**CICLO: DESARROLLO APLICACIONES WEB-DISTANCIA**

**MÓDULO: DESARROLLO DE APLICACIONES WEB EN ENTORNO SERVIDOR**

**TAREA: DWES-07**

**ALUMNO: DAVID MEDINA GARCIA**

**-Realizar una instalación de Laravel (versión 10 u 11), creando un proyecto llamado 'tarea7' mostrando y explicando en un documento \*.pdf las capturas de pantalla en las que se detalle el proceso.**

Vamos a instalar **Laravel 11.**

Para instalar **Laravel** en nuestro equipo debemos tener instalados ciertos programas y recursos antes:

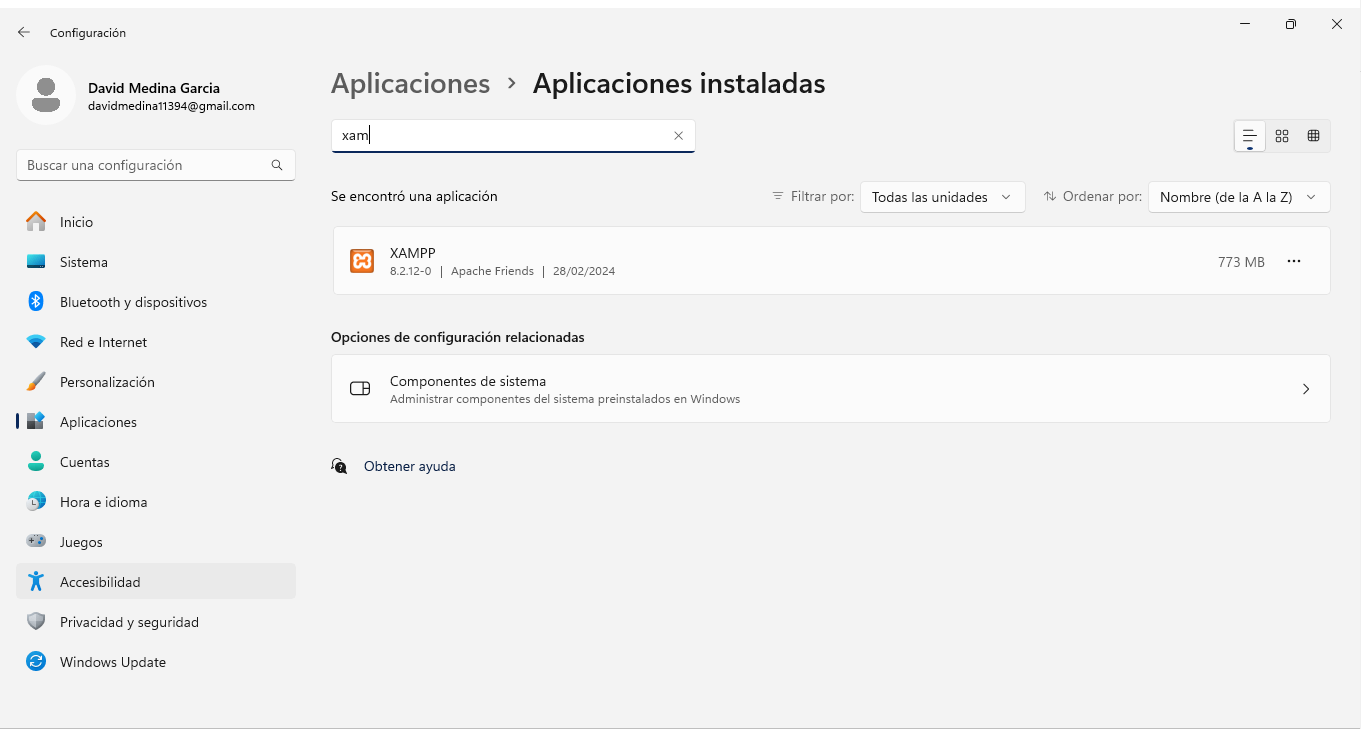
* PHP (Versiones compatibles con **Laravel 11:** 8.2/8.3)
* Composer
* Git (opcional)
* Node.js (opcional)

Aunque **Git y** **Node** son recomendables e incluso necesarios dependiendo las características del proyecto, para una tarea “simple” como esta no será necesario instalarlos.

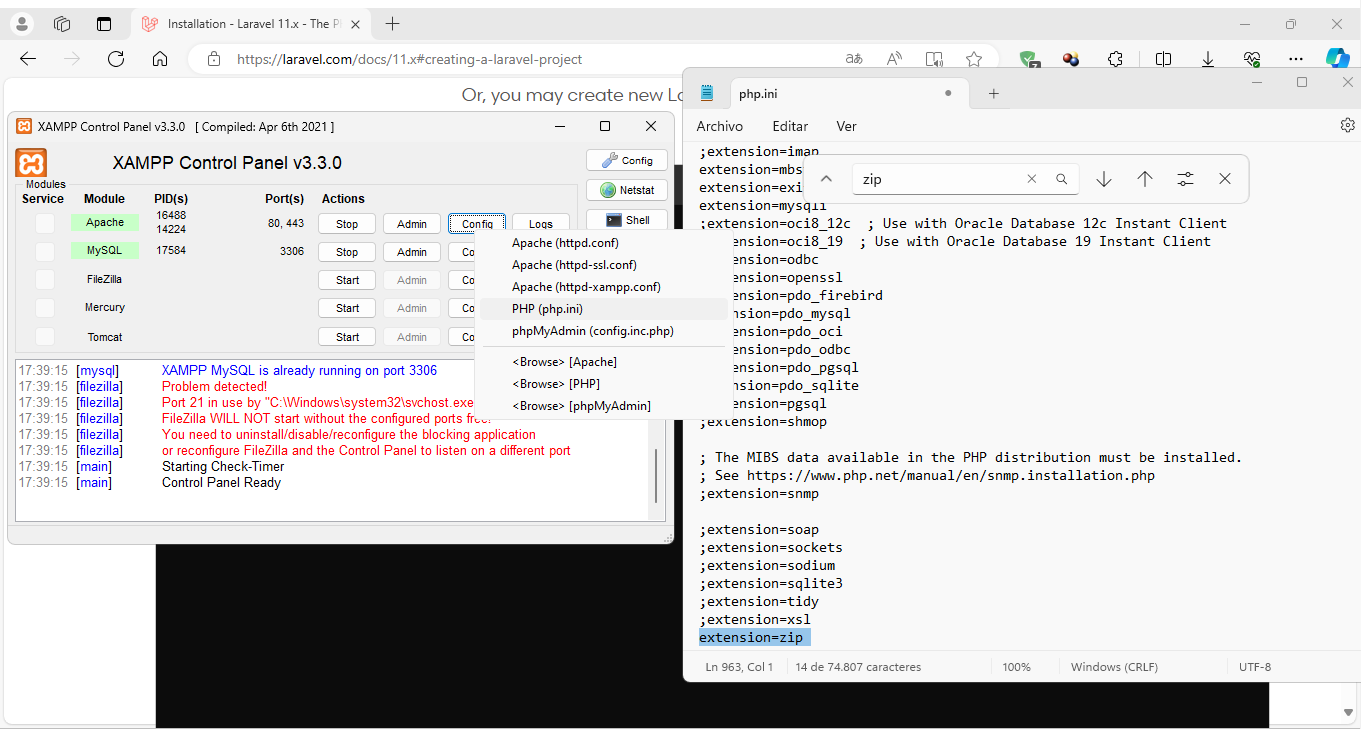
Además como conectaremos el proyecto a una base de datos necesitaremos un **gestor de bases de datos**.

Para la instalación de **PHP** y el **gestor de base de datos** usaremos **XAMPP** que ya tenemos previamente instalado de las practicas anteriores.

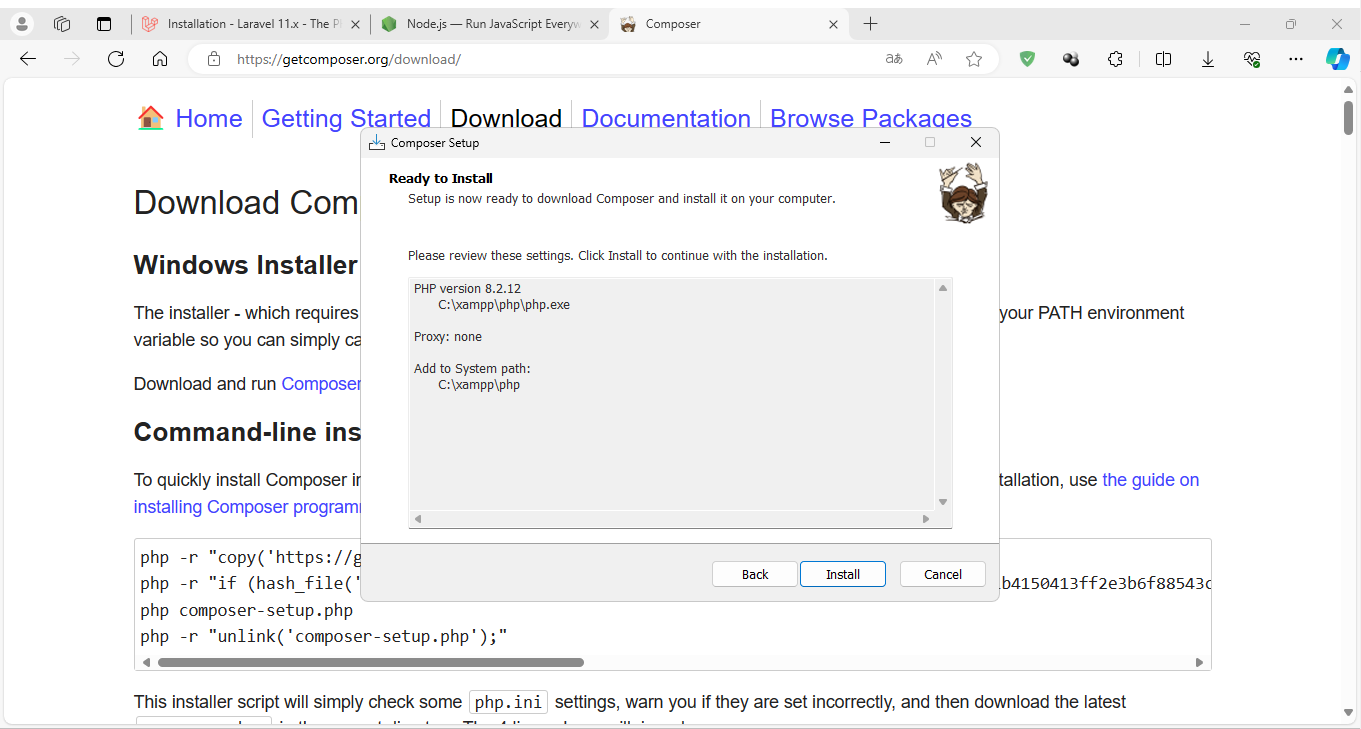
Si no está instalado, desde la página de **xampp** nos descargamos el instalador y lo ejecutamos.

