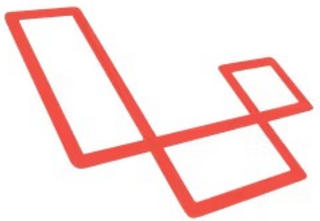
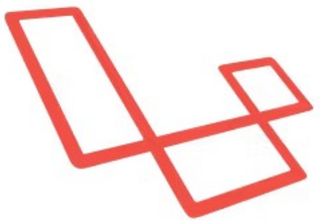
UD 3.3

Relaciones



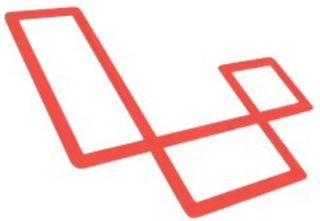
Relaciones en Eloquent

Eloquent simplifica el manejo de relaciones entre tablas en una base de datos relacional. Permite definir y trabajar con asociaciones entre modelos mediante métodos predefinidos, facilitando consultas complejas y operaciones relacionales.



Tipos de relaciones en Eloquent

1. **One-to-One (Uno a Uno)**
2. **One-to-Many (Uno a Muchos)**
3. **Many-to-Many (Muchos a Muchos)**
4. **Has-Many-Through**



1. One-to-One (Uno a Uno)

Definición: Un registro en una tabla está relacionado con exactamente un registro en otra tabla.

Un `User` tiene un `Profile`.

- **Modelo User:**

  ```
  public function perfil() { return $this->hasOne(Perfil::class); }
  ```

- **Modelo Perfil:**

  ```
  public function usuario() { return $this->belongsTo(User::class); }
  ```

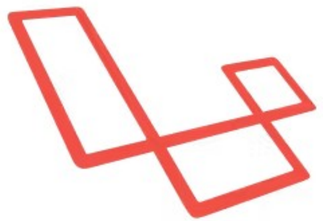
Uso:

- Obtener el perfil de un usuario:

  ```
  $perfil = User::find(1)->perfil;
  ```

- Obtener el usuario al que pertenece un perfil:

  ```
  $usuario = Perfil::find(1)->usuario;
  ```



2. One-to-Many (Uno a Muchos)

Definición: Un registro en una tabla está relacionado con múltiples registros en otra tabla.

Un `User` tiene muchos `Post`.

- **Modelo User:**

  `public function posts() { return $this->hasMany(Post::class); }`
- **Modelo Post:**

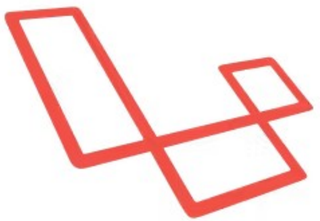
  `public function usuario() { return $this->belongsTo(User::class); }`

Uso:

- Obtener los posts de un usuario

  `$posts = User::find(1)->posts;`
- Obtener el usuario que escribió un post:

  `$usuario = Post::find(1)->usuario;`



3. Many-to-Many (Muchos a Muchos)

Definición: Múltiples registros en una tabla pueden estar relacionados con múltiples registros en otra tabla.

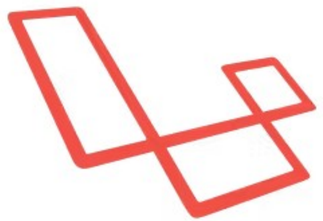
Un `Post` puede tener muchas `Categorías`, y una `Categoría` puede tener muchos `Posts`.

- **Modelo Post:**

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

- **Modelo Categoria:**

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

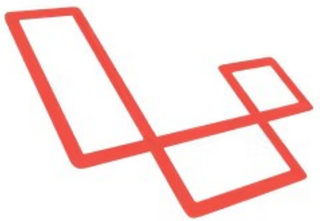


3. Many-to-Many (Muchos a Muchos)

Migración de la tabla pivote:

Crea una tabla pivote `categoria_post`:

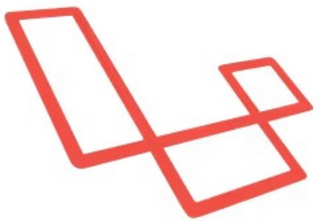
```
Schema::create('categoria_post', function (Blueprint $table) {
    $table->id();
    $table->unsignedBigInteger('post_id');
    $table->unsignedBigInteger('categoria_id');
    $table->timestamps();
});
```



3. Many-to-Many (Muchos a Muchos)

Uso:

- Obtener categorías de un post:
  `$categorias = Post::find(1)->categorias;`
- Obtener posts de una categoría:
  `$posts = Categoria::find(1)->posts;`



4. Has-Many-Through

Definición: Permite acceder a una relación "a través de" otra relación intermedia.

Ejemplo:

Un `Country` tiene muchos `Posts` a través de `Users`.

- **Modelo Country:**

```
public function posts() {
    return $this->hasManyThrough(Post::class, User::class);
}
```

- **Modelo User:** (relación intermedia)

```
public function posts() { return $this->hasMany(Post::class); }
```

Uso:

- Obtener todos los posts de un país:

```
$posts = Country::find(1)->posts;
```