

Práctica 1: Instalación de Laravel y Creación de una Tabla de Libros

Alumno: Daniel Aquilino Ndong Bibang Eyang
Master Fullstack web developer

1. Una vez instalado [Xamp](#) y [Composer](#) en nuestra máquina comprobamos si composer ya se encuentra en correctamente instalada en nuestro equipo de ésta forma:



The screenshot shows the Visual Studio Code interface. The 'Extension: Composer' panel is open, displaying the Composer extension by DEVSENSE. Below the extension details, the 'TERMINAL' tab is active, showing the command prompt where the 'composer' command was entered. The terminal output displays the Composer logo, the version (2.6.2), and a list of usage options.

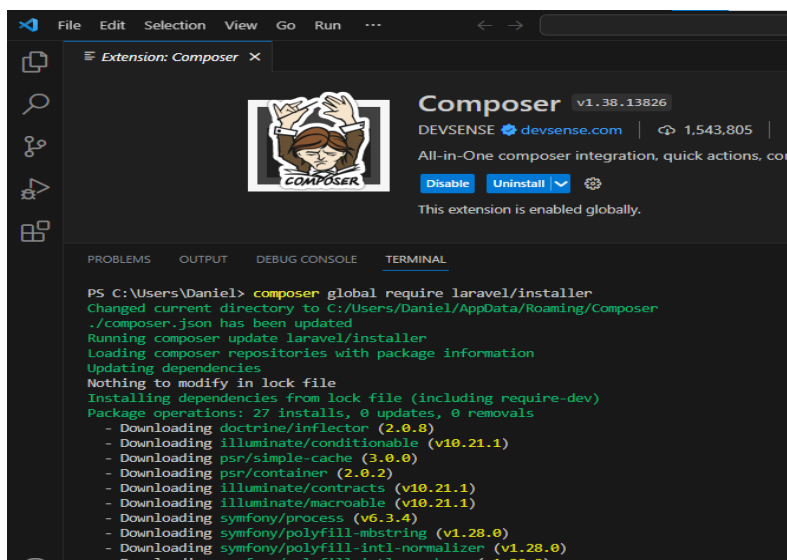
```
PS C:\Users\Daniel> composer

Composer version 2.6.2 2023-09-03 14:09:15

Usage:
  command [options] [arguments]

Options:
  -h, --help                Display help for the given command. When no command is given display help for the list command
  -q, --quiet               Do not output any message
  -V, --version              Display this application version
  --ansi|--no-ansi          Force (or disable --no-ansi) ANSI output
  -n, --no-interaction      Do not ask any interactive question
  --profile                 Display timing and memory usage information
  --no-plugins              Whether to disable plugins.
  --no-scripts              Skips the execution of all scripts defined in composer.json file.
  -d, --working-dir=WORKING-DIR If specified, use the given directory as working directory.
  --no-cache                Prevent use of the cache
```

