



**ESCUELA POLITÉCNICA NACIONAL**  
ESCUELA DE FORMACIÓN DE TECNÓLOGOS



**DESARROLLO DE APLICACIONES WEB**  
**(TDSD414)**

ASIGNATURA:

Desarrollo de Aplicaciones Web

PROFESOR:

Ing. Ivonne Maldonado

PERÍODO ACADÉMICO:

2022-B

**LABORATORIO - 8**

**TÍTULO:**

**RELACIONES**



## PROPÓSITO DE LA PRÁCTICA

Familiarizar al estudiante con el manejo relaciones en Laravel.

## OBJETIVO GENERAL

Determinar el manejo de relaciones.

## OBJETIVOS ESPECÍFICOS

- Configurar las herramientas para el laboratorio.
- Crear un proyecto en Laravel.
- Comprender los archivos y directorios más importantes en Laravel.
- Comprender como funciona las relaciones.
- Comprender como funciona las relaciones polimórficas.
- Comprender como visualizar los resultados de las relaciones en un proyecto.
- Visualizar los resultados obtenidos.

## LINK DE GITHUB

<https://github.com/IvonneSDQ/Relaciones>

## INSTRUCCIONES

1. Crear un nuevo proyecto y verificar la ejecución del mismo.

```
PS C:\Users\IvonneSDQ05\Laboratorios Web 2022B> laravel new Relaciones

  LARAVEL

Creating a "laravel/laravel" project at "./Relaciones"
Info from https://repo.packagist.org: #StandWithUkraine
Installing laravel/laravel (v9.5.0)
- Downloading laravel/laravel (v9.5.0)
  - Installing laravel/laravel (v9.5.0): Extracting archive
Created project in C:\Users\IvonneSDQ05\Laboratorios Web 2022B\Relaciones
> @php -r "file_exists('.env') || copy('.env.example', '.env');"

80 packages you are using are looking for funding.
Use the `composer fund` command to find out more!
> @php artisan vendor:publish --tag=laravel-assets --ansi --force

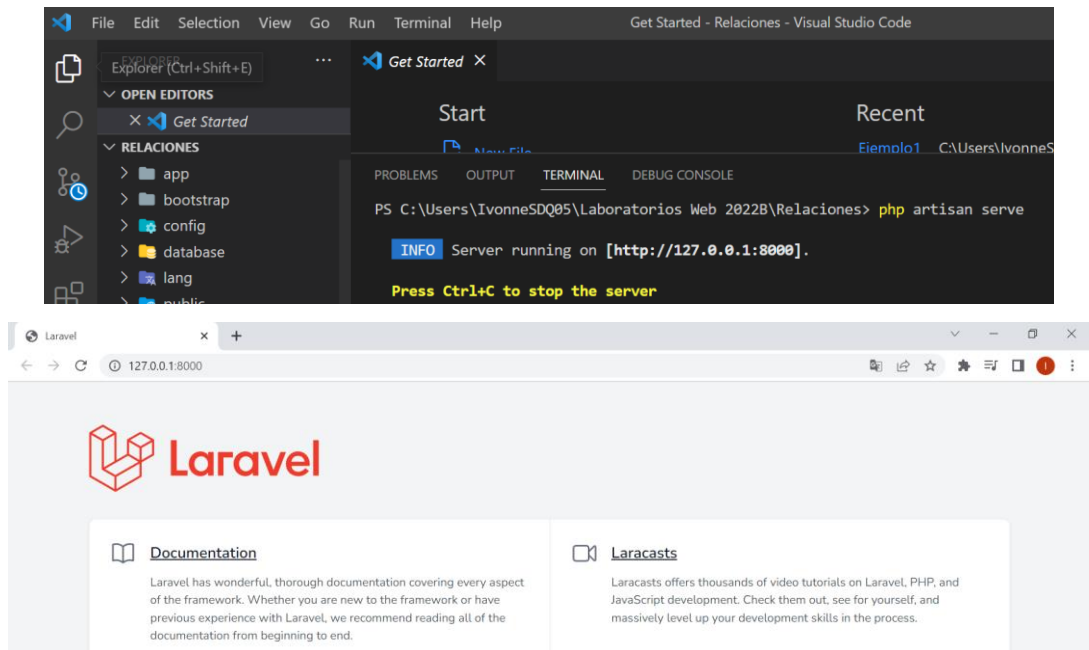
INFO No publishable resources for tag [laravel-assets].

No security vulnerability advisories found
> @php artisan key:generate --ansi

INFO Application key set successfully.

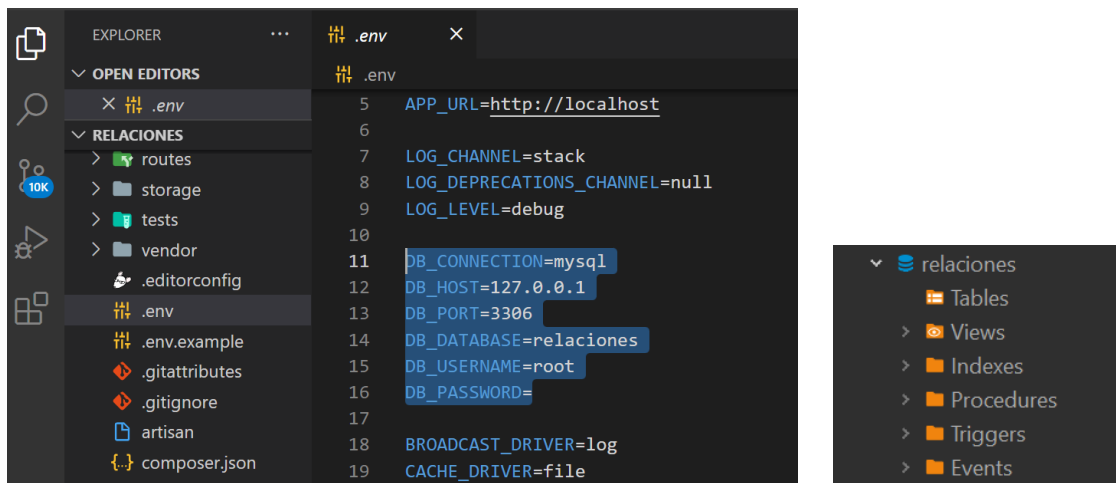
INFO Application ready! Build something amazing.

PS C:\Users\IvonneSDQ05\Laboratorios Web 2022B> |
```



## 2. Realizar la conexión con la base de datos

<https://laravel.com/docs/9.x/database#introduction>



## 3. Verificar que la conexión esta configurada de manera correcta, ejecutar las migraciones con el siguiente comando: **php artisan migrate**

<https://laravel.com/docs/9.x/migrations#running-migrations>

```

PS C:\Users\IvonneSDQ05\Laboratorios Web 2022B\Relaciones> php artisan migrate

INFO: Preparing database.

Creating migration table ..... 56ms DONE

INFO: Running migrations.

2014_10_12_000000_create_users_table ..... 47ms DONE
2014_10_12_100000_create_password_resets_table ..... 98ms DONE
2019_08_19_000000_create_failed_jobs_table ..... 40ms DONE
2019_12_14_000001_create_personal_access_tokens_table ..... 54ms DONE

PS C:\Users\IvonneSDQ05\Laboratorios Web 2022B\Relaciones>

```

relaciones	
Tables	
failed_jobs	32K
migrations	16K
password_resets	16K
personal_access_tokens	48K
users	32K

