

1. Creación de una Migración:

Las migraciones en Laravel se utilizan para definir y modificar la estructura de la base de datos. Puedes crear una migración utilizando el comando Artisan:

```
php artisan make:migration nombre_de_la_migracion
```

Por ejemplo, para crear una migración para crear una tabla llamada "productos", puedes usar:

```
php artisan make:migration create_productos_table
```

Esto creará un archivo de migración en el directorio **database/migrations** con un nombre similar a **2023_08_29_000000_create_productos_table.php**. La fecha y hora en el nombre del archivo indican cuándo se creó la migración.

2. Definición de una Migración:

Abre el archivo de migración generado y utiliza los métodos proporcionados por Laravel para definir la estructura de la tabla. Aquí hay un ejemplo de cómo podría verse una migración para crear la tabla "productos":

```
use Illuminate\Database\Migrations\Migration;
```

```
use Illuminate\Database\Schema\Blueprint;
```

```
use Illuminate\Support\Facades\Schema;
```

```
class CreateProductosTable extends Migration  
{  
  
    public function up()  
    {  
        Schema::create('productos', function (Blueprint $table) {  
            $table->id();  
            $table->string('nombre');  
            $table->text('descripcion');  
            $table->decimal('precio', 8, 2);  
            $table->timestamps();  
        });  
    }  
  
    public function down()  
    {  
        Schema::dropIfExists('productos');  
    }
```

```
}
```

```
}
```

En este ejemplo:

- **up()**: Define la estructura de la tabla que se creará. En este caso, estamos creando una tabla "productos" con columnas para el nombre, la descripción, el precio y las marcas de tiempo (**created_at** y **updated_at**).
- **down()**: Define cómo se deshará la migración. En este caso, simplemente eliminamos la tabla "productos" si necesitamos revertir la migración.

3. Ejecución de Migraciones:

Para ejecutar todas las migraciones pendientes y crear la estructura de la base de datos, ejecuta el siguiente comando:

php artisan migrate

Laravel ejecutará todas las migraciones en orden y creará las tablas especificadas en ellas.

4. Revertir Migraciones:

Si necesitas revertir la última migración que ejecutaste, puedes utilizar el siguiente comando:

php artisan migrate:rollback

También puedes especificar el número de pasos que deseas revertir:

php artisan migrate:rollback --step=1

5. Otros Comandos de Migración:

- **php artisan migrate:status**: Muestra el estado actual de todas las migraciones.
- **php artisan migrate:refresh**: Revierte y luego vuelve a ejecutar todas las migraciones.
- **php artisan migrate:reset**: Revierte todas las migraciones.

6. Personalización de Migraciones:

Puedes personalizar aún más tus migraciones para agregar índices, claves foráneas, restricciones y otros elementos que necesites en tu base de datos.

Estas son las bases para trabajar con migraciones en Laravel. Las migraciones son una parte fundamental del flujo de desarrollo de Laravel, ya que permiten mantener y actualizar la estructura de la base de datos de manera controlada y consistente a lo largo del tiempo.