2. Procedemos a instalar Laravel y para eso introducimos el comando **composer global require laravel/installer**

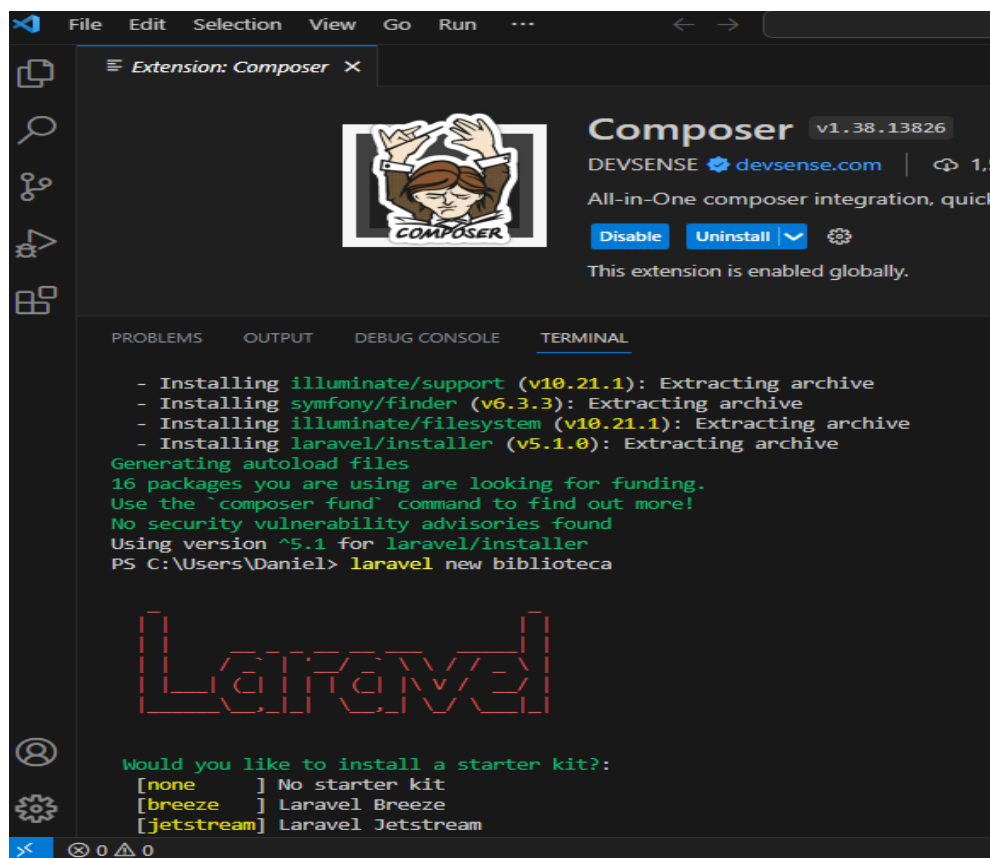


The screenshot shows the Visual Studio Code interface. The 'Extension: Composer' panel is open. Below it, the 'TERMINAL' tab is active, showing the command prompt where the command 'composer global require laravel/installer' was entered. The terminal output shows the command being executed, the current directory being changed, and the installation of the laravel/installer package along with its dependencies.

```
PS C:\Users\Daniel> composer global require laravel/installer
Changed current directory to C:/Users/Daniel/AppData/Roaming/Composer
./composer.json has been updated
Running composer update laravel/installer
Loading composer repositories with package information
Updating dependencies
Nothing to modify in lock file
Installing dependencies from lock file (including require-dev)
Package operations: 27 installs, 0 updates, 0 removals
- Downloading doctrine/inflector (2.0.8)
- Downloading illuminate/conditionable (v10.21.1)
- Downloading psr/simple-cache (3.0.0)
- Downloading psr/container (2.0.2)
- Downloading illuminate/contracts (v10.21.1)
- Downloading illuminate/macroable (v10.21.1)
- Downloading symfony/process (v6.3.4)
- Downloading symfony/polyfill-mbstring (v1.28.0)
- Downloading symfony/polyfill-intl-normalizer (v1.28.0)
```

3. Definimos el nombre de la base de datos en la que trabajaremos.

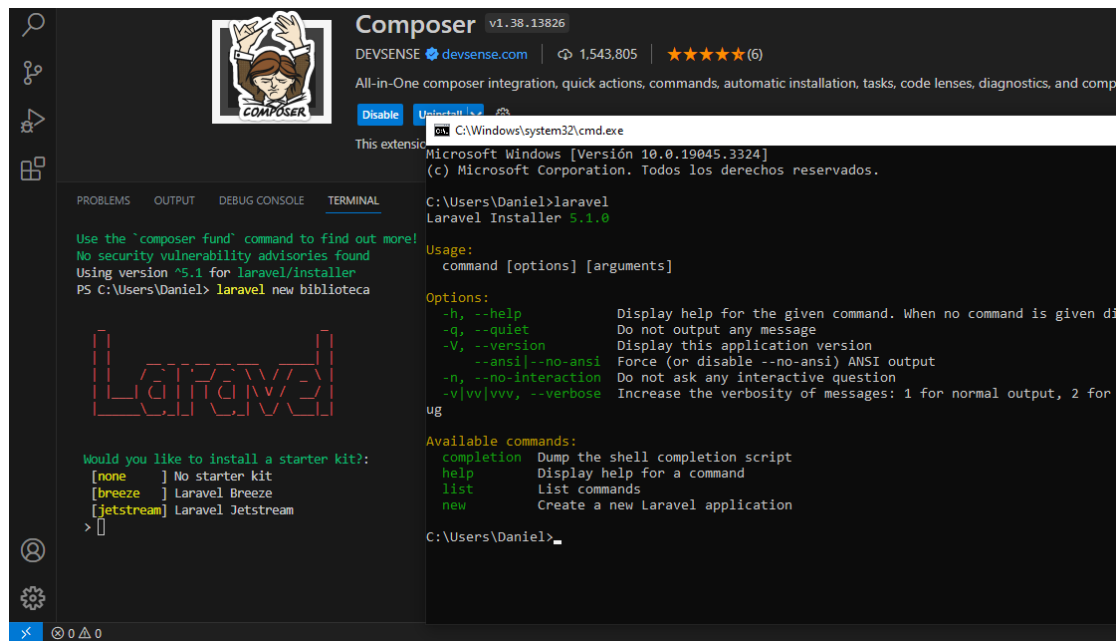
NOTA: En todo caso esto se puede cambiar una vez finalizada la instalación