#### 4. Trabajar con las relaciones

<https://laravel.com/docs/9.x/eloquent#introduction>

<https://laravel.com/docs/9.x/eloquent-relationships#introduction>

<https://laravel.com/docs/9.x/eloquent-relationships#one-to-one>

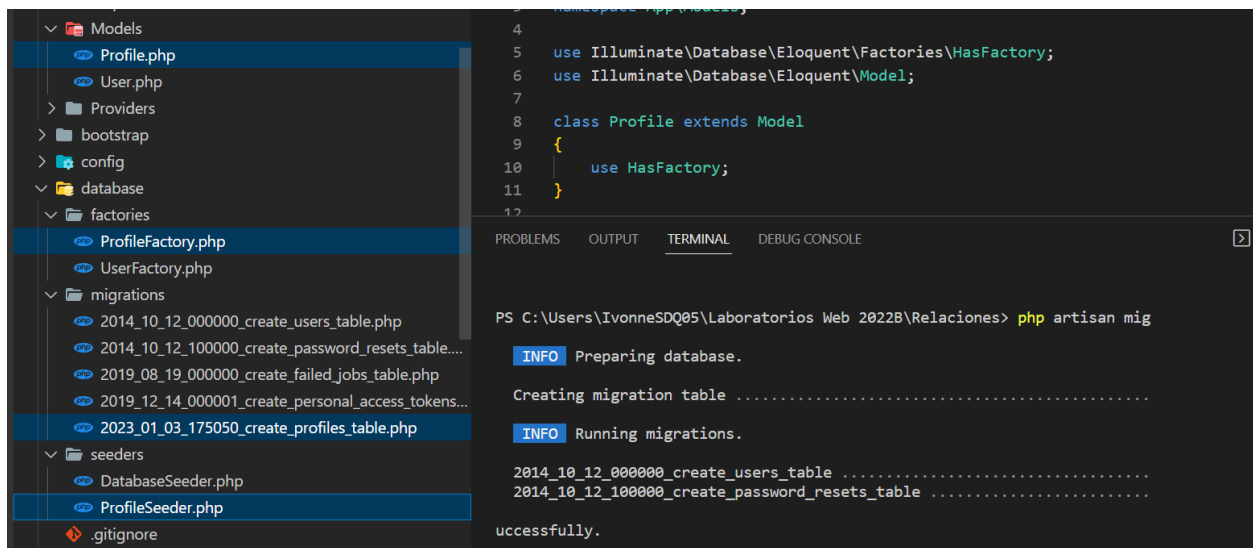
### RELACIÓN DE UNO A UNO

usuarios			perfil					
id	nombre	email	id	nombre	bibliografia	github	website	user_id
1	Ivonne	<a href="mailto:ivonne@gmail.com">ivonne@gmail.com</a>	1	backend	soy un backend	<a href="https://github.com/">https://github.com/</a>	<a href="https://esfot.epn.edu.ec">esfot.epn.edu.ec</a>	1
2	Edison	<a href="mailto:edison@gmail.com">edison@gmail.com</a>	2	frontend	soy un frontend	<a href="https://github.com/">https://github.com/</a>	<a href="https://esfot.epn.edu.ec">esfot.epn.edu.ec</a>	2
			3	fullstack	soy un fullstack	<a href="https://github.com/">https://github.com/</a>	<a href="https://esfot.epn.edu.ec">esfot.epn.edu.ec</a>	

<

5. Crear un nuevo modelo llamado Profile, con sus respectiva migración, factory y sedder utilizando el comando: **php artisan make:model Profile -mfs**

<https://laravel.com/docs/9.x/eloquent#generating-model-classes>

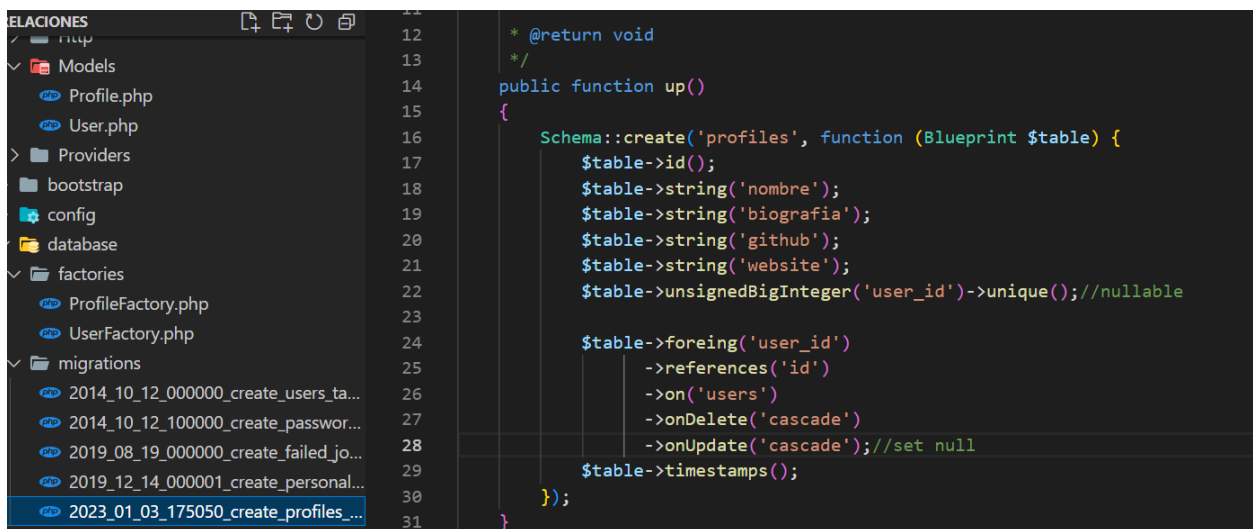


- Definir los campos (nombre de las columnas) para la tabla de la base de datos, esto se lo realiza en la migración del modelo respectivo.

**NOTA:** Recordar que un usuario tiene un perfil y que el perfil le pertenece a un único usuario

Este procedimiento realiza la relación a nivel de tablas de la **Base de Datos** únicamente.

<https://laravel.com/docs/9.x/migrations#tables>



## Código

```

$table->id();
$table->string('nombre');
$table->string('biografia');
$table->string('github');
$table->string('website');
$table->unsignedBigInteger('user_id')->unique(); //->nullable()

$table->foreign('user_id')
    ->references('id')
    ->on('users')
    ->onDelete('cascade')
    ->onUpdate('cascade'); // set null

$table->timestamps();

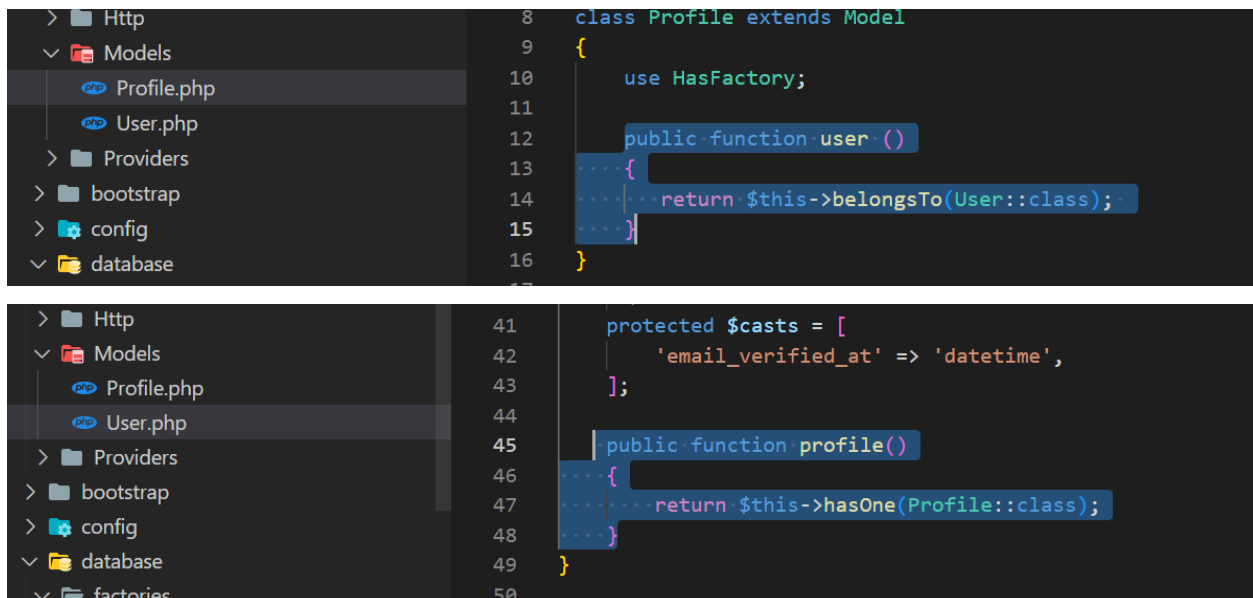
```