****

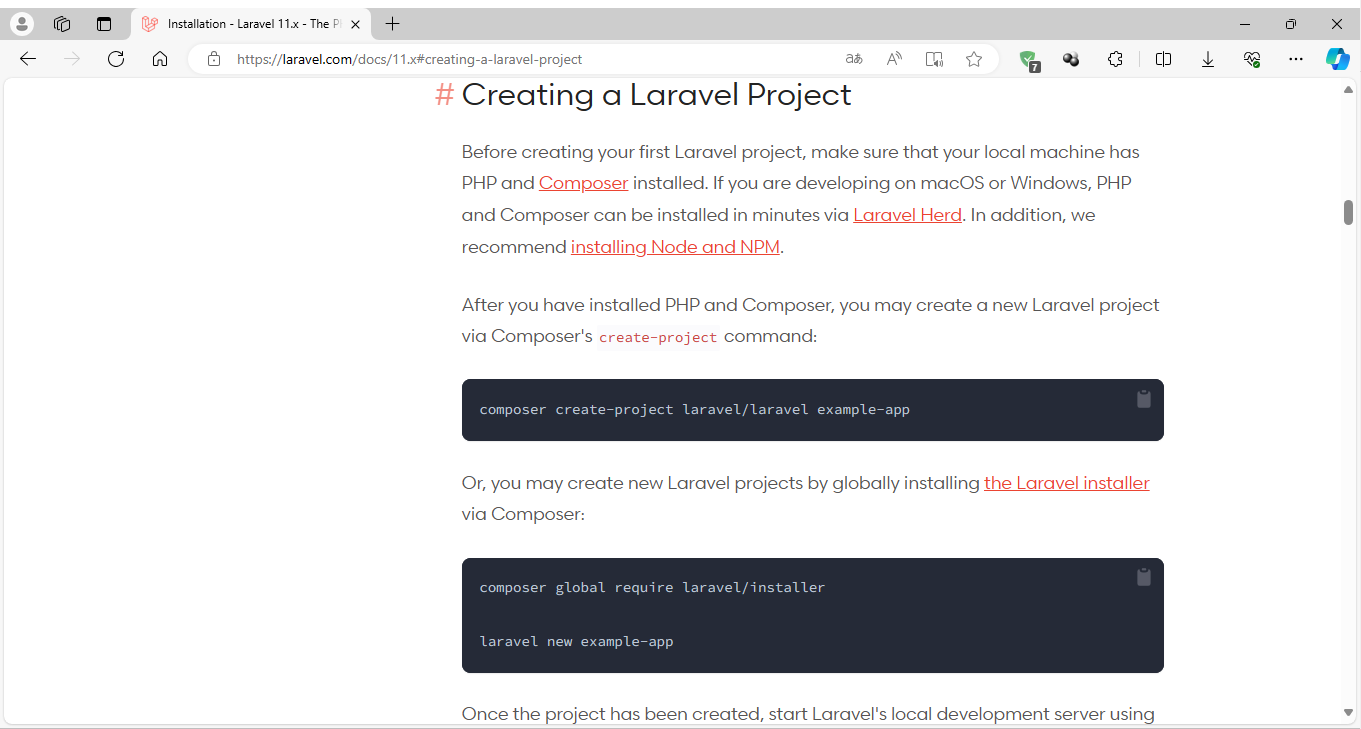
Abrimos **XAMPP** y arrancamos **Apache**. Abrimos el archivo de configuración de **php** (**php.ini**), y descomentamos la **extensión=zip** para que pueda tratar con estos archivos. Reiniciamos **Apache** para aplicar los cambios.

****

A continuación instalamos **Composer.** Lo descargamos desde su página, iniciamos el instalador y seguimos los pasos (darle a next).



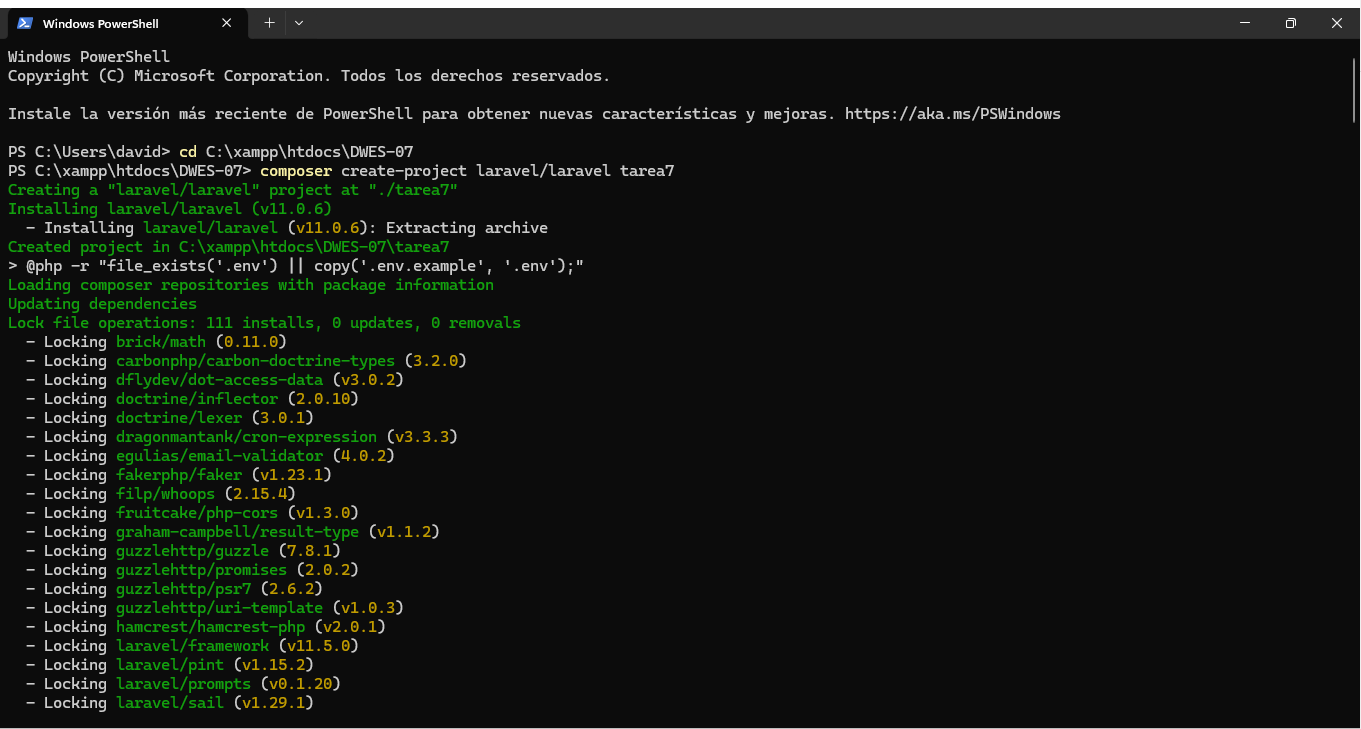
A continuaciónnos vamos a la página de instalación de **Laravel** al apartado #Creating a Laravel Project.