The screenshot shows the Visual Studio Code interface. At the top, the 'Extension: Composer' panel is open, displaying the Composer logo (a cartoon character with a headband) and the version 'v1.38.13826'. Below the logo, it says 'DEVSENSE devsense.com' and 'All-in-One composer integration, quick'. There are buttons for 'Disable', 'Uninstall', and a settings gear icon. A message at the bottom says 'This extension is enabled globally.' Below the extension panel, the 'TERMINAL' tab is active, showing the output of the 'laravel new' command. The output includes the installation of several packages (illuminate/support, symfony/finder, illuminate/filesystem, and laravel/installer), generating autoload files, and a message about funding. It then shows the command 'laravel new biblioteca' being executed, resulting in a red ASCII art logo for 'Laravel'. Below the logo, it asks 'Would you like to install a starter kit?:' with three options: '[none]' (No starter kit), '[breeze]' (Laravel Breeze), and '[jetstream]' (Laravel Jetstream). The 'jetstream' option is highlighted.

```
File Edit Selection View Go Run ...  
Extension: Composer X  
Composer v1.38.13826  
DEVSENSE devsense.com | 1,5  
All-in-One composer integration, quick  
Disable Uninstall [v]  
This extension is enabled globally.  
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL  
- Installing illuminate/support (v10.21.1): Extracting archive  
- Installing symfony/finder (v6.3.3): Extracting archive  
- Installing illuminate/filesystem (v10.21.1): Extracting archive  
- Installing laravel/installer (v5.1.0): Extracting archive  
Generating autoload files  
16 packages you are using are looking for funding.  
Use the `composer fund` command to find out more!  
No security vulnerability advisories found  
Using version ^5.1 for laravel/installer  
PS C:\Users\Daniel> laravel new biblioteca  
  
Laravel  
  
Would you like to install a starter kit?:  
[none] No starter kit  
[breeze] Laravel Breeze  
[jetstream] Laravel Jetstream  
X 0 0 0
```

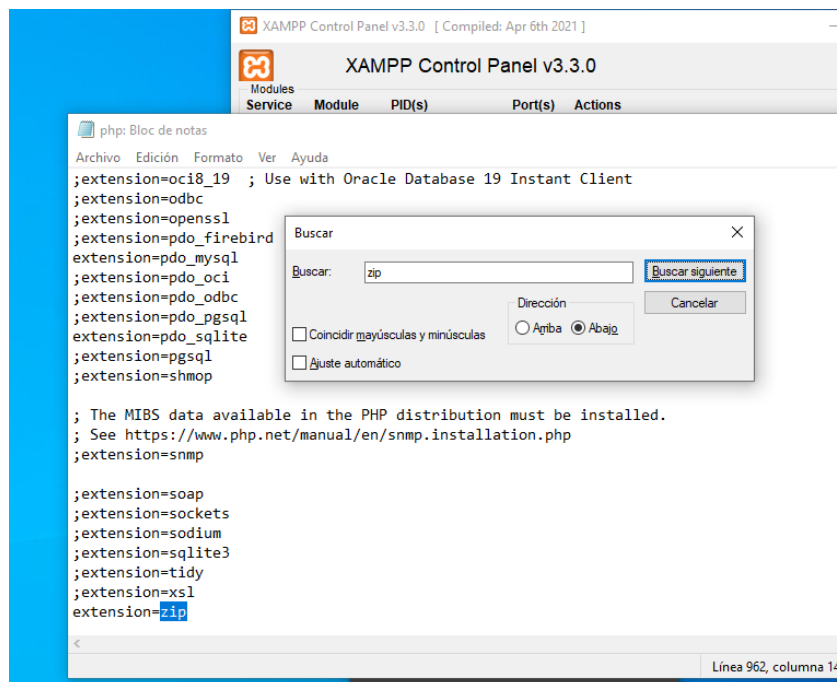
Configuramos nuestro entorno de trabajo usando el kit de instalación que trae Laravel