7. Este mismo procedimiento debe hacerse a nivel de **Eloquent de Laravel**.

**NOTA: Recordar que un usuario tiene un perfil y que el perfil le pertenece a un único usuario**

<https://laravel.com/docs/9.x/eloquent-relationships#one-to-one>

<https://laravel.com/docs/9.x/eloquent-relationships#one-to-one-defining-the-inverse-of-the-relationship>



## Código

```

// RELACIÓN DE UNO A UNO
public function user()
{
    return $this->belongsTo(User::class);
}

```

```
// RELACIÓN DE UNO A UNO
public function profile()
{
    return $this->hasOne(Profile::class);
}
```

8. Ejecutar la migración y verificar si realmente esta correcta la implementación de la relación uno a uno, con el siguiente comando: **php artisan migrate**

```
PS C:\Users\IvonneSDQ05\Laboratorios Web 2022B\Relaciones> php artisan migrate

INFO Running migrations.

2023_01_03_175050_create_profiles_table ..... 166ms DONE

PS C:\Users\IvonneSDQ05\Laboratorios Web 2022B\Relaciones>
```

9. Insertar datos en la base de datos, por esta ocasión hagalo de forma manual. Recuerde que podría hacerlo con datos fake.

### Código

```
INSERT INTO `relaciones`.`users` (`id`, `name`, `email`, `email_verified_at`,
`password`, `remember_token`) VALUES ('1', 'Ivonne', 'ivonne@hotmail.com',
'2023-01-01 13:19:25', 'sistemas', 'sistemas');
INSERT INTO `relaciones`.`users` (`id`, `name`, `email`, `email_verified_at`,
`password`, `remember_token`) VALUES ('2', 'Edison', 'edison@hotmail.com',
'2023-01-01 13:19:25', 'sistemas', 'sistemas');
INSERT INTO `relaciones`.`users` (`id`, `name`, `email`, `email_verified_at`,
`password`, `remember_token`) VALUES ('3', 'Gaby', 'gaby@hotmail.com',
'2023-01-01 13:19:25', 'sistemas', 'sistemas');
```

users 1 x

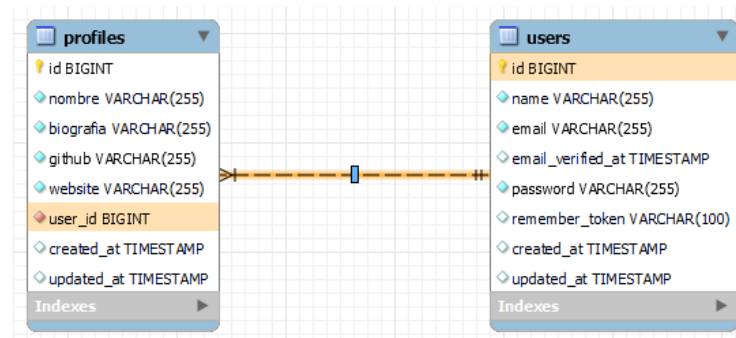
select \* from relaciones.users

	id	name	email	email_verified_at	password	remember_token	created_at	updated_at
1	1	Ivonne	ivonne@hotmail.com	2023-01-01 13:19:25	sistemas	sistemas	[NULL]	[NULL]
2	2	Edison	edison@hotmail.com	2023-01-01 13:19:25	sistemas	sistemas	[NULL]	[NULL]
3	3	Gaby	gaby@hotmail.com	2023-01-01 13:19:25	sistemas	sistemas	[NULL]	[NULL]

## Código

```
INSERT INTO `relaciones`.`profiles` (`id`, `nombre`, `biografia`, `github`,
`website`, `user_id`) VALUES ('1', 'backend', 'soy un backend',
'https://github.com/', 'https://esfot.epn.edu.ec/', '1');
INSERT INTO `relaciones`.`profiles` (`id`, `nombre`, `biografia`, `github`,
`website`, `user_id`) VALUES ('2', 'frontend', 'soy un frontend',
'https://github.com/', 'https://esfot.epn.edu.ec/', '2');
INSERT INTO `relaciones`.`profiles` (`id`, `nombre`, `biografia`, `github`,
`website`, `user_id`) VALUES ('3', 'fullstack', 'soy un fullstack',
'https://github.com/', 'https://esfot.epn.edu.ec/', '3');
```

	id	nombre	biografia	github	website	user_id	created_at	updated_at
1	1	backend	soy un backend	https://github.com/	https://esfot.epn.edu.ec/	1	[NULL]	[NULL]
2	2	frontend	soy un frontend	https://github.com/	https://esfot.epn.edu.ec/	2	[NULL]	[NULL]
3	3	fullstack	soy un fullstack	https://github.com/	https://esfot.epn.edu.ec/	3	[NULL]	[NULL]



### 10. Verificar ejecutando el comando: **php artisan tinker**

```
> use App\Models\User
> $user = User::find(1)
= App\Models\User {#4017
  id: 1,
  level_id: null,
  name: "Ivonne",
  email: "ivonne@hotmail.com",
  email_verified_at: "2023-01-01 13:19:25",
  #password: "sistemas",
  #remember_token: "sistemas",
  created_at: null,
  updated_at: null,
}
```



```

> use App\Models\Profile
> $profile = Profile::find(1)
= App\Models\Profile {#4636
  id: 1,
  nombre: "backend",
  biografia: "soy un backend",
  github: "https://github.com/",
  website: "https://esfot.epn.edu.ec/",
  user_id: 1,
  created_at: null,
  updated_at: null,
}

> $profile -> user
= App\Models\User {#3681
  id: 1,
  level_id: null,
  name: "Ivonne",
  email: "ivonne@hotmail.com",
  email_verified_at: "2023-01-01 13:19:25", #password: "sistemas",
  #remember_token: "sistemas",
  created_at: null,
  updated_at: null,
}
>

```

## RELACIÓN DE UNO A MUCHOS

<https://laravel.com/docs/9.x/eloquent-relationships#one-to-many>

usuarios				niveles	
id	nombre	email	nivel_id	id	nombre
1	Ivonne	<a href="mailto:ivonne@gmail.com">ivonne@gmail.com</a>	1	1	oro
2	Edison	<a href="mailto:edison@gmail.com">edison@gmail.com</a>	2	2	plata
3	Gaby	<a href="mailto:gaby@hotmail.com">gaby@hotmail.com</a>	1	3	bronce

nivel\_id

Entidad fuerte onDelete

Entidad débil onUpdate

11. Crear un nuevo modelo llamado **Level** con su respectiva migración, factory y sedder con el siguiente comando: **php artisan make:model Level -mfs**

The screenshot shows an IDE with a file explorer on the left and a code editor on the right. The file explorer shows the project structure with folders like 'Models', 'Factories', 'Migrations', and 'Seeders'. The code editor shows the following PHP code for the 'Level' model:

```

1 <?php
2 namespace App\Models;
3
4 use Illuminate\Database\Eloquent\Factories\HasFactory;
5 use Illuminate\Database\Eloquent\Model;
6
7 class Level extends Model
8 {
9     use HasFactory;
10 }
11
12

```

The terminal at the bottom shows the output of the command `php artisan make:model Level -mfs`:

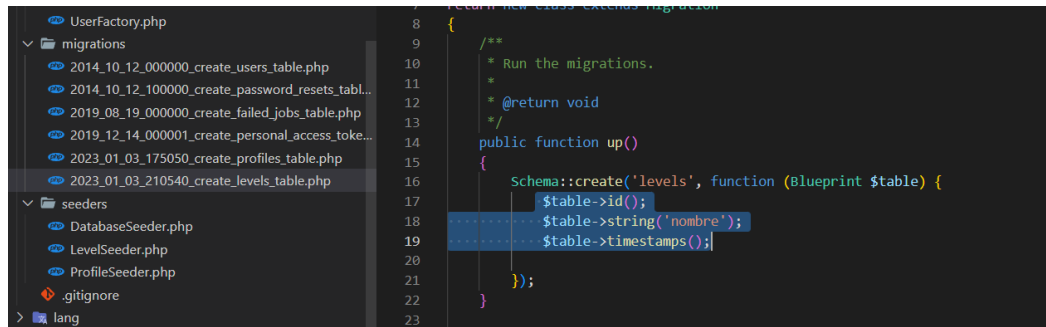
```

PS C:\Users\IvonneSQ05\Laboratorios Web 2022B\Relaciones> php artisan make:model Level -mfs
INFO Model [C:\Users\IvonneSQ05\Laboratorios Web 2022B\Relaciones\app\Models\Level.php] created successfully.
INFO Factory [C:\Users\IvonneSQ05\Laboratorios Web 2022B\Relaciones\database\factories\LevelFactory.php] created successfully.
INFO Migration [C:\Users\IvonneSQ05\Laboratorios Web 2022B\Relaciones\database\migrations\2023_01_03_210540_create_levels_table.php] created successfully.
INFO Seeder [C:\Users\IvonneSQ05\Laboratorios Web 2022B\Relaciones\database\seeders\LevelSeeder.php] created successfully.
PS C:\Users\IvonneSQ05\Laboratorios Web 2022B\Relaciones>
PS C:\Users\IvonneSQ05\Laboratorios Web 2022B\Relaciones>
PS C:\Users\IvonneSQ05\Laboratorios Web 2022B\Relaciones>
PS C:\Users\IvonneSQ05\Laboratorios Web 2022B\Relaciones>

```

12. Definir los campos (nombre de las columnas) para la tabla de la base de datos, esto se lo realiza en la migración del modelo respectivo.

**NOTA:** Recordar que un nivel tiene varios usuarios y que el usuario pertenece a un nivel



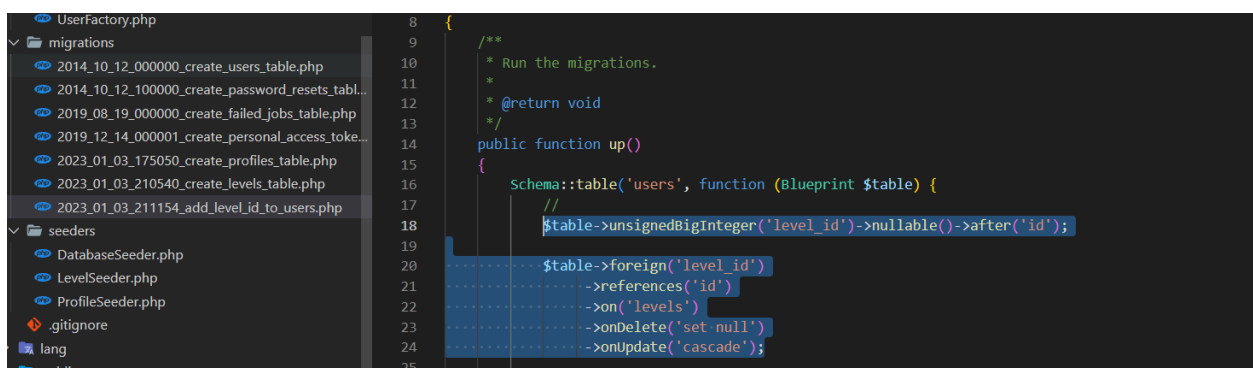
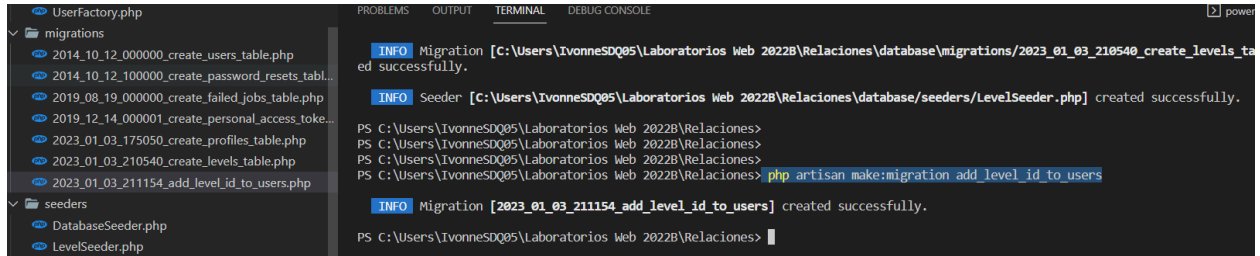
### Código

```
$table->id();
$table->string('nombre');
$table->timestamps();
```

13. Con este procedimiento se realizó la relación a nivel de tablas de la **Base de Datos** únicamente, pero hay que tener en cuenta sobre las relaciones fuertes y débiles. Para ello se agrega una migración personalizada, ejecutando el siguiente comando: **php artisan make:migration add\_level\_id\_to\_users**

<https://laravel.com/docs/7.x/migrations#column-modifiers>

<https://laravel.com/docs/9.x/migrations#column-modifiers>

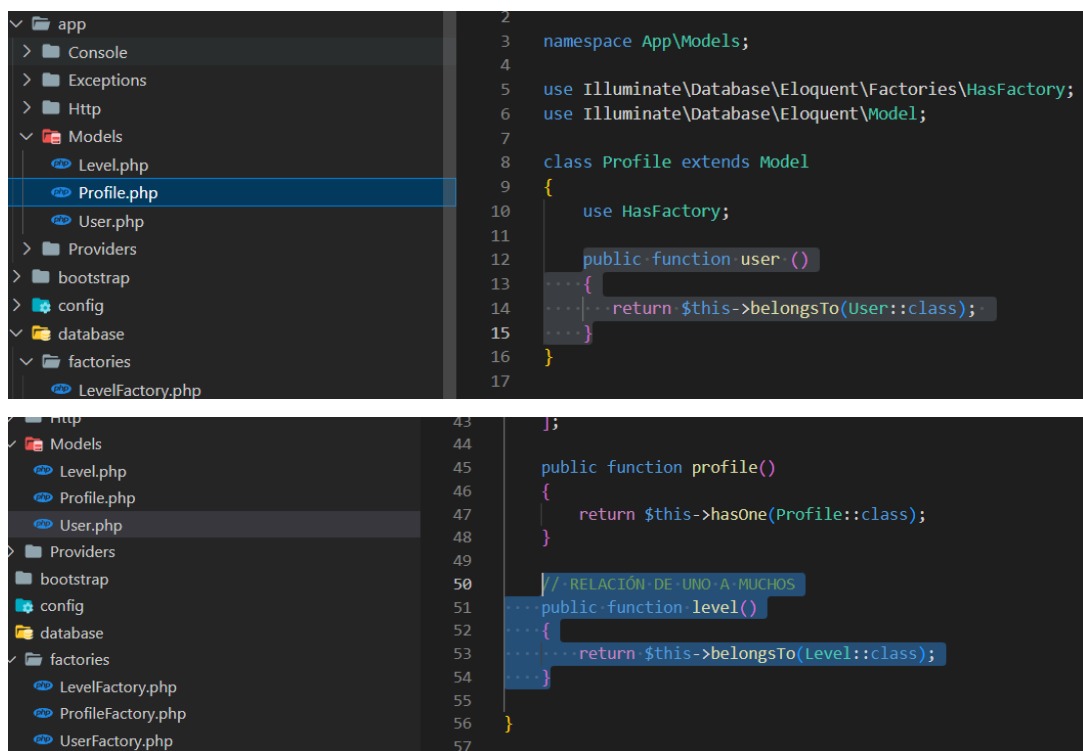


## Código

```
$table->unsignedBigInteger('level_id')->nullable()->after('id');

$table->foreign('level_id')
    ->references('id')
    ->on('levels')
    ->onDelete('set null')
    ->onUpdate('cascade');
```