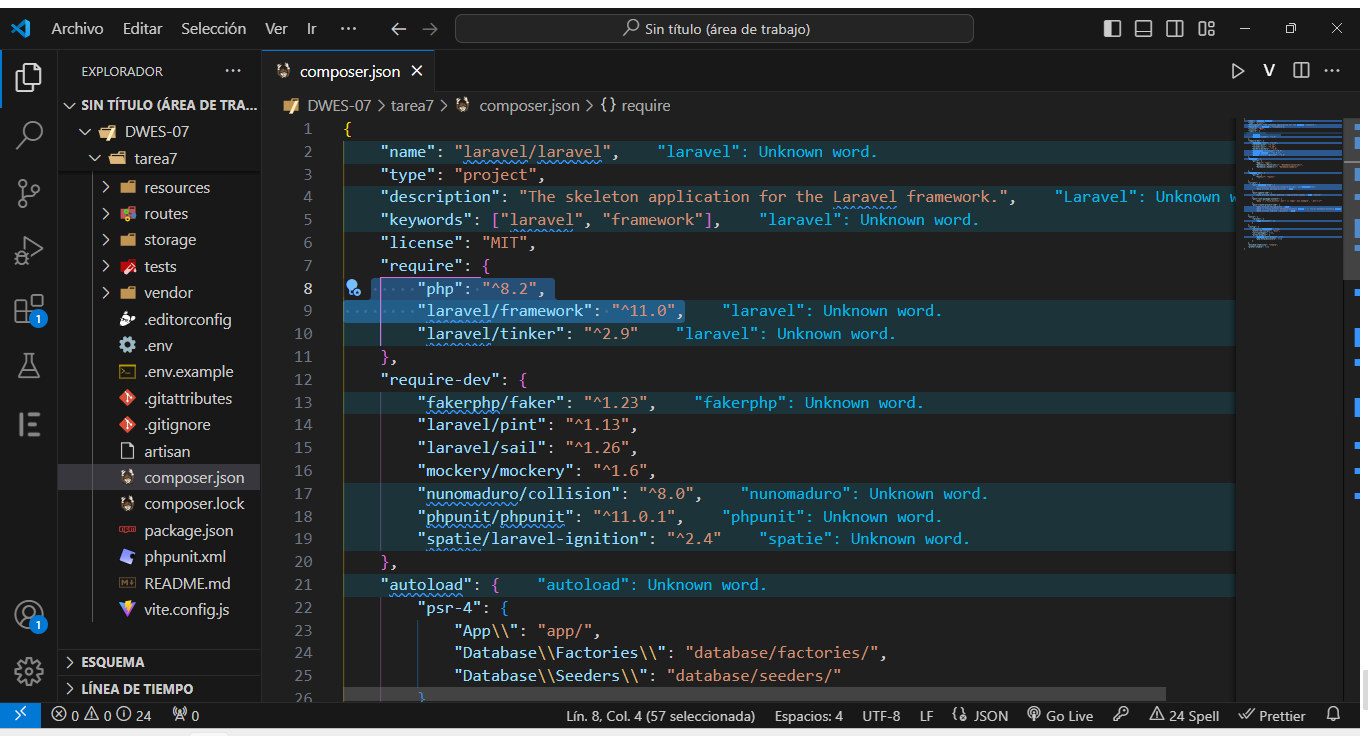
Ahora abrimos un terminal en el equipo y nos situamos en el directorio donde irá nuestro proyecto con **Laravel.**

En mi caso es: **C:\xampp\htdocs\DWES-07**

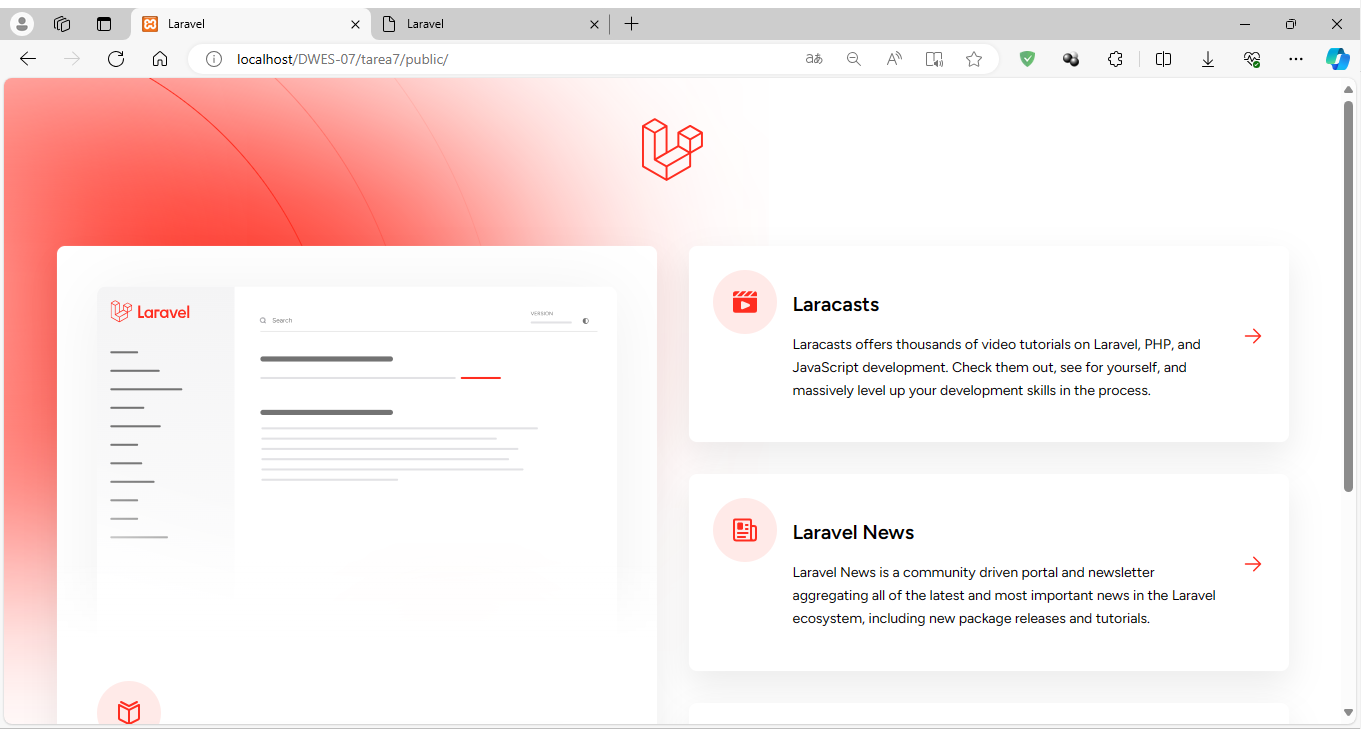
Una vez situamos tenemos que introducir el siguiente comando en el terminal (“tarea7” es el nombre del proyecto): **composer create-project laravel/laravel tarea7**



Si abrimos el archivo **composer.jason** en un IDE, podemos ver las versiones de **PHP** y **Laravel** que tenemos instalados.

