

Imagina crear un pequeño **"Gestor de Tareas"** (una To-Do List). Necesitamos decirle a la base de datos cómo será la tabla donde guardaremos esas tareas.

Abre tu terminal en la carpeta de tu proyecto Laravel y sigue esta secuencia:

1. Generar la migración de tareas:

Primero, le pedimos a Laravel que nos cree el archivo en blanco. En la terminal escribe:

Bash

```
php artisan make:migration create_tareas_table
```

Notarás que el archivo se crea en la carpeta database/migrations/ con la fecha y hora actual en el nombre.

2. Definir las columnas de la tabla:

Abre el archivo que se acaba de crear. Vamos a modificar el método up() para decirle a la base de datos qué información lleva una tarea (un título, una descripción, si está completada o no, y una fecha límite):

PHP

```
public function up(): void
{
    Schema::create('tareas', function (Blueprint $table) {
        $table->id(); // ID único automático

        $table->string('titulo'); // Un texto corto (VARCHAR)
        $table->text('descripcion')->nullable(); // Texto largo, puede quedar vacío
        $table->boolean('completada')->default(false); // Falso por defecto
        $table->date('fecha_limite')->nullable();

        $table->timestamps(); // Crea 'created_at' y 'updated_at'
    });
}
```

3. Ejecutar la migración:

Hasta ahora, esto solo es código PHP. Para que Laravel vaya a tu base de datos (MySQL, SQLite, etc.) y construya la tabla de verdad, ejecuta:

Bash

php artisan migrate

Si revisas tu base de datos ahora, verás la tabla tareas creada exactamente como la pediste.

4. ¡Ups! Olvidamos un campo: El flujo de trabajo en la vida real.

Imagina que al día siguiente tu profesor te dice: *"Faltó añadir un campo para saber la prioridad de la tarea"*.

Como la regla de oro es **no editar migraciones pasadas**, creamos una nueva migración solo para añadir esa columna:

Bash

php artisan make:migration add_prioridad_to_tareas_table

5. Añadir la nueva columna:

Abre este nuevo archivo. Fíjate que ahora dice Schema::table en lugar de Schema::create, porque la tabla ya existe. Añade tu columna allí:

PHP

```
public function up(): void
```

```
{
```

```
    Schema::table('tareas', function (Blueprint $table) {
```

```
        // Añadimos la columna 'prioridad' justo después del 'titulo'
```

```
        $table->string('prioridad')->default('baja')->after('titulo');
```

```
    });
```

```
}
```

```
public function down(): void
```

```
{
```

```
    Schema::table('tareas', function (Blueprint $table) {
```

```
        // El método down debe deshacer exactamente lo que hizo el up
```

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

```
    });
```

```
}
```

6. Aplicar el último cambio:

Vuelve a correr el comando de migración en tu terminal:

Bash

`php artisan migrate`

Laravel es inteligente: sabe que la tabla original ya se creó, así que solo ejecutará esta nueva migración para añadir la columna.

Ya tienes tu base de datos lista. El siguiente paso natural en Laravel, una vez que tienes tu tabla, es crear un "Modelo" (Model) para poder guardar y leer datos de esa tabla usando PHP en lugar de escribir código SQL.