<https://laravel.com/docs/9.x/eloquent-relationships#one-to-many>



## Código

```
// RELACIÓN DE UNO A MUCHOS
public function users()
{
    return $this->hasMany('App\Models\User');
}
```

```
// RELACIÓN DE UNO A MUCHOS
public function level()
{
    return $this->belongsToMany(Level::class);
}
```

15. Ejecutar la migración y verificar si realmente esta correcta la implementación de la relación uno a uno, ejecutando el comando: **php artisan migrate**

```

PS C:\Users\IvonneSDQ05\Laboratorios Web 2022B\Relaciones> php artisan migrate

INFO: Running migrations.

2023_01_03_210540_create_levels_table ..... 51ms DONE
2023_01_03_211154_add_level_id_to_users ..... 88ms DONE

PS C:\Users\IvonneSDQ05\Laboratorios Web 2022B\Relaciones>

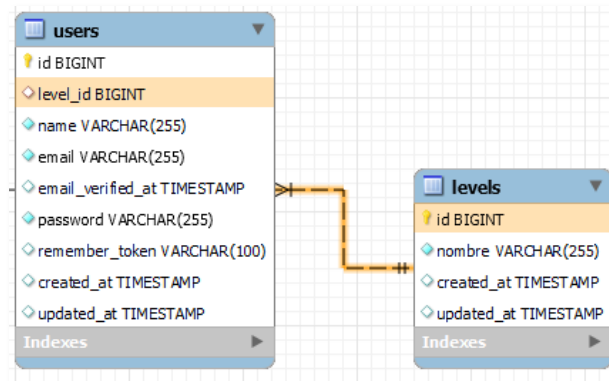
```

## Código

```

INSERT INTO `relaciones`.`levels` (`id`, `nombre`, `created_at`, `updated_at`) VALUES
('1', 'oro', '2023-01-01 13:19:25', '2021-11-20 13:19:25');
INSERT INTO `relaciones`.`levels` (`id`, `nombre`, `created_at`, `updated_at`) VALUES
('2', 'plata', '2023-01-01 13:19:25', '2021-11-20 13:19:25');
INSERT INTO `relaciones`.`levels` (`id`, `nombre`, `created_at`, `updated_at`) VALUES
('3', 'bronce', '2023-01-01 13:19:25', '2021-11-20 13:19:25');

```



16. Para probar se puede verificar ejecutando: **php artisan tinker**

```

> use App\Models\User
> $user = User::find(1)
= App\Models\User {#4379
  id: 1,
  level_id: null,
  name: "Ivonne",
  email: "ivonne@hotmail.com",
  email_verified_at: "2023-01-01 13:19:25",
  #password: "sistemas",
  #remember_token: "sistemas",
  created_at: null,
  updated_at: null,
}

```

```

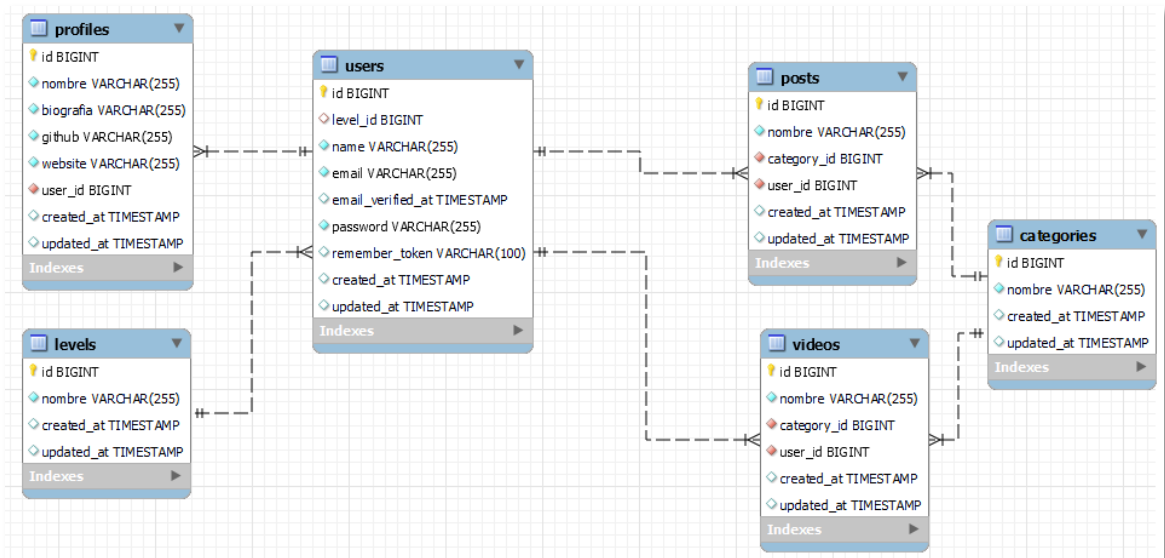
> use App\Models\Level
> $level = Level::find(1)
= App\Models\Level {#4017
  id: 1,
  nombre: "oro",
  created_at: "2023-01-01 13:19:25",
  updated_at: "2021-11-20 13:19:25",
}

> $level -> user
= null

>

```

17. Realizar las otras relaciones restantes



- **Un usuario puede hacer varios posts y que un post le pertenece a un usuario**

## MIGRACIÓN – POST

```
$table->id();
      $table->string('nombre');
      $table->timestamps();
```

## MIGRACIÓN - add\_post\_id\_to\_users

```
$table->unsignedBigInteger('user_id');

        $table->foreign('user_id')
            ->references('id')
            ->on('users')
            ->onDelete('cascade')
            ->onUpdate('cascade');
```

## MODELO - USER

```
// RELACIÓN DE UNO A MUCHOS
public function posts()
{
    return $this->hasMany(Post::class);
}
```

## MODELO – POST

```
// RELACIÓN DE UNO A MUCHOS
public function user()
{
    return $this->belongsTo(User::class);
}
```

- **Un usuario puede tener varios videos y que un video le pertenece a un usuario**

## MIGRACIÓN – VIDEOS

```
$table->id();  
$table->string('nombre');
```

```
$table->timestamps();
```

### MIGRACIÓN - add\_video\_id\_to\_users

```
$table->unsignedBigInteger('user_id');

$table->foreign('user_id')
    ->references('id')
    ->on('users')
    ->onDelete('cascade')
    ->onUpdate('cascade');
```

### MODELO – USER

```
// RELACIÓN DE UNO A MUCHOS
public function videos()
{
    return $this->hasMany(Video::class);
}
```

### MODELO – VIDEO

```
// RELACIÓN DE UNO A MUCHOS
public function user()
{
    return $this->belongsTo(User::class);
}
```

- Una categoría puede tener varios videos y que un video le pertenece a una categoría

### MIGRACIÓN – CATEGORÍAS

```
$table->id();
$table->string('nombre');
$table->timestamps();
```

### MIGRACIÓN - add\_video\_id\_to\_categories

```
$table->unsignedBigInteger('category_id');

$table->foreign('category_id')
    ->references('id')
    ->on('categories')
    ->onDelete('cascade')
    ->onUpdate('cascade');
```

### MODELO – CATEGORIA

```
// RELACIÓN DE UNO A MUCHOS
public function videos()
{
    return $this->hasMany(Video::class);
}
```

## MODELO – VIDEO

```
// RELACIÓN DE UNO A MUCHOS
public function category()
{
    return $this->belongsTo(Category::class);
}
```

- Una categoría puede tener varios posts y que un post le pertenece a una categoría

## MIGRACIÓN - add\_post\_id\_to\_categories

```
$table->unsignedBigInteger('category_id');

$table->foreign('category_id')
->references('id')
->on('categories')
->onDelete('cascade')
->onUpdate('cascade');
```