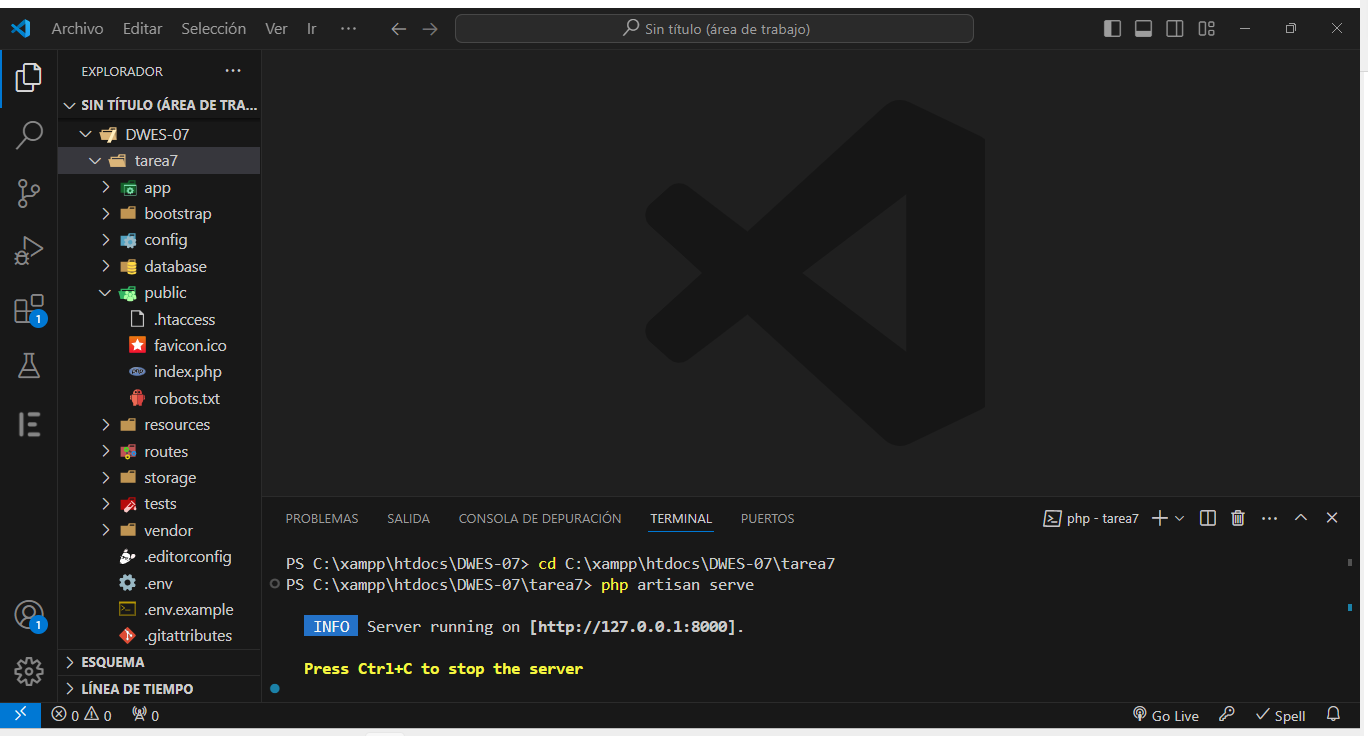


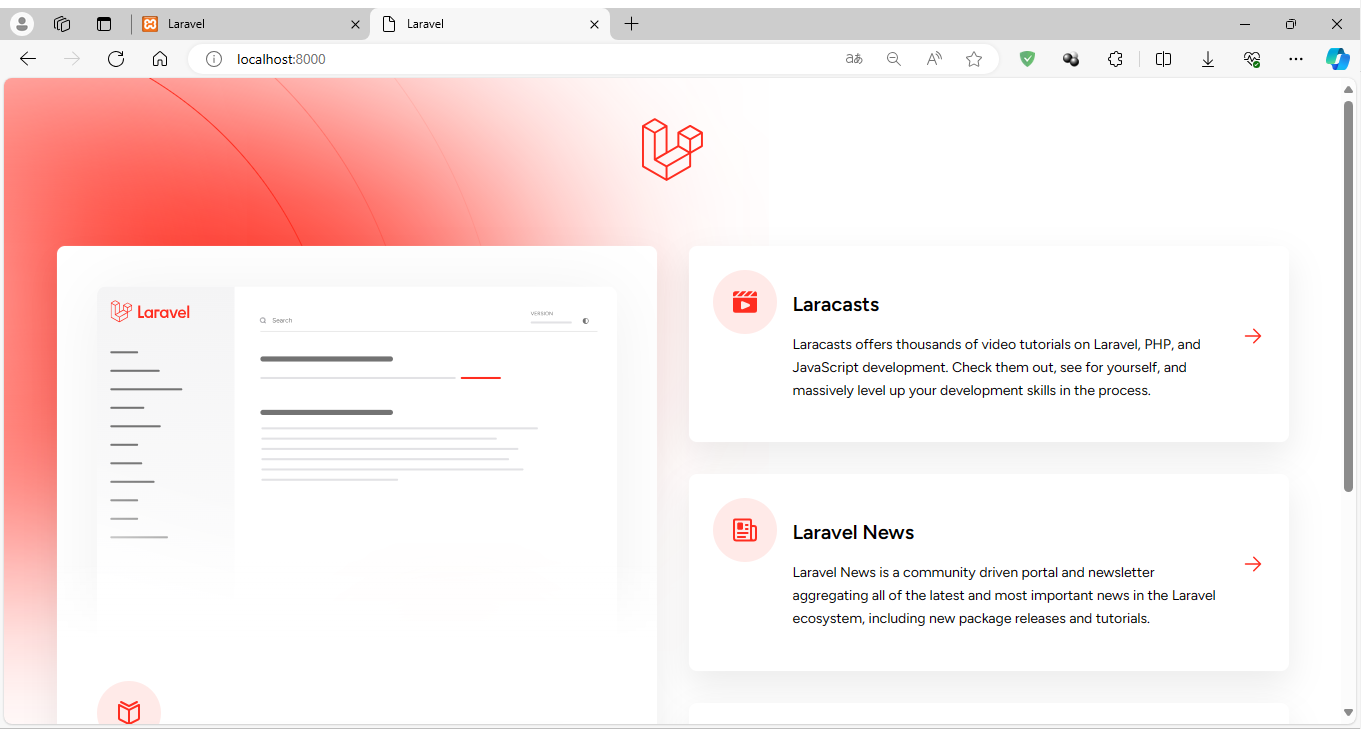
Ahora si abrimos un navegador y vamos a la dirección **http://localhost/DWES-07/tarea7/public/**  o a la dirección **http://localhost/DWES-07/tarea7/public/index.php** abriremos la página por defecto de **Laravel**.



Otra manera de abrir esa página es situarnos en la carpeta del proyecto (tarea7) e introducir el siguiente comando: **php artisan serve.**

Con esto activamos un servidor local y si en un navegador nos vamos a **localhost:8000** también abriremos la página por defecto.

