

Abstract geometric shapes in the top-left corner, including a light blue parallelogram, a green parallelogram, a cyan parallelogram, an orange parallelogram, and a magenta parallelogram.

LARAVEL

AULA 05

Abstract geometric shapes in the top-right corner, including a light blue parallelogram, a green parallelogram, a cyan parallelogram, a magenta parallelogram, a blue parallelogram, and an orange parallelogram.

Abstract geometric shapes in the top-left corner, including a light blue parallelogram, a green parallelogram, a brown parallelogram, and a large magenta parallelogram.

RELACIONAMENTOS

Gerenciamento de relacionamentos em Laravel

Abstract geometric shapes in the top-right corner, including a light blue parallelogram, a green parallelogram, a magenta parallelogram, and a red parallelogram.

RELACIONAMENTOS

DIRECIONALIDADE

A -- pertence--> B

Não é igual a...

B -- pertence--> A

CARDINALIDADE

A -- tem um --> B

Não é igual a...

A -- tem muitos --> B

ELOQUENT: RELACIONAMENTOS

N:1 - Belongs To

- ✗ Episode **belongsTo** Season
- ✗ Profile **belongsTo** User

1:0...1 - Has One

- ✗ User **hasOne** Profile
- ✗ Person **hasOne** DNI

1:0...N - Has Many

- ✗ Season **hasMany** Episode
- ✗ Genre **hasMany** Movie

N:M - Belongs to Many

- ✗ Actor **belongsToMany** Movie
- ✗ Client **belongsToMany** Product

```
class User extends Model
{
    // hasOne / belongsTo
    public function profile()
    {
        return $this->hasOne(Profile::class, 'foreign_key', 'other_key');
    }
}
```

```
class Profile extends Model
{
    public function user()
    {
        return $this->belongsTo(User::class, 'foreign_key', 'other_key');
    }
}
```

```
// A partir de um Objeto User
$user = User::find(1);
```

```
// Obtemos o objeto Profile relacionado
$profile = $user->profile;
```

```
// E vice-versa...
$profile = Profile::find(42);
$user = $profile->user;
```

```
class User extends Model
{
    // hasMany / belongsTo
    public function pets()
    {
        return $this->hasMany(Pet::class, 'foreign_key', 'local_key');
    }
}

class Pet extends Model
{
    public function user()
    {
        return $this->belongsTo(User::class, 'foreign_key', 'other_key');
    }
}

// A partir de um Objeto User, obtemos a coleção de objetos Pet relacionados
$user = User::find(1);
$pets = $user->pets();

// E vice-versa
$pet = Pet::find(9);
$user = $pet->user();

// Podemos fazer uma query para obter apenas alguns
$dogs = $user->pets()->where('type', 'dog')->get();
```

```
class Actor extends Model
{
    // belongsToMany
    public function movies()
    {
        return $this->belongsToMany(Movie::class, 'table', 'foreign_key', 'other_key');
    }
}

class Movie extends Model
{
    public function actors()
    {
        return $this->belongsToMany(Actor::class, 'table', 'foreign_key', 'other_key');
    }
}
```

// A partir de um Objeto Actor, obtemos a coleção de objetos Movies relacionados

```
$actor = Actor::find(1);
```

```
$movies = $actor->movies;
```

// E vice-versa

```
$movie = Movie::find(23);
```

```
$actors = $movie->actors;
```

// Também podemos fazer queries

```
$bestActors = $movie->actors()->where('rating', '>', 8)->orderBy('last_name')->get();
```

HORA DE PRATICAR!



Fazer o exercício 1

The background features abstract, overlapping geometric shapes in various colors including light blue, green, cyan, magenta, orange, and red. These shapes are arranged in a way that creates a sense of depth and movement, framing the central text.

**ATÉ A
PRÓXIMA!**