## MODELO – CATEGORIA

```
// RELACIÓN DE UNO A MUCHOS
public function posts()
{
    return $this->hasMany(Post::class);
}
```

## MODELO - POST

```
// RELACIÓN DE UNO A MUCHOS
public function category()
{
    return $this->belongsTo(Category::class);
}
```

```
INSERT INTO `relaciones`.`categories` (`id`, `nombre`, `created_at`, `updated_at`)
VALUES ('1', 'backend', '2021-10-11 00:00:00', '2021-10-11 00:00:00');
INSERT INTO `relaciones`.`categories` (`id`, `nombre`, `created_at`, `updated_at`)
VALUES ('2', 'frontend', '2021-10-11 00:00:00', '2021-10-11 00:00:00');
INSERT INTO `relaciones`.`categories` (`id`, `nombre`, `created_at`, `updated_at`)
VALUES ('3', 'redes', '2021-10-11 00:00:00', '2021-10-11 00:00:00');
INSERT INTO `relaciones`.`categories` (`id`, `nombre`, `created_at`, `updated_at`)
VALUES ('4', 'software', '2021-10-11 00:00:00', '2021-10-11 00:00:00');
INSERT INTO `relaciones`.`categories` (`id`, `nombre`, `created_at`, `updated_at`)
VALUES ('5', 'hardware', '2021-10-11 00:00:00', '2021-10-11 00:00:00');
```

```
INSERT INTO `relaciones`.`posts` (`id`, `nombre`, `category_id`, `user_id`, `created_at`,  
`updated_at`) VALUES ('1', 'el desarrollo de laravel', '1', '1', '2021-10-11 00:00:00', '2021-  
10-11 00:00:00');  
INSERT INTO `relaciones`.`posts` (`id`, `nombre`, `category_id`, `user_id`, `created_at`,  
`updated_at`) VALUES ('2', 'el desarrollo de nodejs', '1', '1', '2021-10-11 00:00:00', '2021-  
10-11 00:00:00');  
INSERT INTO `relaciones`.`posts` (`id`, `nombre`, `category_id`, `user_id`, `created_at`,  
`updated_at`) VALUES ('3', 'el desarrollo de React', '2', '2', '2021-10-11 00:00:00', '2021-10-  
11 00:00:00');  
INSERT INTO `relaciones`.`posts` (`id`, `nombre`, `category_id`, `user_id`, `created_at`,  
`updated_at`) VALUES ('4', 'el desarrollo de Angular', '2', '2', '2021-10-11 00:00:00', '2021-  
10-11 00:00:00');  
INSERT INTO `relaciones`.`posts` (`id`, `nombre`, `category_id`, `user_id`, `created_at`,  
`updated_at`) VALUES ('5', 'La arquitectura de una PC', '5', '3', '2021-10-11 00:00:00',  
'2021-10-11 00:00:00');  
INSERT INTO `relaciones`.`posts` (`id`, `nombre`, `category_id`, `user_id`, `created_at`,  
`updated_at`) VALUES ('6', 'La arquitectura de una Laptop', '5', '3', '2021-10-11 00:00:00',  
'2021-10-11 00:00:00');  
  
INSERT INTO `relaciones`.`videos` (`id`, `nombre`, `category_id`, `user_id`,  
`created_at`, `updated_at`) VALUES ('1', 'el desarrollo de laravel - video', '1', '2',  
'2021-10-11 00:00:00', '2021-10-11 00:00:00');  
INSERT INTO `relaciones`.`videos` (`id`, `nombre`, `category_id`, `user_id`,  
`created_at`, `updated_at`) VALUES ('2', 'el desarrollo de nodejs - video', '1', '2',  
'2021-10-11 00:00:00', '2021-10-11 00:00:00');  
INSERT INTO `relaciones`.`videos` (`id`, `nombre`, `category_id`, `user_id`,  
`created_at`, `updated_at`) VALUES ('3', 'el desarrollo de React - video', '2', '3',  
'2021-10-11 00:00:00', '2021-10-11 00:00:00');  
INSERT INTO `relaciones`.`videos` (`id`, `nombre`, `category_id`, `user_id`,  
`created_at`, `updated_at`) VALUES ('4', 'el desarrollo de Angular - video', '2', '3',  
'2021-10-11 00:00:00', '2021-10-11 00:00:00');  
INSERT INTO `relaciones`.`videos` (`id`, `nombre`, `category_id`, `user_id`,  
`created_at`, `updated_at`) VALUES ('5', 'La arquitectura de una PC - video', '5',  
'3', '2021-10-11 00:00:00', '2021-10-11 00:00:00');  
INSERT INTO `relaciones`.`videos` (`id`, `nombre`, `category_id`, `user_id`,  
`created_at`, `updated_at`) VALUES ('6', 'La arquitectura de una Laptop - video',  
'5', '3', '2021-10-11 00:00:00', '2021-10-11 00:00:00');
```

Verificar ejecutando: **php artisan tinker**



```

> use App\Models\User
> $user = User::find(1)
    name: "Ivonne",
    email: "ivonne@hotmail.co
m",
    email_verified_at: "2023-
01-01 13:19:25",
    #password: "sistemas",
    #remember_token: "sistema
s",
    created_at: null,
    updated_at: null,
  }

>

> $user -> post
= null

> $user -> posts
= Illuminate\Database\Eloquent
Collection {#4633
  all: [
    App\Models\Post {#4634
      id: 1,
      nombre: "el desarroll
o de laravel",
      category_id: 1,
      user_id: 1,
      created_at: "2021-10-
11 00:00:00",
      updated_at: "2021-10-
11 00:00:00",
    },
    App\Models\Post {#3686
      id: 2,
      nombre: "el desarrollo de nodejs",
      category_id: 1,
      user_id: 1,
      created_at: "2021-10-11 00:00:00",
      updated_at: "2021-10-11 00:00:00",
    },
  ],
}
>

```

```

> $user = App\Models\User::with('posts')->find(1)
= App\Models\User {#4640
  id: 1,
  level_id: 1,
  name: "Ivonne",
  email: "ivonne@hotmail.com",
  email_verified_at: "2023-01-01 13:19:25",
  #password: "sistemas",
  #remember_token: "sistemas",
  created_at: null,
  updated_at: null,
  posts: Illuminate\Database\Eloquent\Collection {#4644
    all: [
      App\Models\Post {#4651
        id: 1,
        nombre: "el desarrollo de laravel",
        category_id: 1,
        user_id: 1,
        created_at: "2021-10-11 00:00:00",
        updated_at: "2021-10-11 00:00:00",
      },
      App\Models\Post {#4649
        id: 2,
        nombre: "el desarrollo de nodejs",
        category_id: 1,
        user_id: 1,
        created_at: "2021-10-11 00:00:00",
        updated_at: "2021-10-11 00:00:00",
      },
    ],
  },
}
>

```

```

> $category = App\Models\Category::with('videos')->find(5)
= App\Models\Category {#4419
  id: 5,
  nombre: "hardware",
  created_at: "2021-10-11 00:00:00",
  updated_at: "2021-10-11 00:00:00",
  videos: Illuminate\Database\Eloquent\Collection {#4652
    all: [
      App\Models\Video {#4659
        id: 5,
        nombre: "La arquitectura de una PC - video",
        category_id: 5,
        user_id: 3,
        created_at: "2021-10-11 00:00:00",
        updated_at: "2021-10-11 00:00:00",
      },
      App\Models\Video {#4657
        id: 6,
        nombre: "La arquitectura de una Laptop - video",
        category_id: 5,
        user_id: 3,
        created_at: "2021-10-11 00:00:00",
        updated_at: "2021-10-11 00:00:00",
      },
    ],
  },
}
>