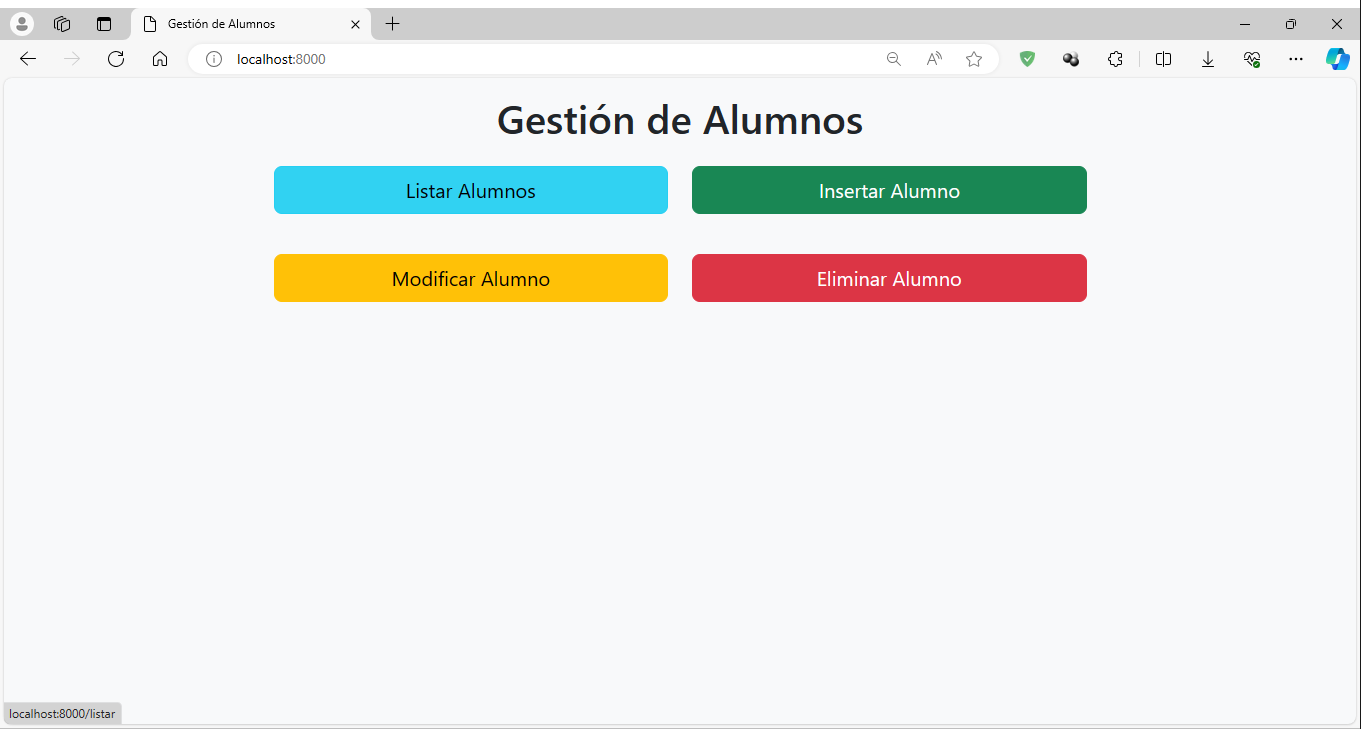


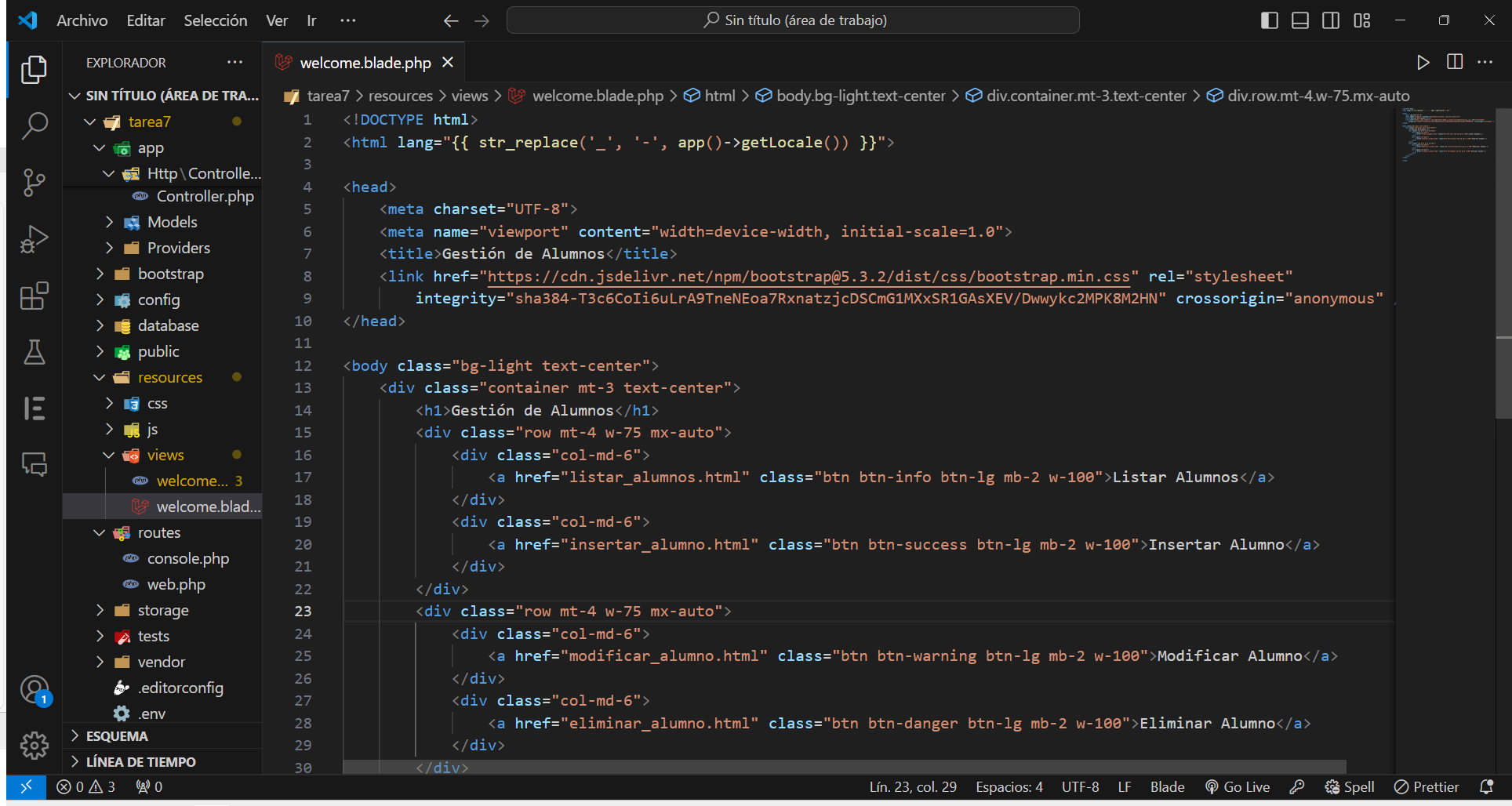


**-Modificar la vista wellcome que viene por defecto con un mesaje personalizado.**

Para modificar la vista **welcome** se debe modificar el archivo **views/welcome.blade.php**

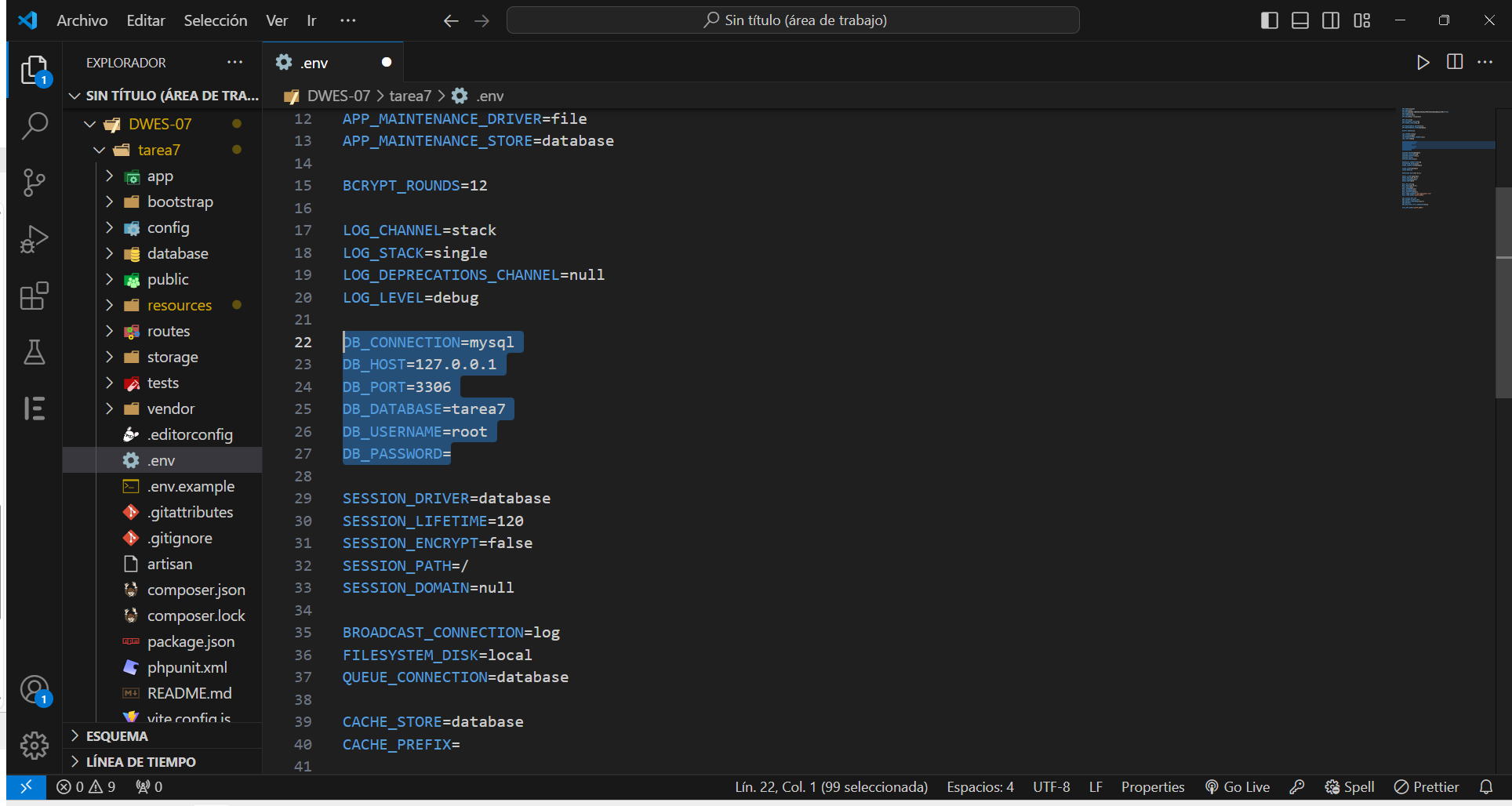
La modificación que he hecho ha sido sustituir al código que venía por una página a través de la cual podemos hacer diversas acciones sobre una base de datos con la que vamos a trabajar en esta unidad.



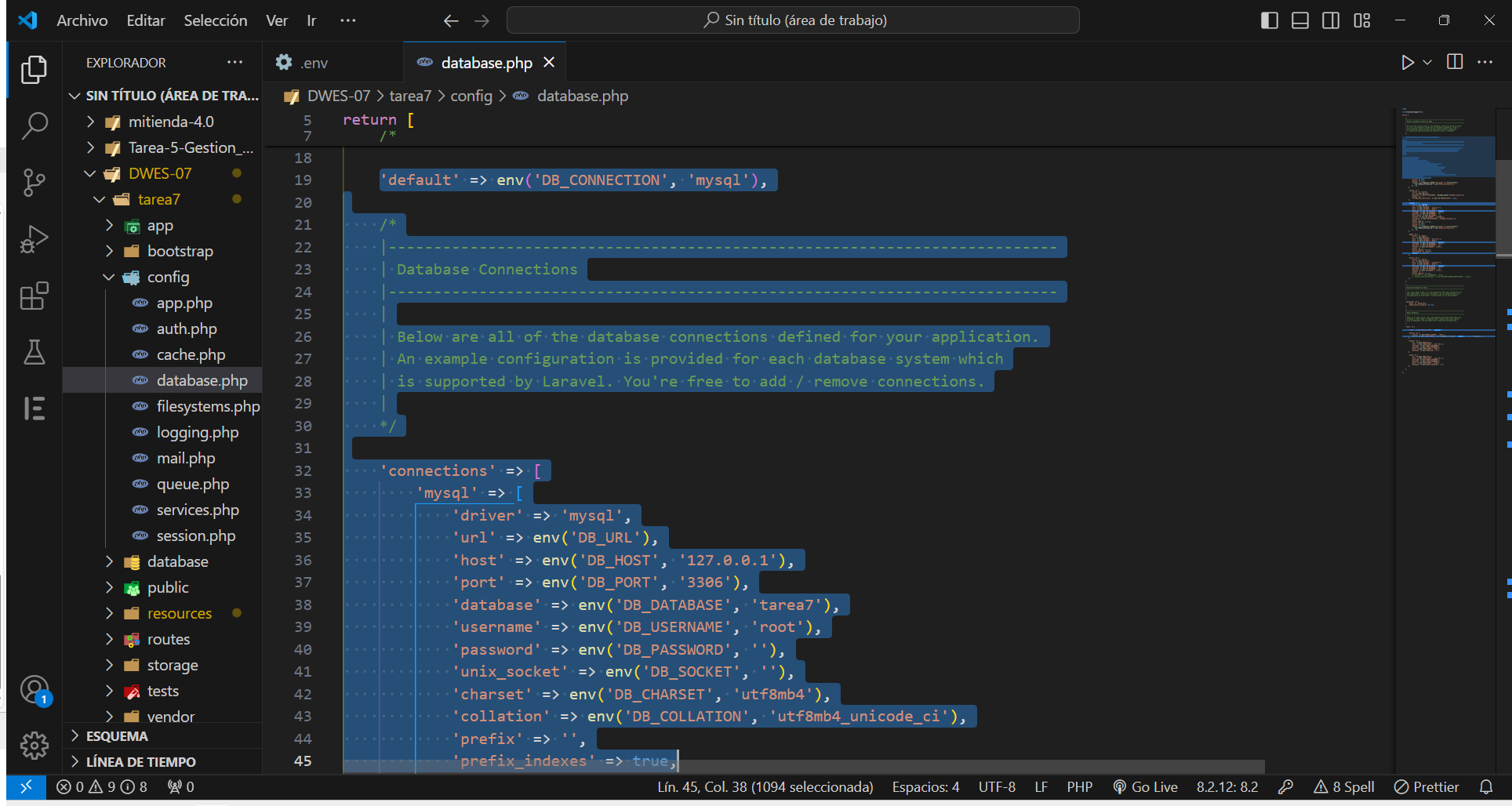


**-Conectar el proyecto con la base de datos de la tarea anterior e indicar como se ha realizado**

Para conectar el proyecto con nuestra base de datos tenemos que abrir el archivo **.env** y modificar las siguientes líneas con los datos que necesitemos.



