

LARAVEL II - Migraciones

Titular	MARIA JOSE SALINAS PARRA
Etiquetas	Laravel
Autor	Henry Gutierrez Maho Salinas

Hola! 🦖 Esperamos que esta guía te sirva para hacer migraciones con Laravel, te prometemos que lo aprenderás fácilmente. No somos profesionales, así que si cometimos algún error, por favor, háznoslo saber. 🐥 Por cierto, si quieres ayudarnos a mejorar la guía lo agradeceríamos muchísimo ✨ - Maho y Henry



Créditos sobre migración inversa: Ing. Rodmy Orellana

ÍNDICE

ÍNDICE

[Antes de empezar](#)

[Migración Inversa](#) 🔄

1. Importación de un esquema ya creado

2. Instalación del importador

Migración manual

Antes de empezar

Hay varias formas de migrar, se puede hacer manualmente o con una migración inversa. Para hacer la segunda opción (que además es la más rápida), debes tener **Node.JS** puedes descargarlo en [Node.js — Run JavaScript Everywhere \(nodejs.org\)](#). Dale click en *Download Node.js (LTS)*.



Tendrás el Wizard de instalación. Todo next, luego install. Para comprobar que está instalado:

```
node --version
```

```
on Selecciconar Smbolo del sistema
Microsoft Windows [Versi3n 10.0.19045.4291]
(c) Microsoft Corporation. Todos los derechos reservados.

:\Users\Henry24>node --version
v20.14.0

:\Users\Henry24>
```

Migraci3n Inversa

1. Importaci3n de un esquema ya creado

Este proceso se puede realizar a trav3s de un generador de migraci3n, lo utilizaremos en el siguiente repositorio: <https://github.com/bennett-treptow/laravel-migration-generator>

2. Instalaci3n del importador

Primero utilizamos el siguiente comando en consola para el generador de migraciones

```
composer require --dev bennett-treptow/laravel-migration-gene
```

Instalamos el importador con los siguientes comando:

```
php artisan vendor:publish --provider="LaravelMigrationGenera
```

Con ese comando obtendremos el siguiente resultado:

```

PS C:\xampp\htdocs\nn-merch-app> composer require --dev bennett-treptow/laravel-migration-generator
./composer.json has been updated
Running composer update bennett-treptow/laravel-migration-generator
Loading composer repositories with package information
Updating dependencies
Lock file operations: 2 installs, 0 updates, 0 removals
  - Locking bennett-treptow/laravel-migration-generator (4.4.1)
  - Locking marcj/topsort (2.0.0)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 2 installs, 0 updates, 0 removals
  - Downloading marcj/topsort (2.0.0)
  - Downloading bennett-treptow/laravel-migration-generator (4.4.1)
  - Installing marcj/topsort (2.0.0): Extracting archive
  - Installing bennett-treptow/laravel-migration-generator (4.4.1): Extracting archive
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi

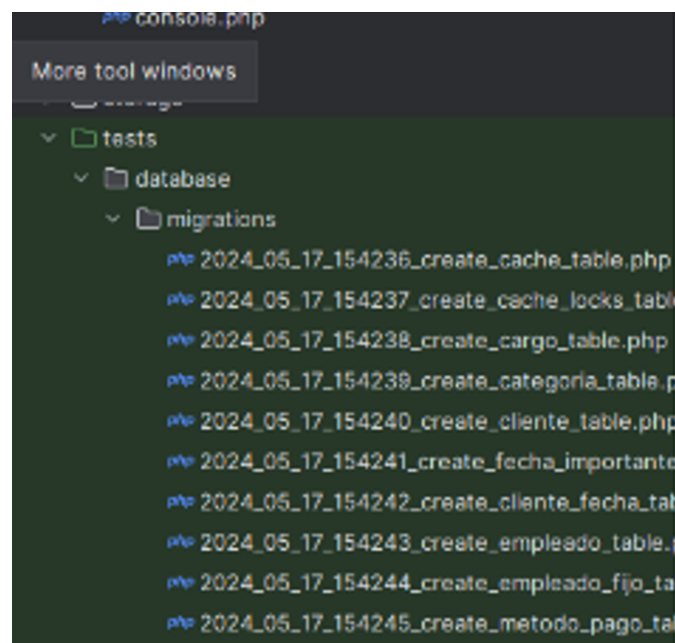
```

Seguido de esto procedemos a ejecutar el siguiente comando:

Con estos pasos ya tenemos instalado el migrador de la base de datos, por lo que procedemos a ejecutar el siguiente comando para ejecutar la migración

```
php artisan generate:migrations
```

Con esto se habrán generado las migraciones de todas nuestras tablas de nuestro modelo E-R



- Para cambiar la ruta de las migraciones debemos ejecutar este comando:

```
php artisan generate:migrations --path=database/migrations
```

- Para eliminar los archivos de migración utilizamos el siguiente comando:

```
php artisan generate:migrations --empty-path
```

💡 **NOTA:** Si quieres cambiar algo de tu base de datos, tendrás que hacer todo el proceso? bueno si es algo pequeño, como agregar una columna a una tabla, por suerte no. Lo único que debes hacer es añadir el cambio en tu base de datos mediante comandos o un DBMS y agregar estos mismos cambios en tus migraciones.

Por ejemplo, si quiero añadir un nuevo campo en mi tabla clientes, lo haría aquí:

```
public function up()
{
    Schema::create( table: 'cliente', function (Blueprint $table) {
        $table->integer( column: 'id')->primary();
        $table->string( column: 'nombre', length: 30);
        $table->string( column: 'correo', length: 30);
        $table->integer( column: 'telefono');
        $table->string( column: 'categoria', length: 30);
        $table->string( column: 'tipo_cliente', length: 15);
        $table->string( column: 'usuario', length: 20)->nullable();
        $table->string( column: 'contraseña', length: 20)->nullable();
        $table->boolean( column: 'habilitado')->default( value: 1);
    });
}

/**
 * Reverse the migrations.
 *
 * @return void
 */
public function down()
{
    Schema::dropIfExists('cliente');
}
```

Migración manual

completaremos esta parte cuando sepamos la información adicional 😊 por ahora puedes ver más aquí: [make migration artisan command parameters in laravel - TutsForWeb](#)