```

```

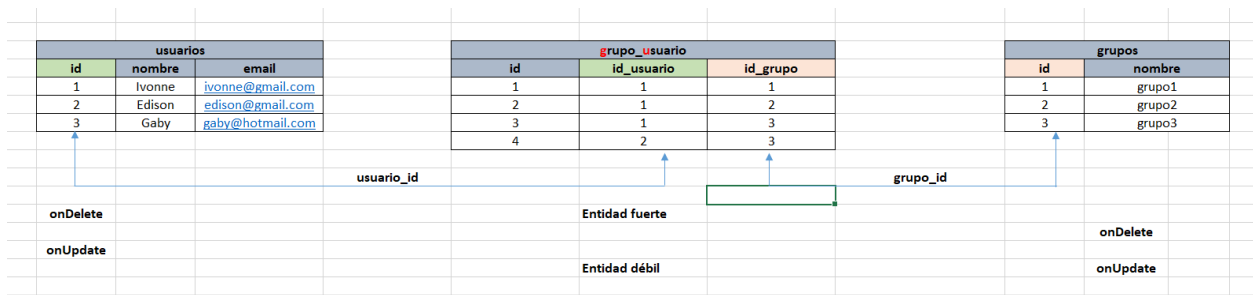
> $post = App\Models\Post::with('category')->find(3)
= App\Models\Post {#4632
  id: 3,
  nombre: "el desarrollo de React",
  category_id: 2,
  user_id: 2,
  created_at: "2021-10-11 00:00:00",
  updated_at: "2021-10-11 00:00:00",
  category: App\Models\Category {#4668
    id: 2,
    nombre: "frontend",
    created_at: "2021-10-11 00:00:00",
    updated_at: "2021-10-11 00:00:00",
  },
}
>

```

Realizar mas verificaciones.

## RELACIÓN DE MUCHOS A MUCHOS

<https://laravel.com/docs/9.x/eloquent-relationships#many-to-many>



18. Crear un nuevo modelo llamado **Group** con su respectiva migración, factory y seeder con el siguiente comando: **php artisan make:model Group -mfs**

```

1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7
8 class Group extends Model
9 {
10     use HasFactory;
11 }
12
  
```

```

PS C:\Users\IvonneSQ05\Laboratorios Web 2022B\Relaciones> php artisan make:model Group -mfs

INFO Model [C:\Users\IvonneSQ05\Laboratorios Web 2022B\Relaciones\app\Models\Group.php] created successfully.
INFO Factory [C:\Users\IvonneSQ05\Laboratorios Web 2022B\Relaciones\database\factories\GroupFactory.php] created successfully.
INFO Migration [C:\Users\IvonneSQ05\Laboratorios Web 2022B\Relaciones\database\migrations\2023_01_03_213726_create_groups_table.php] created successfully.
INFO Seeder [C:\Users\IvonneSQ05\Laboratorios Web 2022B\Relaciones\database\seeders\GroupSeeder.php] created successfully.

PS C:\Users\IvonneSQ05\Laboratorios Web 2022B\Relaciones>
  
```

19. Definir los campos (nombre de las columnas) para la tabla de la base de datos, esto se lo realiza en la migración del modelo respectivo.

**NOTA: Un usuario puede estar en varios grupos y que un grupo tiene varios usuarios**

```

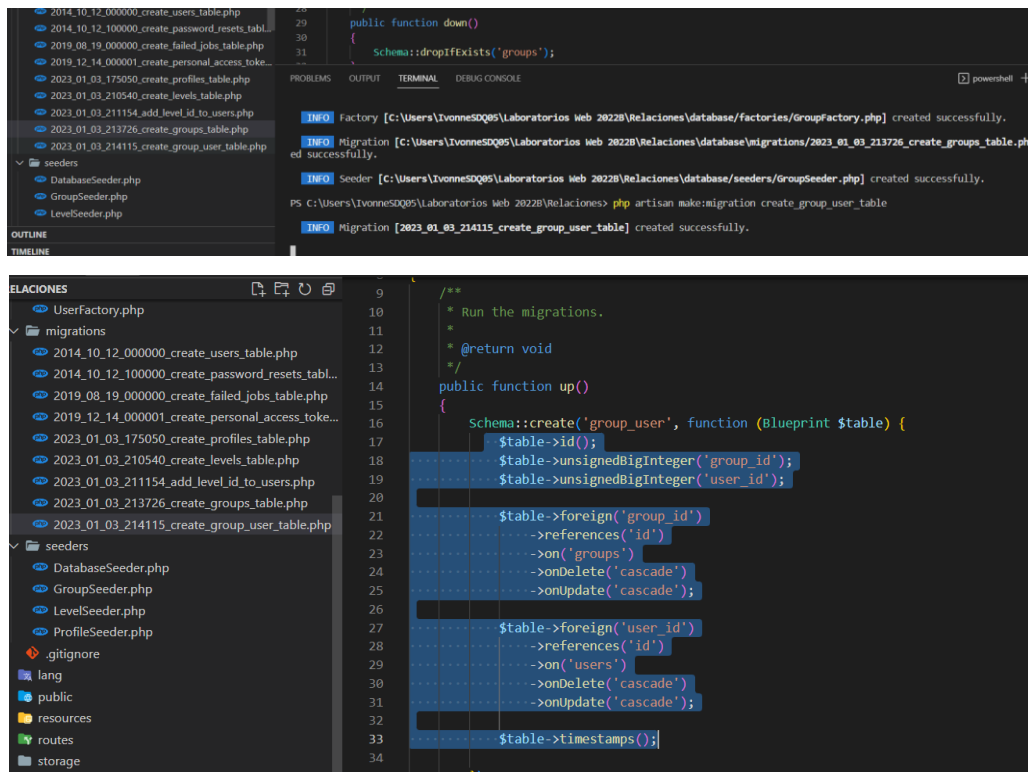
9 /**
10  * Run the migrations.
11  *
12  * @return void
13  */
14 public function up()
15 {
16     Schema::create('groups', function (Blueprint $table) {
17         $table->id();
18         $table->string('nombre');
19         $table->timestamps();
20     });
21 }
22
23 /**
24  * Reverse the migrations.
25  *
26  * @return void
27  */
28
  
```

### Código

```
$table->id();
$table->string('nombre');
$table->timestamps();
```

20. Se debe tener en cuenta que si se trata de una relación de muchos a muchos se debe trabajar con una tabla intermedia o tabla pivote que haga referencia a las dos tablas que son usuarios y grupos. Para lo cual se debe crear una nueva migración, ejecutando el siguiente comando: **php artisan make:migration create\_group\_user\_table**

<https://laravel.com/docs/9.x/eloquent-relationships#many-to-many-table-structure>



## Código

```
$table->id();
$table->unsignedBigInteger('group_id');
$table->unsignedBigInteger('user_id');
$table->foreign('group_id')
    ->references('id')
    ->on('groups')
    ->onDelete('cascade')
    ->onUpdate('cascade');
$table->foreign('user_id')
    ->references('id')
    ->on('users')
    ->onDelete('cascade')
    ->onUpdate('cascade');

$table->timestamps();
```

21. Ahora se debe hacer el procedimiento a nivel de **Eloquent de Laravel**.

**NOTA: Un usuario puede estar en varios grupos y que un grupo tiene varios usuarios**

```

47     return $this->hasOne(Profile::class);
48 }
49
50 // RELACIÓN DE UNO A MUCHOS
51 public function level()
52 {
53     return $this->belongsTo(Level::class);
54 }
55
56 // RELACIÓN DE MUCHOS A MUCHOS
57 public function groups()
58 {
59     return $this->belongsToMany(Group::class)->withTimestamps();
60 }
61
62
63

```

```

5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7
8 class Group extends Model
9 {
10     use HasFactory;
11     // RELACIÓN DE MUCHOS A MUCHOS
12     public function users()
13     {
14         return $this->belongsToMany(User::class)->withTimestamps();
15     }
16 }
17
18

```

### Código

```

// RELACIÓN DE MUCHOS A MUCHOS
public function groups()
{
    return $this->belongsToMany(Group::class)->withTimestamps();
}