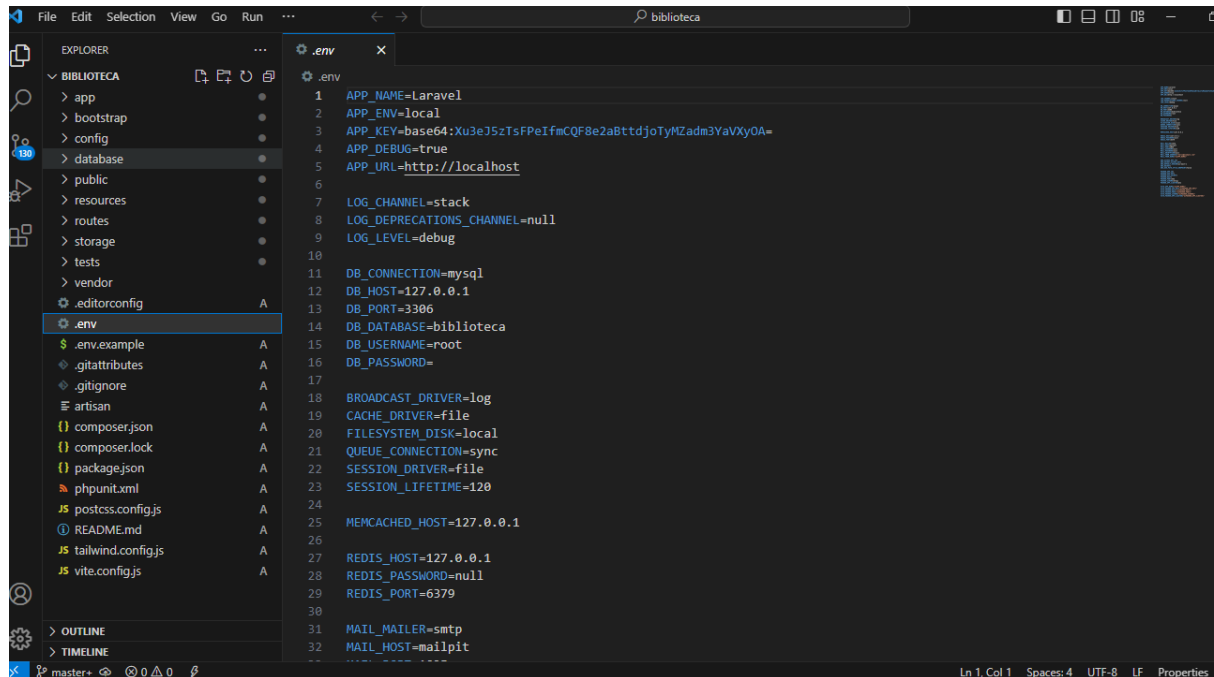
4. Puede que aparezca un error durante la instalación de Laravel:

“git was not found in your PATH...”

Para solucionarlo solo hay que abrir el archivo php.ini de xampp y buscar “;extension=zip”, una vez hallado quitarle el “;” tal y como aparece en la imagen de abajo, guardar y probar de nuevo.



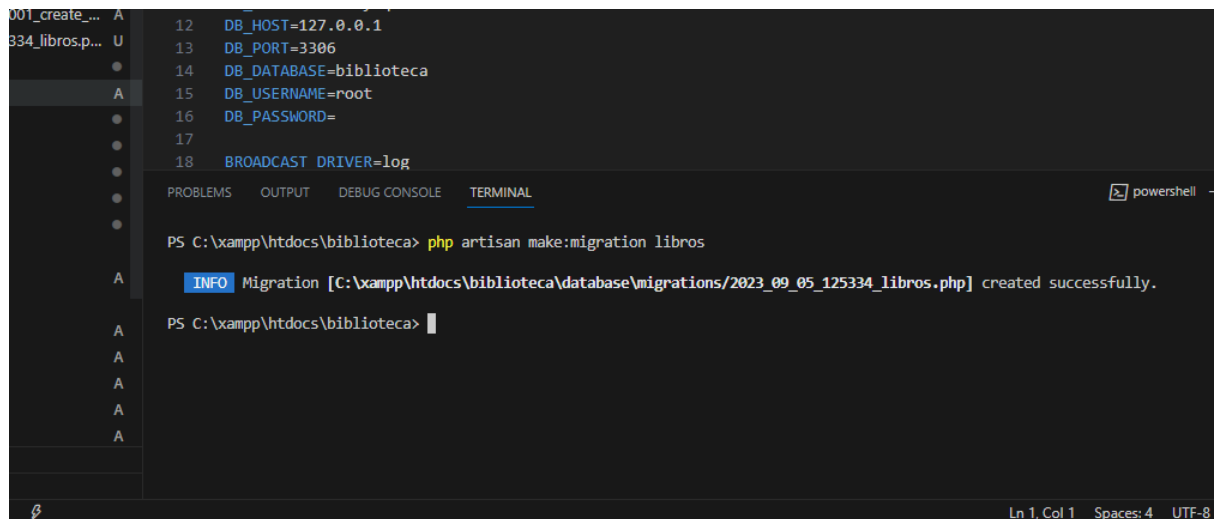
5. Configuración de la base de datos **BIBLIOTECA** en el archivo .env