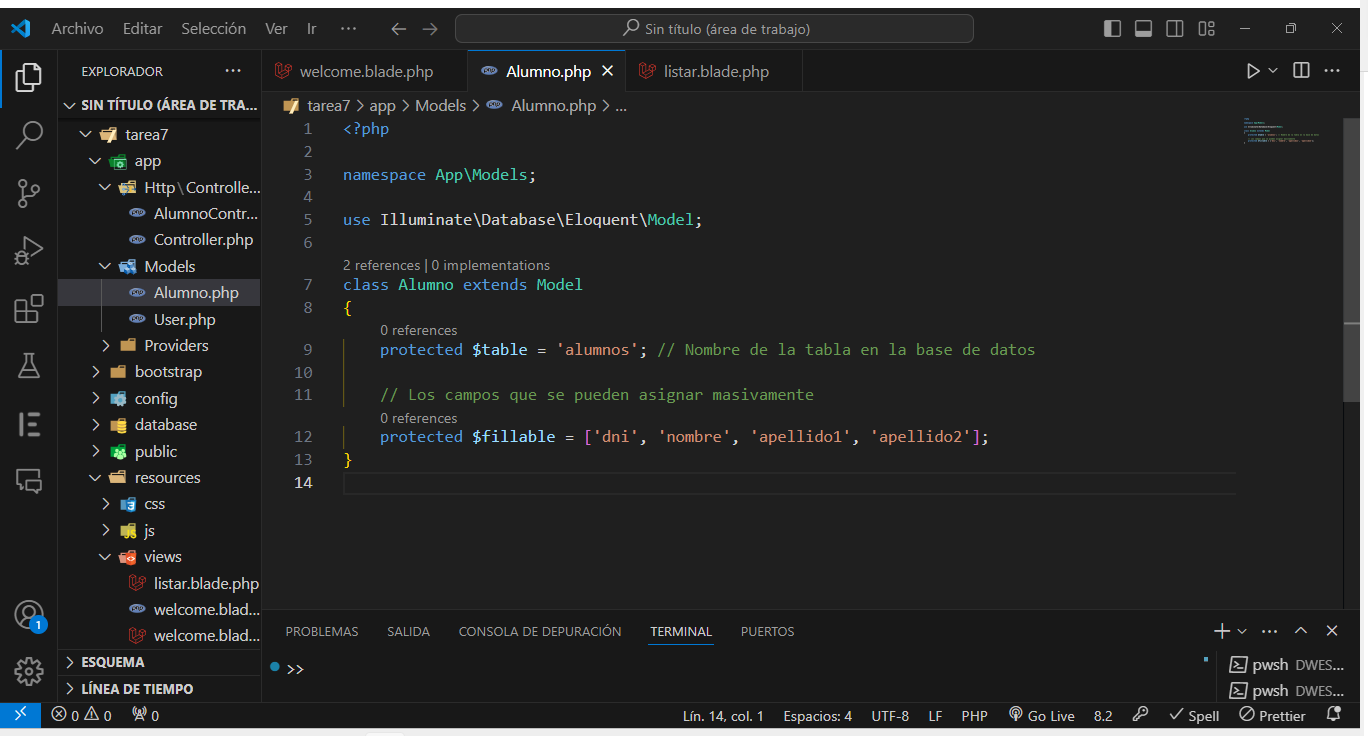
También hay que modificar los datos del archivo **config/database.php**



En mi caso, indico que use **mysql**  y la base de datos **tarea7.** Como vamos a trabajar con el usuario root, no tocamos nada más.

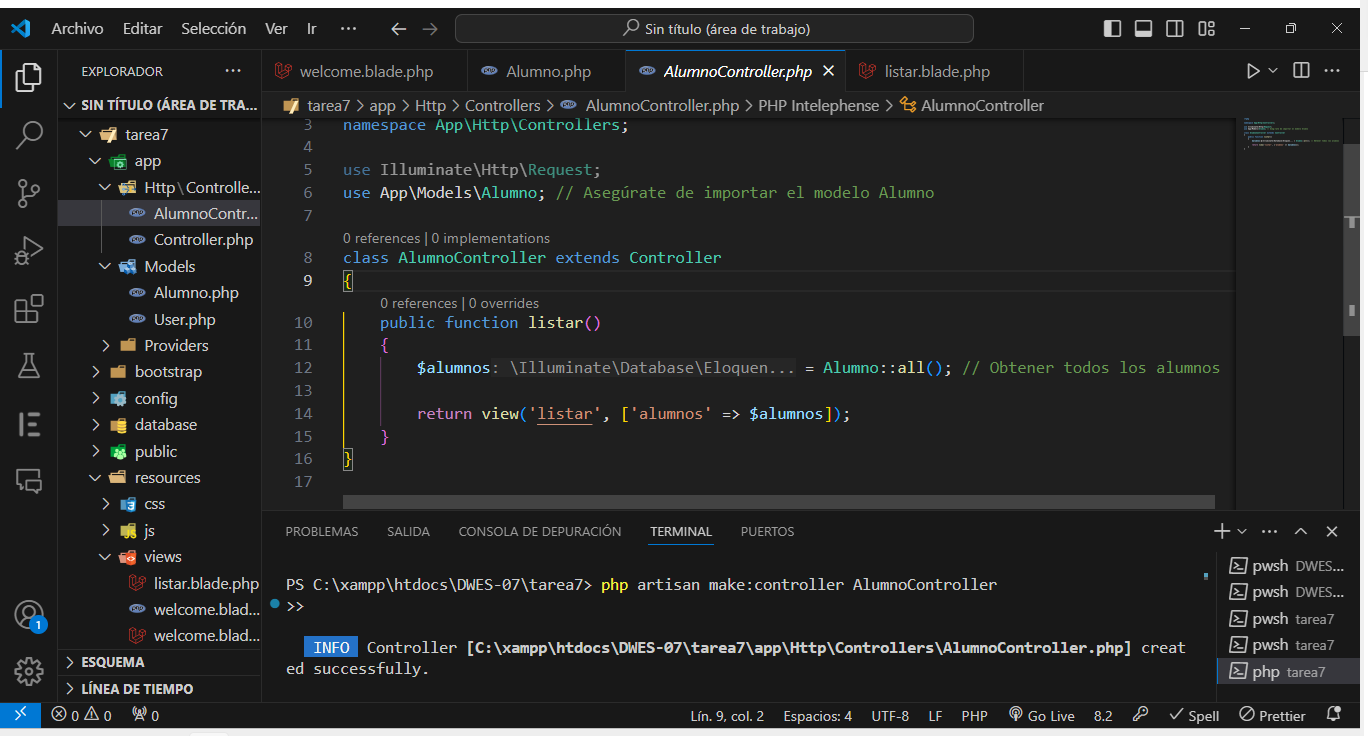
**-Crear una vista llamada /listar que muestre en una tabla html todos los alumos de la tabla alumnos.sql.**

Primero tenemos que crear el modelo alumno.



Después creamos el controlador del alumno. Ponemos en la terminal lo siguiente (AlumnoController es el nombre): **php artisan make:controller AlumnoController**

La función listar usa el método **All**() para obtener todos los alumnos y después carga la vista “listar”.

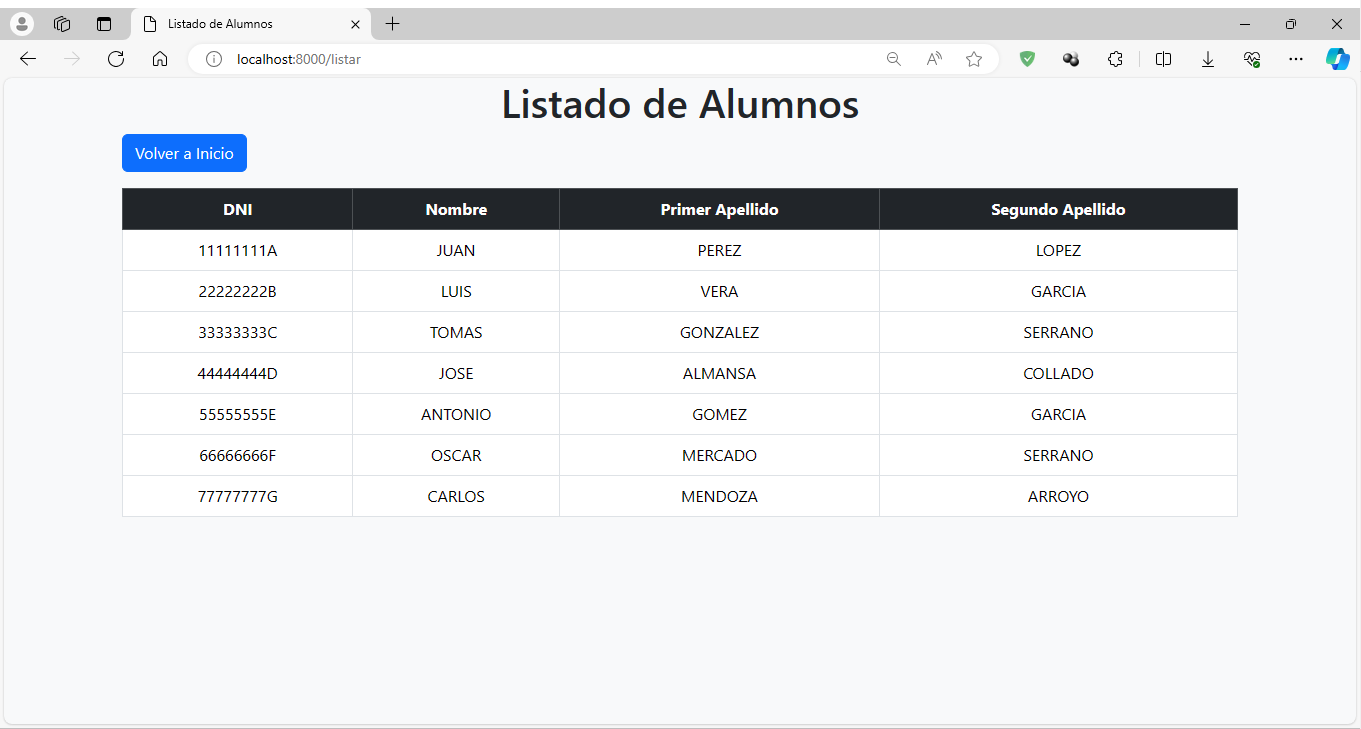


A continuación en el archivo **routes/web.php** añadimos una ruta:

**Route::get('/listar', 'App\Http\Controllers\AlumnoController@listar';**

Cuando un usuario accede a esta URL (listar), se llama al método listar del controlador **AlumnoController**.

Finalmente creamos nuestra vista “listar” (**listar.blade.php**).



**-Crear una vista llamada /alta que permita insertar alumnos en la tabla alumnos.sql.**

Para este apartado debemos añadir dos funciones a **AlumnoController**.

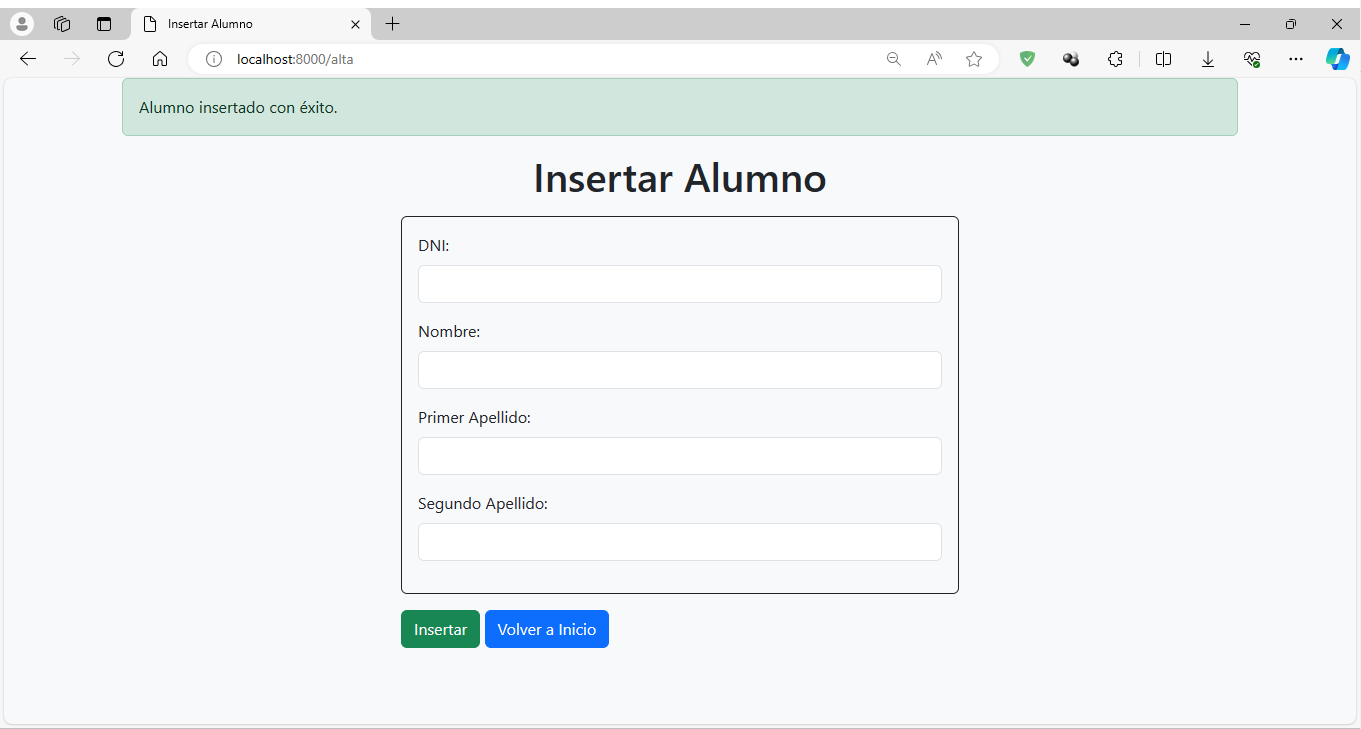
Una llamado **showAlta** () para mostrar la vista del formulario de inserción y otra llamado **store**() que procesa la solicitud para insertar un nuevo alumno y muestra un mensaje de éxito en caso de hacerlo.

Luego tenemos que añadir dos rutas nuevas a nuestro archivo **web.php.** Una llama al método **showAlta** y otra al método **store.**

**Route::get('/alta', 'App\Http\Controllers\AlumnoController@showAlta);**

**Route::post('/alta', 'App\Http\Controllers\AlumnoController@store')->name(alta.alumno);**

Finalmente creamos nuestra vista “alta” (**alta.blade.php**).



**-Crear una vista llamada /actualizar que permita actualizar datos de un alumno en la tabla alumnos.sql.**

Para este apartado debemos añadir tres funciones a **AlumnoController**.

Una llamado **showUpdate**() para mostrar la vista de actualización, otra llamado **buscarAlumnoActualizar**() que buscara al alumno con el DNI introducido y mostrara los datos en un formulario editable, y otra llamada **actualizar()** que recoge los datos del formulario y actualiza la base de datos.

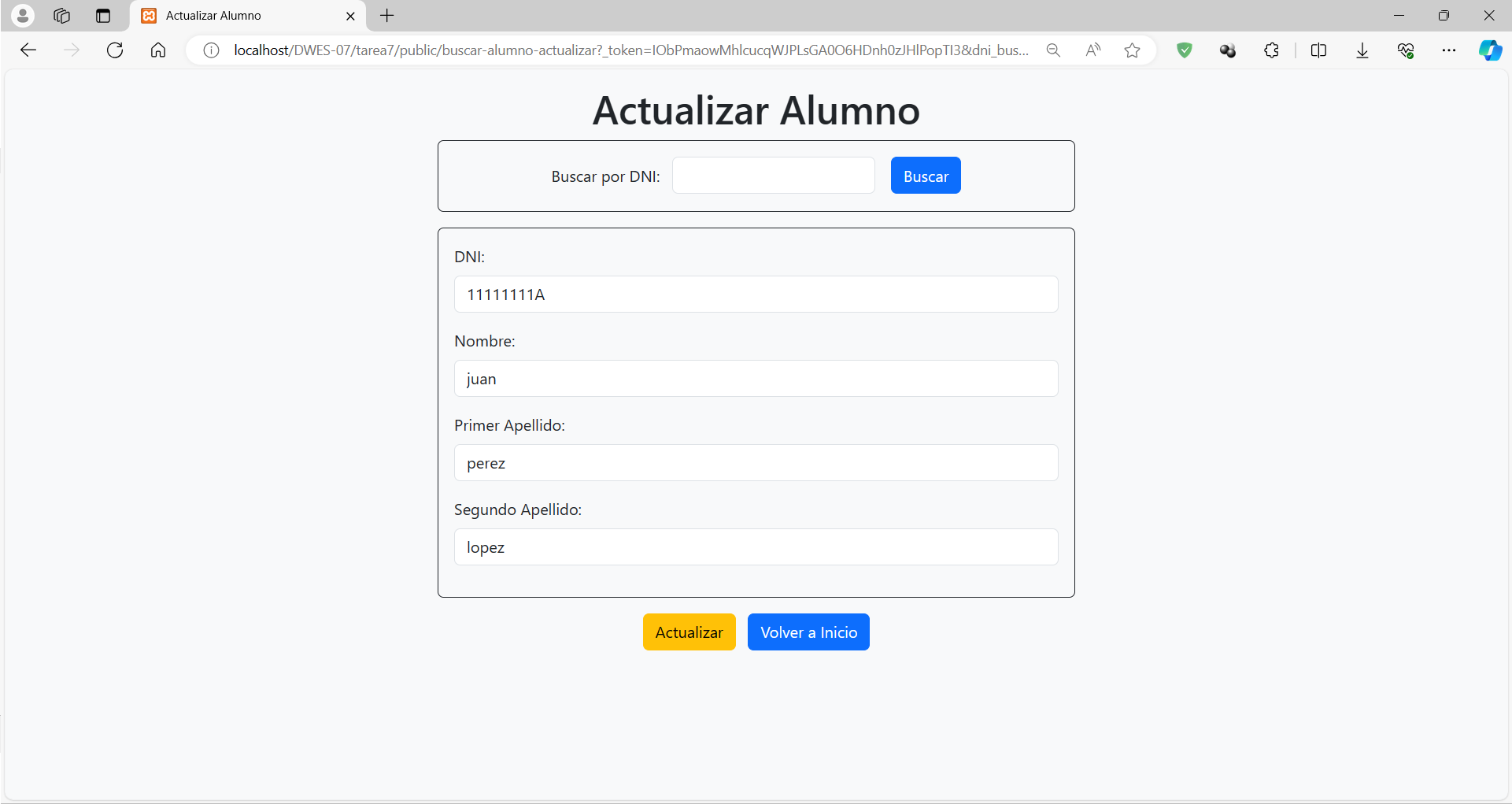
Luego tenemos que añadir dos rutas nuevas a nuestro archivo **web.php.** Una llama al método **showUpdate,** otra al método **buscarAlumnoActualizar** y otra al método **actualizar.**

