

Consolidando o seu conhecimento

Chegou a hora de você seguir todos os passos realizados por mim durante esta aula. Caso já tenha feito, excelente. Se ainda não, siga os passos abaixo:

1) Na pasta `app/`, crie uma nova pasta `Services`, e dentro dela, crie a classe `CriadorDeSerie`, com o seguinte conteúdo (já refatorado):

```
<?php

namespace App\Services;

use App\Serie;
use Illuminate\Support\Facades\DB;

class CriadorDeSerie
{
    public function criarSerie(
        string $nomeSerie,
        int $qtdTemporadas,
        int $epPorTemporada
    ): Serie {
        DB::beginTransaction();
        $serie = Serie::create(['nome' => $nomeSerie]);
        $this->criaTemporadas($qtdTemporadas, $epPorTemporada, $serie);
        DB::commit();

        return $serie;
    }

    /**
     * @param int $qtdTemporadas
     * @param int $epPorTemporada
     * @param Serie $serie
     */
    private function criaTemporadas(int $qtdTemporadas, int $epPorTemporada, Serie $serie): void
    {
        for ($i = 1; $i <= $qtdTemporadas; $i++) {
            $temporada = $serie->temporadas()->create(['numero' => $i]);

            $this->criaEpisodios($epPorTemporada, $temporada);
        }
    }

    /**
     * @param int $epPorTemporada
     * @param \Illuminate\Database\Eloquent\Model $temporada
     */
    private function criaEpisodios(int $epPorTemporada, \Illuminate\Database\Eloquent\Model $temporada)
    {
        for ($j = 1; $j <= $epPorTemporada; $j++) {
            $temporada->episodios()->create(['numero' => $j]);
        }
    }
}
```

```

    }
}

```

2) Na mesma pasta `Services`, crie a classe `RemovedorDeSerie`, para controlar a remoção da série:

```

<?php
namespace App\Services;

use App\{Serie, Temporada, Episodio};
use Illuminate\Support\Facades\DB;

class RemovedorDeSerie
{
    public function removerSerie(int $serieId): string
    {
        $nomeSerie = '';
        DB::transaction(function () use ($serieId, &$nomeSerie) {
            $serie = Serie::find($serieId);
            $nomeSerie = $serie->nome;

            $this->removerTemporadas($serie);
            $serie->delete();
        });

        return $nomeSerie;
    }

    /**
     * @param $serie
     */
    private function removerTemporadas(Serie $serie): void
    {
        $serie->temporadas->each(function (Temporada $temporada) {
            $this->removerEpisodios($temporada);
            $temporada->delete();
        });
    }

    /**
     * @param Temporada $temporada
     */
    private function removerEpisodios(Temporada $temporada): void
    {
        $temporada->episodios->each(function (Episodio $episodio) {
            $episodio->delete();
        });
    }
}

```

3) Abra a classe `SeriesController` e use a classe `CriadorDeSerie` no método `store`:

```

public function store(
    SeriesFormRequest $request,
    CriadorDeSerie $criadorDeSerie
) {
    //
}

```

```
) {  
    $serie = $criadorDeSerie->criarSerie(  
        $request->nome,  
        $request->qtd_temporadas,  
        $request->ep_por_temporada  
    );  
  
    $request->session()  
        ->flash(  
            'mensagem',  
            "Série {$serie->id} e suas temporadas e episódios criados com sucesso {$serie->nome}"  
        );  
  
    return redirect()->route('listar_series');  
}
```

4) Na mesma classe, altere o método `destroy` para usar o `RemovedorDeSerie` :

```
public function destroy(Request $request, RemovedorDeSerie $removedorDeSerie)  
{  
    $nomeSerie = $removedorDeSerie->removerSerie($request->id);  
    $request->session()  
        ->flash(  
            'mensagem',  
            "Série $nomeSerie removida com sucesso"  
        );  
    return redirect()->route('listar_series');  
}
```