```

```

// RELACIÓN DE MUCHOS A MUCHOS
public function users()
{
    return $this->belongsToMany(User::class)->withTimestamps();
}

```

22. Ahora se procede a ejecutar la migración y verificar si realmente esta correcta la implementación de la relación muchos a muchos, ejecutando el comando: **php artisan migrate**

```

PS C:\Users\IvonneSDQ05\Laboratorios Web 2022B\Relaciones> php artisan migrate

INFO Running migrations.

2023_01_03_213726_create_groups_table ..... 40ms DONE
2023_01_03_214115_create_group_user_table ..... 30ms DONE

PS C:\Users\IvonneSDQ05\Laboratorios Web 2022B\Relaciones>

```

### Código

```

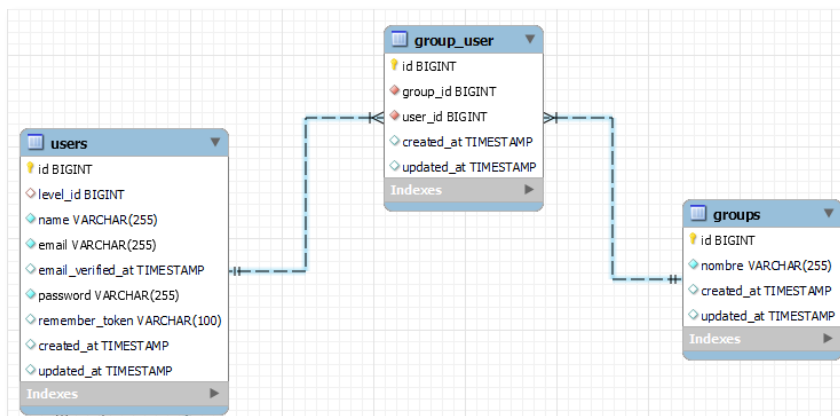
INSERT INTO `relaciones`.`groups` (`id`, `nombre`, `created_at`, `updated_at`)
VALUES ('1', 'grupo-1', '2021-10-11 00:00:00', '2021-10-11 00:00:00');
INSERT INTO `relaciones`.`groups` (`id`, `nombre`, `created_at`, `updated_at`)
VALUES ('2', 'grupo-2', '2021-10-11 00:00:00', '2021-10-11 00:00:00');
INSERT INTO `relaciones`.`groups` (`id`, `nombre`, `created_at`, `updated_at`)
VALUES ('3', 'grupo-3', '2021-10-11 00:00:00', '2021-10-11 00:00:00');

```

```

INSERT INTO `relaciones`.`group_user` (`id`, `group_id`, `user_id`, `created_at`,
`updated_at`) VALUES ('1', '1', '3', '2021-10-11 00:00:00', '2021-10-11 00:00:00');
INSERT INTO `relaciones`.`group_user` (`id`, `group_id`, `user_id`, `created_at`,
`updated_at`) VALUES ('2', '1', '3', '2021-10-11 00:00:00', '2021-10-11 00:00:00');
INSERT INTO `relaciones`.`group_user` (`id`, `group_id`, `user_id`, `created_at`,
`updated_at`) VALUES ('3', '2', '2', '2021-10-11 00:00:00', '2021-10-11 00:00:00');
INSERT INTO `relaciones`.`group_user` (`id`, `group_id`, `user_id`, `created_at`,
`updated_at`) VALUES ('4', '2', '2', '2021-10-11 00:00:00', '2021-10-11 00:00:00');
INSERT INTO `relaciones`.`group_user` (`id`, `group_id`, `user_id`, `created_at`,
`updated_at`) VALUES ('5', '3', '1', '2021-10-11 00:00:00', '2021-10-11 00:00:00');
INSERT INTO `relaciones`.`group_user` (`id`, `group_id`, `user_id`, `created_at`,
`updated_at`) VALUES ('6', '3', '1', '2021-10-11 00:00:00', '2021-10-11 00:00:00');

```



### 23. Verificar ejecutando: **php artisan tinker**

```

> $users = App\Models\Group::with('users')->find(2)
= App\Models\Group {#4672
  id: 2,
  nombre: "grupo-2",
  created_at: "2021-10-11 00:00:00",
  updated_at: "2021-10-11 00:00:00",
  users: Illuminate\Database\Eloquent\Collection {#4678
    all: [
      App\Models\User {#4682
        id: 2,
        level_id: 2,
        name: "Edison",
        email: "edison@hotmail.com",
        email_verified_at: "2023-01-01 13:19:25",
        #password: "sistemas",
        #remember_token: "sistemas",
        created_at: null,
        updated_at: null,
        pivot: Illuminate\Database\Eloquent\Relations\Pivot {#4681
          group_id: 2,
          user_id: 2,
          created_at: "2021-10-11 00:00:00",
          updated_at: "2021-10-11 00:00:00",
        },
      },
    ],
  },
  App\Models\User {#4683
    id: 2,
    level_id: 2,
    name: "Edison",
    email: "edison@hotmail.com",
    email_verified_at: "2023-01-01 13:19:25",
    #password: "sistemas",
    #remember_token: "sistemas",
    created_at: null,
    updated_at: null,
    pivot: Illuminate\Database\Eloquent\Relations\Pivot {#4680
      group_id: 2,
      user_id: 2,
      created_at: "2021-10-11 00:00:00",
      updated_at: "2021-10-11 00:00:00",
    },
  },
],
}

```

```

> $groups = App\Models\User::with('groups')->find(1)
= App\Models\User {#4693
  id: 1,
  level_id: 1,
  name: "Ivonne",
  email: "ivonne@hotmail.com",
  email_verified_at: "2023-01-01 13:19:25",
  #password: "sistemas",
  #remember_token: "sistemas",
  created_at: null,
  updated_at: null,
  groups: Illuminate\Database\Eloquent\Collection {#4684
    all: [
      App\Models\Group {#4697
        id: 3,
        nombre: "grupo-3",
        created_at: "2021-10-11 00:00:00",
        updated_at: "2021-10-11 00:00:00",
        pivot: Illuminate\Database\Eloquent\Relations\Pivot {#4696
          user_id: 1,
          group_id: 3,
          created_at: "2021-10-11 00:00:00",
          updated_at: "2021-10-11 00:00:00",
        },
      },
      App\Models\Group {#4698
        id: 3,
        nombre: "grupo-3",
        created_at: "2021-10-11 00:00:00",
        updated_at: "2021-10-11 00:00:00",
        pivot: Illuminate\Database\Eloquent\Relations\Pivot {#4695
          user_id: 1,
          group_id: 3,
          created_at: "2021-10-11 00:00:00",
          updated_at: "2021-10-11 00:00:00",
        },
      },
    ],
  },
},
}

```

```
> use App\Models\User
> $user = User::find(1)
= App\Models\User {#3686
  id: 1,
  level_id: 1,
  name: "Ivonne",
  email: "ivonne@hotmail.com",
  email_verified_at: "2023-01-01 13:19:25",
  #password: "sistemas",
  #remember_token: "sistemas",
  created_at: null,
  updated_at: null,
}

> $user -> groups()->attach([1,2])
= null
```

## TAREA

Crear 4 relaciones adicionales a las vistas en clase, las cuales están divididas de la siguiente manera:

- 1 relación de uno a uno
- 2 relaciones de uno a muchos
- 1 relación de muchos a muchos

Recuerda llenar con datos cada una de las tablas en la Base de Datos y hacer consultas con tinker.

Referencias bibliográficas

- <https://geekflare.com/es/laravel-eloquent-model-relationship/>
- <https://laravel.com/docs/9.x/eloquent-relationships>
- <https://www.oulub.com/docs/laravel/es-es/eloquent#retrieving-models>

## PRESENTACIÓN

Al finalizar tu tarea deberás subir:

- Un archivo en formato pdf con el nombre (Laboratorio8\_ AWeb\_2022B \_NApellido)

## RECURSOS NECESARIOS

- Visual Studio Code
- Herramienta ofimática
- Internet
- Material de clase