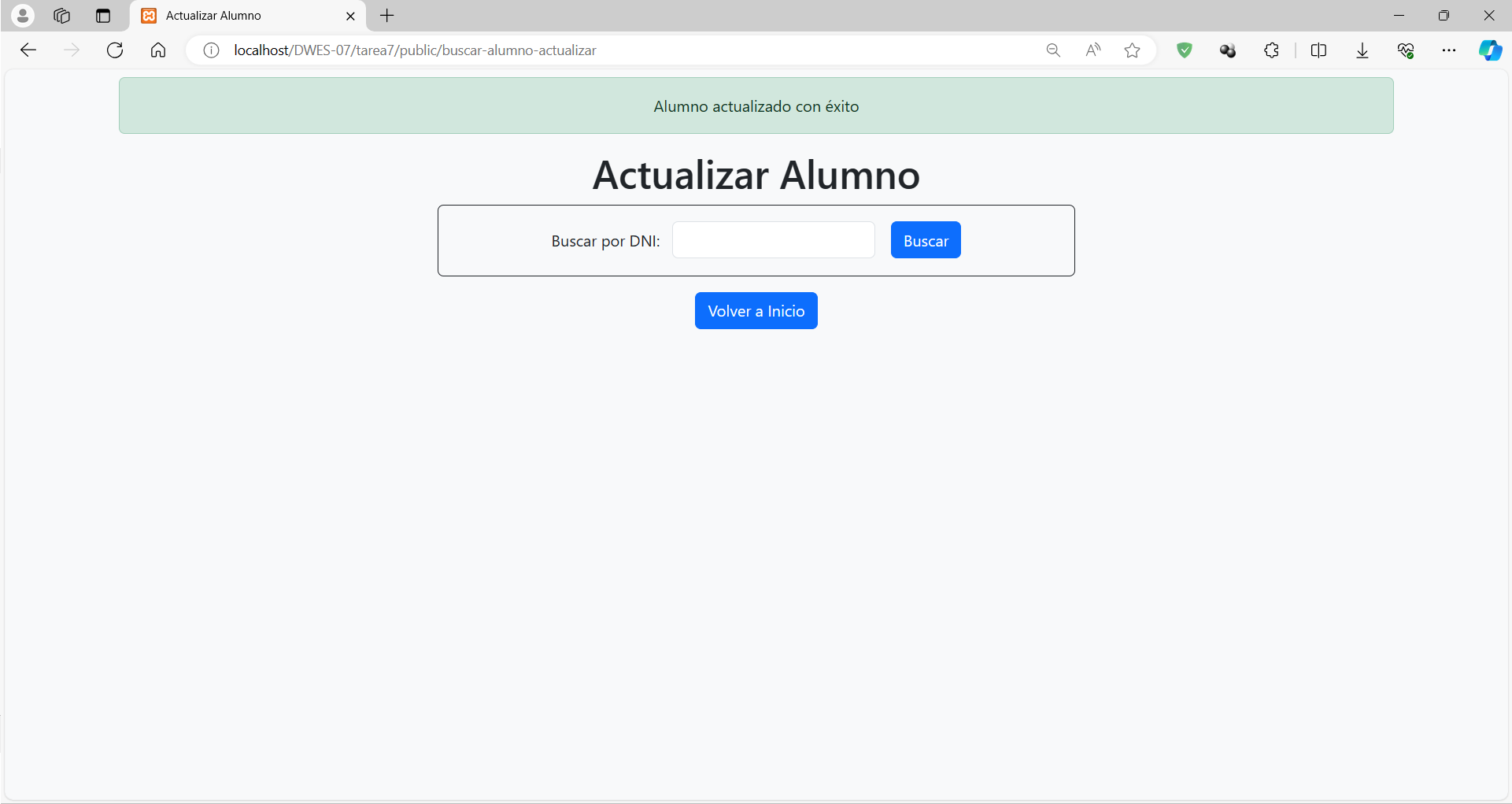
**Route::get('/actualizar, 'App\Http\Controllers\AlumnoController@showUpdate);**

**Route::post('/buscar-alumno-actualizar, 'App\Http\Controllers\AlumnoController@ buscarAlumnoActualizar)->name('buscar.alumno.actualizar');**

**Route::post('/'actualizar-alumno, 'App\Http\Controllers\AlumnoController@actualizar)->name('actualizar.alumno');**

Finalmente creamos nuestra vista “actualizar” (**actualizar.blade.php**).

****

****

**-He añadido una vista llamada /eliminar que permita eliminar datos de un alumno en la tabla alumnos.sql.**

Para este apartado debemos añadir tres funciones a **AlumnoController**.

Una llamado **showDelete**() para mostrar la vista de actualización, otra llamado **buscarAlumnoEliminar**() que buscara al alumno con el dni introducido y mostrara una tabla con sus datos, y otra llamada **eliminar()** que elimina al alumno de la base de datos.

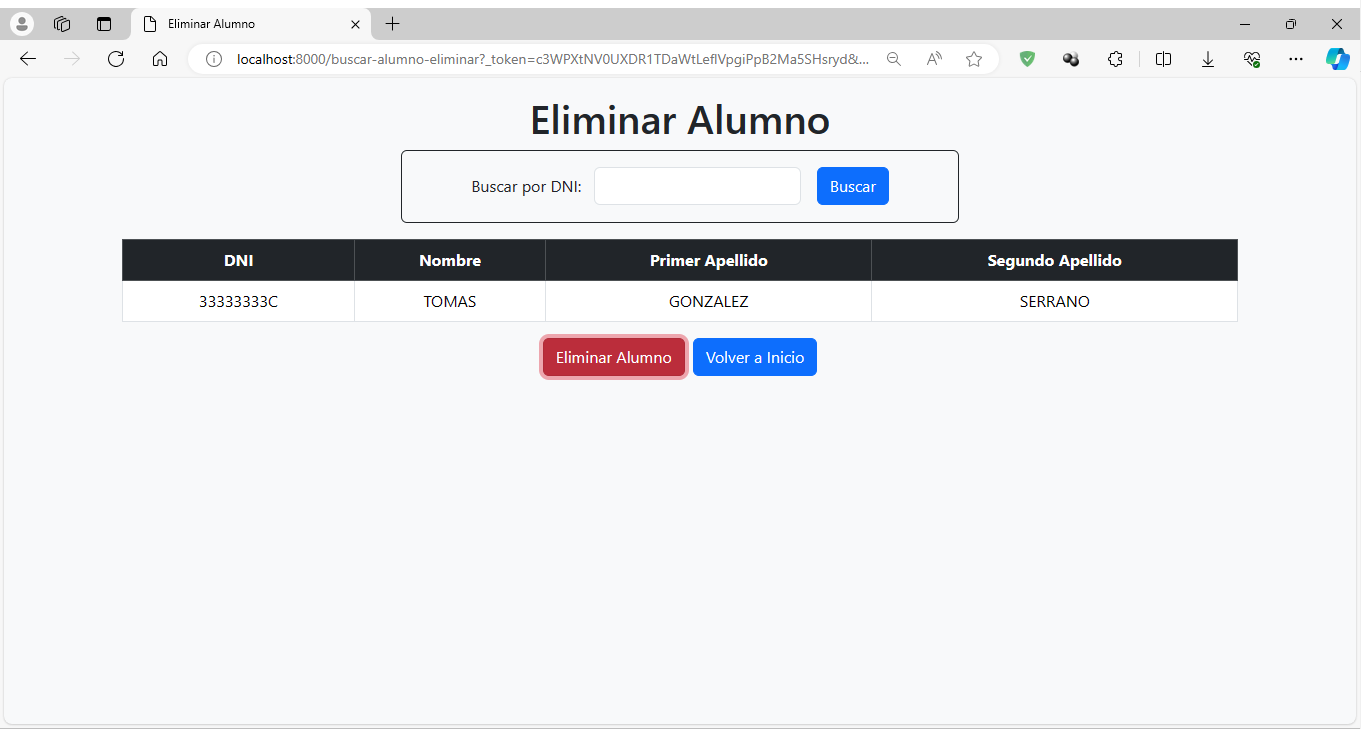
Luego tenemos que añadir dos rutas nuevas a nuestro archivo **web.php.** Una llama al método **showDelete,** otra al método **buscarAlumnoEliminar** y otra al método **eliminar.**

**Route::get('/eliminar, 'App\Http\Controllers\AlumnoController@showDelete);**

**Route::post('/buscar-alumno-eliminar, 'App\Http\Controllers\AlumnoController@buscarAlumnoEliminar)->name('buscar.alumno.eliminar');**

**Route::post('/eliminar -alumno, 'App\Http\Controllers\AlumnoController@eliminar)->name('eliminar.alumno');**

Finalmente creamos nuestra vista “eliminar” (**eliminar.blade.php**).

****