The screenshot shows the Visual Studio Code editor with the `.env` file open. The file contains the following configuration:

```
1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=base64:Xu3eJ5zTsFPeIfmCQF8e2aBtttdJoTyMZadm3YaVXyOA=
4 APP_DEBUG=true
5 APP_URL=http://localhost
6
7 LOG_CHANNEL=stack
8 LOG_DEPRECATIONS_CHANNEL=null
9 LOG_LEVEL=debug
10
11 DB_CONNECTION=mysql
12 DB_HOST=127.0.0.1
13 DB_PORT=3306
14 DB_DATABASE=biblioteca
15 DB_USERNAME=root
16 DB_PASSWORD=
17
18 BROADCAST_DRIVER=log
19 CACHE_DRIVER=file
20 FILESYSTEM_DISK=local
21 QUEUE_CONNECTION=sync
22 SESSION_DRIVER=file
23 SESSION_LIFETIME=120
24
25 MEMCACHED_HOST=127.0.0.1
26
27 REDIS_HOST=127.0.0.1
28 REDIS_PASSWORD=null
29 REDIS_PORT=6379
30
31 MAIL_MAILER=smtp
32 MAIL_HOST=mailpit
```

6. Creamos la migración para la tabla libros



The screenshot shows the Visual Studio Code editor with the `2023_09_05_125334_libros.php` migration file open. The terminal output shows the command `php artisan make:migration libros` being executed successfully.

```
12 DB_HOST=127.0.0.1
13 DB_PORT=3306
14 DB_DATABASE=biblioteca
15 DB_USERNAME=root
16 DB_PASSWORD=
17
18 BROADCAST_DRIVER=log
```

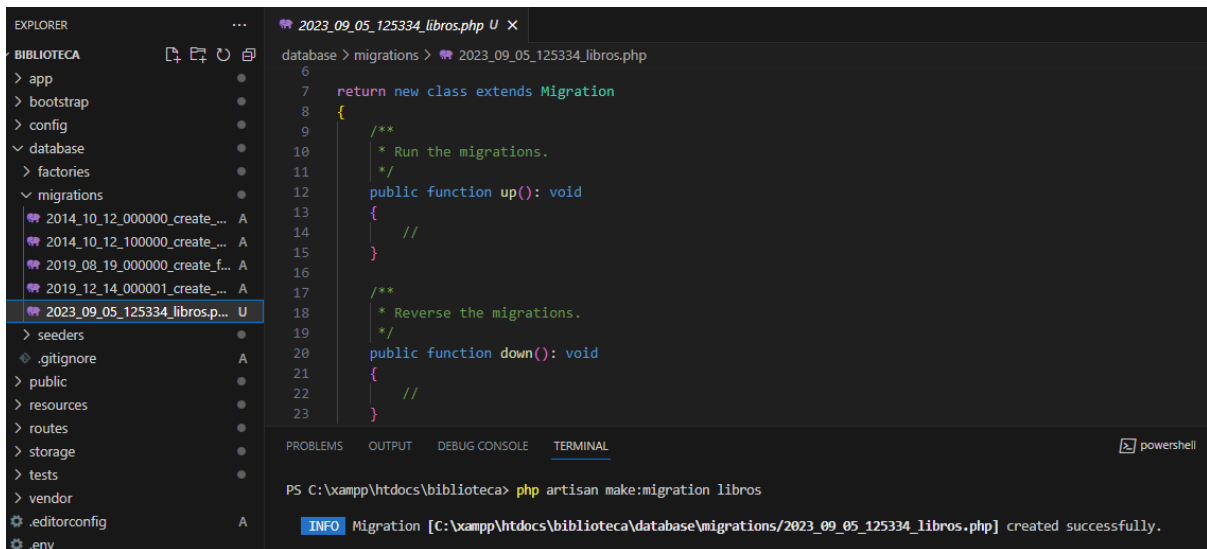
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS C:\xampp\htdocs\biblioteca> php artisan make:migration libros

