

PROIEKTU BAT SORTU

- AULA ADIBIDEA

[N-M erlazioa](#)[Eloquent erlazioak](#)[Seeders - Alumno_curso](#)

N-M erlazioa

Aurreko datu basearekin jarraituz, ikusi dezakegu ikasturte eta ikasleen arteko erlazioa **N-M erlazioa** da, hau da, ikasle baten hainbat kurso eman ditzake eta aldiz ikasturte batek hainbat ikasle izango ditu. Beraz, kasu honetan beste taula bat izango da beharrezkoa.

Horretarako migrazio berri bat egingo dugu, Laravelen konbentzioa jarraituz, erlaziozatzen diren bi taulen izenak izango ditu, singularrean eta alfabetikoki ordenaduta, kasu honetan alumnos eta cursos erlazioatuko ditugunez, taula alumno_curso izango da.

Migrazioa sortzeko:

```
php artisan make:migration create_alumno_curso_table
```

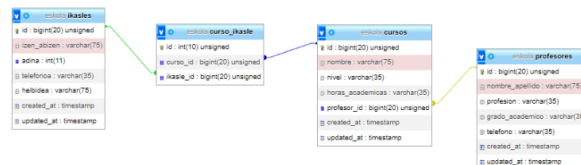
Migrazioan idatziko duguna, alde batetik id biak sortu eta bestetik klabe foraneak direla esango dugu:

```
Schema::create('alumno_curso', function (Blueprint $table) {
    $table->id();
    $table->unsignedBigInteger('curso_id');
    $table->foreign('curso_id')->references('id')->on('cursos')->onDelete('cascade');
    $table->unsignedBigInteger('alumno_id');
    $table->foreign('alumno_id')->references('id')->on('alumnos')->onDelete('cascade');
    $table->timestamps();
});
```

Ohartu zelan kasu honetan onDelete() egitureakoan "cascade" jarri dugula, kasu honetan, bai curso bat, bai alumno bat ezabatzerakoan gure dugulako taula honetatik ere ezabatu daitezten.

Behin eginda, gogoratu exekutatzeko.

```
php artisan migrate
```



Eloquent erlazioak

Erlazioak modeloei adierazi behar dira ere. Horretarako hainbat gauza izan behar ditugu kontutan

- Ikasle batek hainbat kurtsoan egon daiteke eta kurtso bakoitzean hainbat ikasle daude, beraz badugu N-M erlazioa bat.



Hori definitzeko, Alumno modeloean, **belongsToMany** metodoa erabiliko dugu, metodo honek definitzen bait du N-M erlazioak.

```
public function cursos(){
    return $this->belongsToMany(Curso::class);
}
```

Eta modu berean ere, Curso modeloean ere ikasleekiko erlazioa definituko dugu:

```
public function alumnos(){
    return $this->belongsToMany(Alumno::class);
}
```

Seeders - Alumno_curso

"Cursos" Seeder

```
php artisan make:seeder AlumnoCursoSeeder
```

Gogoratu modelo biak gehitu behar ditugula.

```
use App\Models\Curso;  
use App\Models\Alumno;
```

Pivot taulan datuak sartzeko, gure kasuan `alumno_curso`, Eloquent baditu bere funtzioak, eta euren artean **attach()** dugu.

```
$curso->alumnos()->attach(1);  
$curso->alumnos()->attach(2);  
$curso1->alumnos()->attach(1);  
$curso1->alumnos()->attach(3);  
$curso2->alumnos()->attach(1);  
$curso2->alumnos()->attach(2);
```

DataBaseSeeder-a exekutatu dugu, horretarako beharrezkoa den deia jarriko dugu:

```
$this->call(CursoSeeder::class);
```

Eta orain bai exekutatu dezakegula gure Seederra:

```
php artisan db:seed
```

```
$curso = new Curso();  
$curso->nombre = 'Laravel 9';  
$curso->nivel = 'Básico';  
$curso->horas_academicas = '60 horas';  
$curso->profesor_id = 1;  
$curso->save();  
  
$curso1 = new Curso();  
$curso1->nombre = 'Contabilidad Básica';  
$curso1->nivel = 'Básico';  
$curso1->horas_academicas = '40 horas';  
$curso1->profesor_id = 2;  
$curso1->save();  
  
$curso2 = new Curso();  
$curso2->nombre = 'Desarrollo PHP';  
$curso2->nivel = 'Intermedio';  
$curso2->horas_academicas = '80 horas';  
$curso2->profesor_id = 3;  
$curso2->save();  
  
// Pivot taulan datuak sartzeko  
  
$curso->alumnos()->attach(1);  
$curso->alumnos()->attach(2);  
$curso1->alumnos()->attach(1);  
$curso1->alumnos()->attach(3);  
$curso2->alumnos()->attach(1);  
$curso2->alumnos()->attach(2);
```