Com isso, instancie um `Telefone`, que terá esse número e o adicione na coleção de telefones do aluno:

```
for ($i = 2; $i < $argc; $i++) {
  $numeroTelefone = $argv[$i];
  $telefone = new Telefone();
  $telefone->setNumero($numeroTelefone);

  $aluno->addTelefone($telefone);
}
```

- No arquivo **buscar-alunos.php**, dentro do `foreach`, implemente o código para imprimir também os telefones do aluno:

```
foreach ($alunoList as $aluno) {
  $telefones = $aluno
      ->getTelefones()
      ->map(function (Telefone $telefone) {
          return $telefone->getNumero();
      })
      ->toArray();
  echo "ID: {$aluno->getId()}\nNome: {$aluno->getNome()}\n";
  echo "Telefones: " . implode(', ', $telefones);
```

```
        echo "\n\n";
    }
```

Caso já tenha feito, excelente. Se ainda não, é importante que você execute o que foi visto nos vídeos para poder continuar com a próxima aula.