INFO Migration [C:\xampp\htdocs\biblioteca\database\migrations\2023_09_05_125334_libros.php] created successfully.

PS C:\xampp\htdocs\biblioteca>

- Se nos crea automáticamente un nuevo archivo
2023_09_05_125334_libros.php

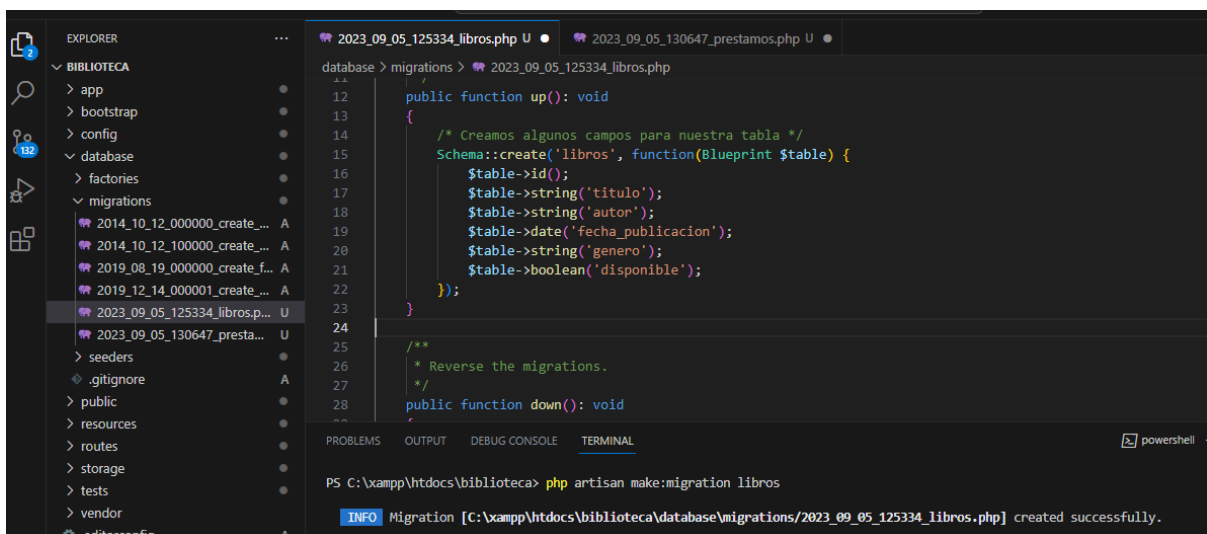


The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar displays the project structure, with the 'migrations' folder expanded. A new file, '2023_09_05_125334_libros.php', has been created and is highlighted. The main editor shows the content of this file, which is a Laravel migration class. The code includes a 'up()' method with a comment in Spanish: '/* Run the migrations.' and a 'down()' method with a comment: '/* Reverse the migrations.' The terminal at the bottom shows the command 'php artisan make:migration libros' being executed, followed by an information message: 'Migration [C:\xampp\htdocs\biblioteca\database\Migrations\2023_09_05_125334_libros.php] created successfully.'

```
6  
7 return new class extends Migration  
8 {  
9  
10     /**  
11      * Run the migrations.  
12      */  
13     public function up(): void  
14     {  
15         //  
16     }  
17  
18     /**  
19      * Reverse the migrations.  
20      */  
21     public function down(): void  
22     {  
23         //  
24     }  
25 }
```

PS C:\xampp\htdocs\biblioteca> php artisan make:migration libros
INFO Migration [C:\xampp\htdocs\biblioteca\database\Migrations\2023_09_05_125334_libros.php] created successfully.

