

# IRUDI BAT IGO

Eremu berri bat sartzen (Alumnos taulan)

File Storage

Irudia igo (Bista eta kontroladorea)

## Eremu berri bat sartzen (Alumnos taulan)

Irudi bat igotzeko orain arte egindako adibidea erabiliko dugu, horretarako gure datu baseko taula aldatuko dugu "foto" eremu berria gehitzen.

Gogoratu hori egin ahal izateko migrazio berri bat egin behar dugula. Oraingoan ere Laraveleko konbentzioak jarrituko ditugu, eta taulak sortzeko "create\_table" esaten genuen moduan, oraingoan eremu bat gehitzeko, ondoren adierazten dugun moduan egingo dugu:

```
php artisan make:migration add_foto_to_alumnos_table
```

Ikusten dugunez, migrazioa prest dago dagokion eremua gehitzeko. *up()* metodoan gehituko dugu eta *down()* metodoan ezabatu.

```
$table->string('foto')->nullable()->after('direccion');
```

Honegaz esaten diogu, "foto" eremua gehitzeko, nullable() dela esaterakoan null balioak izan ditzakeela esaten dugu ere, horrela taulak datuak izatean jada, ez du akatsik emango, zuzenean null jarriko bait dio. Eta azkenik after('direccion') jartzerakoan, adierazten diogu, direccion eremua ostean gehitzeko, create\_at eta update\_at baino lehen.

Down() metodoan, alderantzizko ekintza zehazten dugu; kasu horretan, ezabatu up() metodoan gehitu dugun zutabea:

```
$table->dropColumn('foto');
```

```
return new class extends Migration{

    public function up(): void{
        Schema::table('alumnos', function (Blueprint $table) {
            $table->string('foto')->nullable()->after('direccion');
        });
    }

    public function down(): void{
        Schema::table('alumnos', function (Blueprint $table) {
            $table->dropColumn('foto');
        });
    }
};
```

Migrazioa exekutatzeko, azkenekoa exekutatu du bakarrik, hau da, orain arte exekutatu ez den bakarra:

```
php artisan migrate
```

## File Storage

Laravel dokumentazioa: <https://laravel.com/docs/master/filesystem>

Flysystem integrazioak, ematen dio Laraveleri fitxategiekin lan egiteko kontroladore pilo. Storage-k emango digu aukera beraz fitxategiekin lan egiteko.

Konfigurazioa **config/filesystems.php** fitxategian egiten dira. Hemen topatuko ditugu diska desberdinak, honek balio digu hainbat sistema ezberdinekin lan egiteko.

Disko bakoitzak biltegitratze-kontrolatzaile bat eta biltegitratze-kokaleku partikular bat irudikatzen ditu.

Gure fitxategian, default moduan locale diska erabiliko duela esaten du, baina gul hori public izatera aldatuko dugu:

```
'default' => env('FILESYSTEM_DISK', 'public'),
```

Orain gauza izango litzateke jakitea non gorde behar ditugun gure irudiak. Ba hasiera baten gu edozein lekutan gorde ditzakegu, baina berez Laravelek badu prestatuta "storage" karpeta gure fitxategian gordetzeko, beraz hor gordeko ditugu guk ere. Eta storage barruan app/public karpeta barruan.

Baina horregaz arazo bat sortzen zaigu, zeren eta guk dakigun bezala, nabigatzaileek public karpeta irakurri dezake soilik. Hori konpontzeko badago aukera bat, eta izango da gure public karpetan storage karpetaren "acceso directo" bat sortzea. Hori gure terminalaren bidez egin dezakegu.

```
php artisan storage:link
```

## Irudia igo (Bista eta kontroladorea)

Lehenengo eta behin "alumnos.create" bistan fitxategia igotzeko aukera emango dugu:

```
Argazkia: <input type="file" name="foto" id="foto" accept="image/*">
```

Konturatu zarienez, hemen, accept-en esan diogu irudiak soilik igo ditzakeela, baina honegaz bakarrik ez da nahiko izango, kontroladorean ere balidazioa egin beharko genuke. Gure kasuan StoreAlumno.php-en egingo dugu:

```
public function rules(): array{
    return [
        'nombre' => 'required|min:5',
        'edad' => 'required',
        'foto' => 'image'
    ];
}
```

Gauza biak idatzita, ziurtatuko dugu gure datu basean irudian bakarrik igoko dituztela.

#### AlumnosController-en:

Storage fakade-ri esker fitxategia errez igo dezakegu, hurrengo kodea idatzi beharko dugu hori egin ahal izateko, ikusi dezakezuenek guk put:

```
if ($request->file('foto')){
    $url = Storage::putFile('public/alumnos', $request->file('foto'));
}
$alumno->foto = $url;
```

Bistan, gure irudia ikusteko url() metodoa ereabiliko dugu:

```

```