- Una vez creada la migración agregamos columnas a nuestra tabla
libros

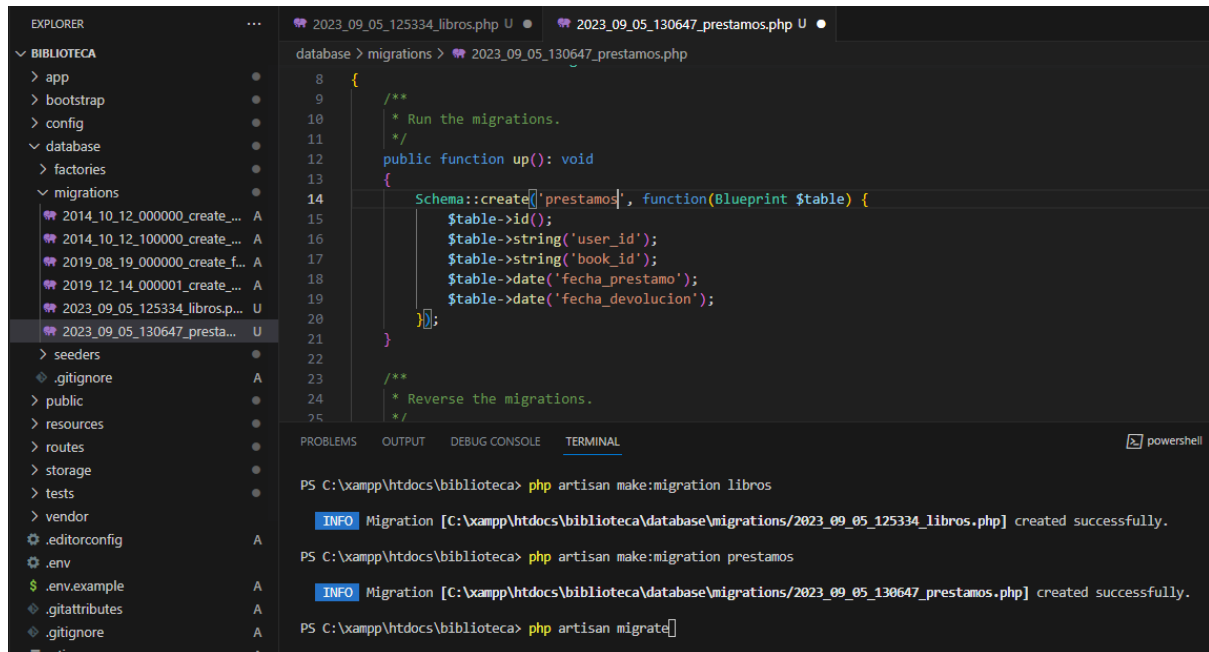


The screenshot shows the Visual Studio Code interface with two migration files open in the editor. The first file, '2023_09_05_125334_libros.php', is the same as in the previous screenshot. The second file, '2023_09_05_130647_prestamos.php', is also open. The main editor shows the content of the first file, which now includes a 'Schema::create()' method with a comment in Spanish: '/* Creamos algunos campos para nuestra tabla */'. The code defines the 'libros' table with columns: 'id', 'titulo', 'autor', 'fecha_publicacion', 'genero', and 'disponible'. The terminal at the bottom shows the command 'php artisan make:migration libros' being executed, followed by an information message: 'Migration [C:\xampp\htdocs\biblioteca\database\Migrations\2023_09_05_125334_libros.php] created successfully.'

```
12 public function up(): void  
13 {  
14     /**  
15      * Creamos algunos campos para nuestra tabla */  
16     Schema::create('libros', function(Blueprint $table) {  
17         $table->id();  
18         $table->string('titulo');  
19         $table->string('autor');  
20         $table->date('fecha_publicacion');  
21         $table->string('genero');  
22         $table->boolean('disponible');  
23     });  
24 }  
25  
26 /**  
27  * Reverse the migrations.  
28  */  
29 public function down(): void  
30 {  
31     //  
32 }  
33 }
```

PS C:\xampp\htdocs\biblioteca> php artisan make:migration libros
INFO Migration [C:\xampp\htdocs\biblioteca\database\Migrations\2023_09_05_125334_libros.php] created successfully.

6. Creamos otra migración llamada **prestamos** y le agregamos columnas a la tabla



The screenshot shows the Visual Studio Code interface. On the left, the Explorer pane displays the project structure, with the 'migrations' folder expanded. The file '2023_09_05_130647_prestamos.php' is selected. The main editor shows the content of this migration file, which defines a schema for a table named 'prestamos' with columns: 'id', 'user_id', 'book_id', 'fecha_prestamo', and 'fecha_devolucion'. The terminal at the bottom shows the execution of two commands: 'php artisan make:migration libros' and 'php artisan make:migration prestamos', both of which were successful. The final command shown is 'php artisan migrate'.

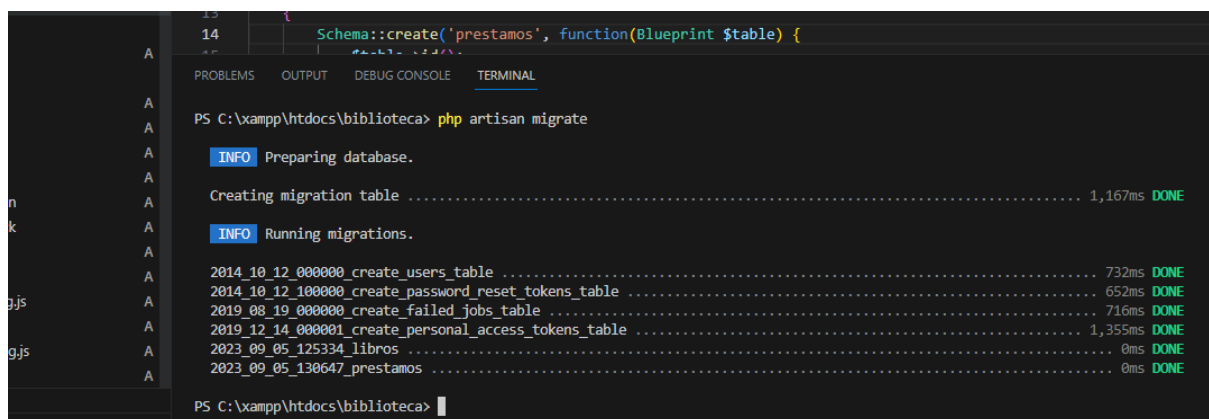
```
8 {
9     /**
10      * Run the migrations.
11      */
12     public function up(): void
13     {
14         Schema::create('prestamos', function(Blueprint $table) {
15             $table->id();
16             $table->string('user_id');
17             $table->string('book_id');
18             $table->date('fecha_prestamo');
19             $table->date('fecha_devolucion');
20         });
21     }
22 }
23
24 /**
25  * Reverse the migrations.
26  */
```

PS C:\xampp\htdocs\biblioteca> php artisan make:migration libros
[INFO] Migration [C:\xampp\htdocs\biblioteca\database\migrations\2023_09_05_125334_libros.php] created successfully.

PS C:\xampp\htdocs\biblioteca> php artisan make:migration prestamos
[INFO] Migration [C:\xampp\htdocs\biblioteca\database\migrations\2023_09_05_130647_prestamos.php] created successfully.

PS C:\xampp\htdocs\biblioteca> php artisan migrate

7. Por último procedemos a aplicar las migraciones a la base de datos con el comando **php artisan migrate**



The screenshot shows the terminal output of the 'php artisan migrate' command. It displays the process of preparing the database, creating a migration table, and running the migrations. The output lists several migrations that were successfully applied, including '2014_10_12_000000_create_users_table', '2014_10_12_100000_create_password_reset_tokens_table', '2019_08_19_000000_create_failed_jobs_table', '2019_12_14_000001_create_personal_access_tokens_table', '2023_09_05_125334_libros', and '2023_09_05_130647_prestamos'. Each migration is followed by its execution time and the status 'DONE'.

```
14 Schema::create('prestamos', function(Blueprint $table) {
15     $table->id();
16     $table->string('user_id');
17     $table->string('book_id');
18     $table->date('fecha_prestamo');
19     $table->date('fecha_devolucion');
20 });
21 }
22 }
23
24 /**
25  * Reverse the migrations.
26  */
```

PS C:\xampp\htdocs\biblioteca> php artisan migrate

[INFO] Preparing database.

Creating migration table 1,167ms DONE

[INFO] Running migrations.

2014_10_12_000000_create_users_table 732ms DONE

2014_10_12_100000_create_password_reset_tokens_table 652ms DONE

2019_08_19_000000_create_failed_jobs_table 716ms DONE

2019_12_14_000001_create_personal_access_tokens_table 1,355ms DONE

2023_09_05_125334_libros 0ms DONE

2023_09_05_130647_prestamos 0ms DONE

PS C:\xampp\htdocs\biblioteca>