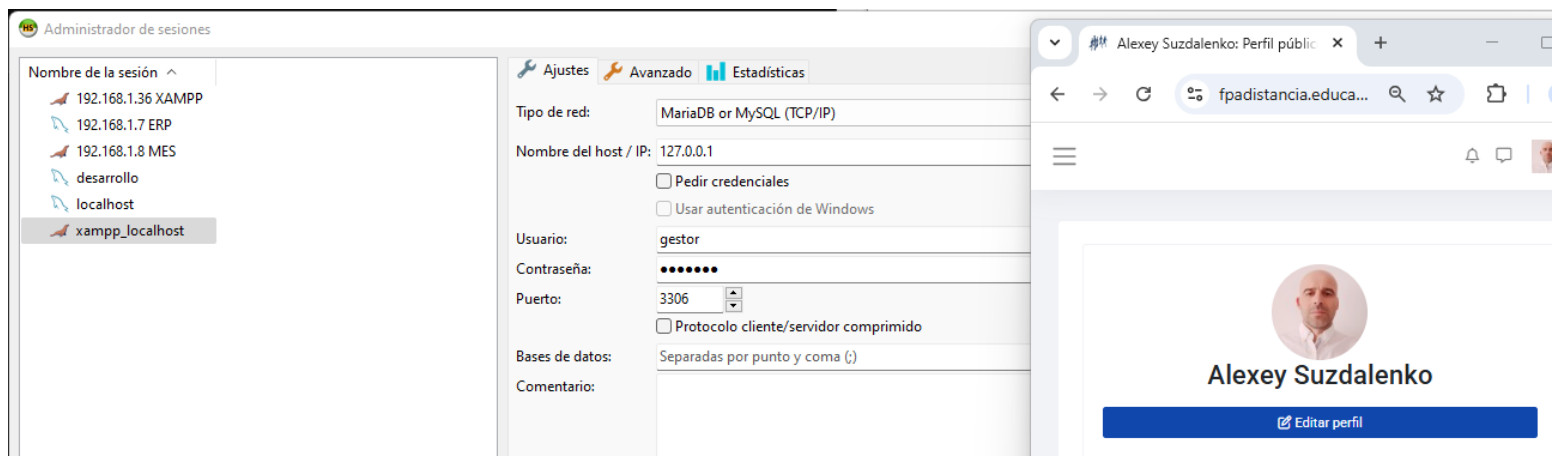
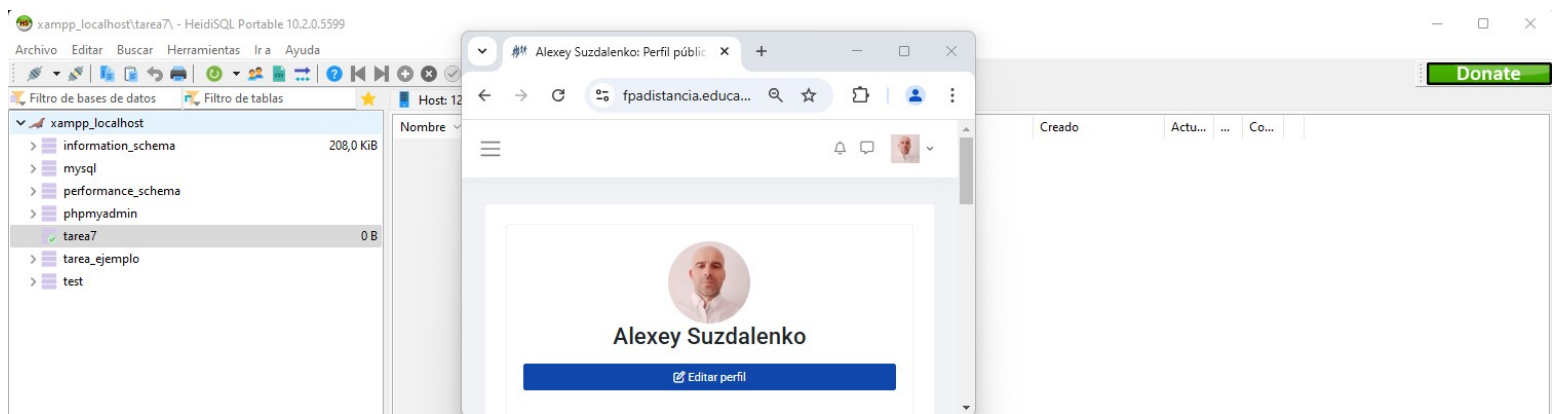


## Suzdalenko Alexey

Lo primero vamos a confirmar que tengo una base de datos creada “tarea7” con el usuario “gestor” y contraseña “secreto”:



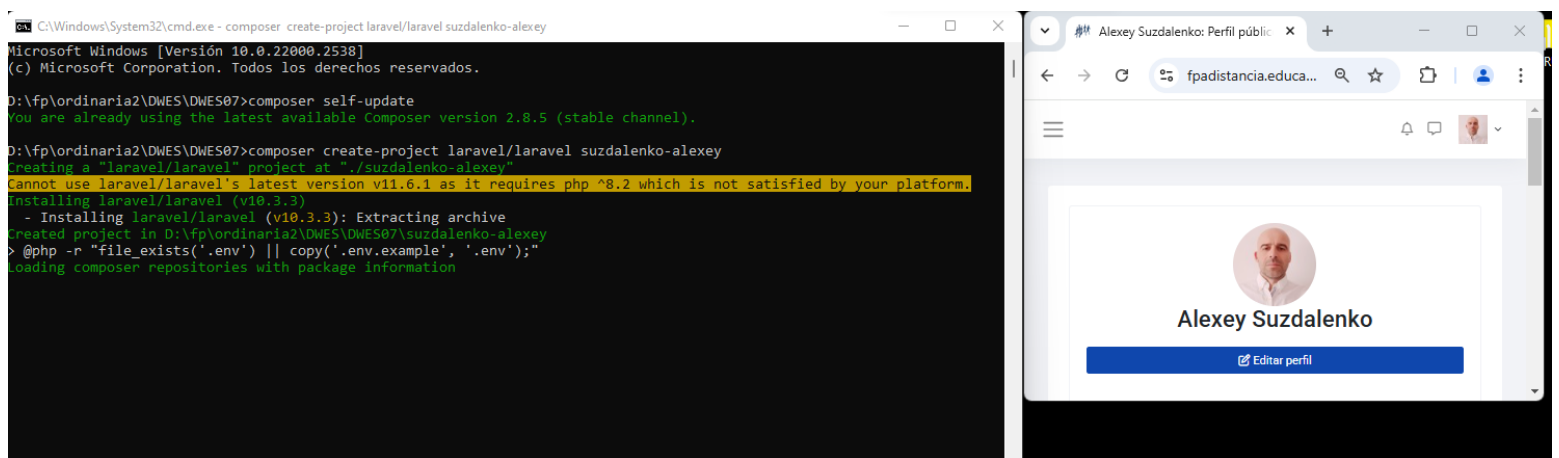
la base datos dentro de la conexión seleccionada:



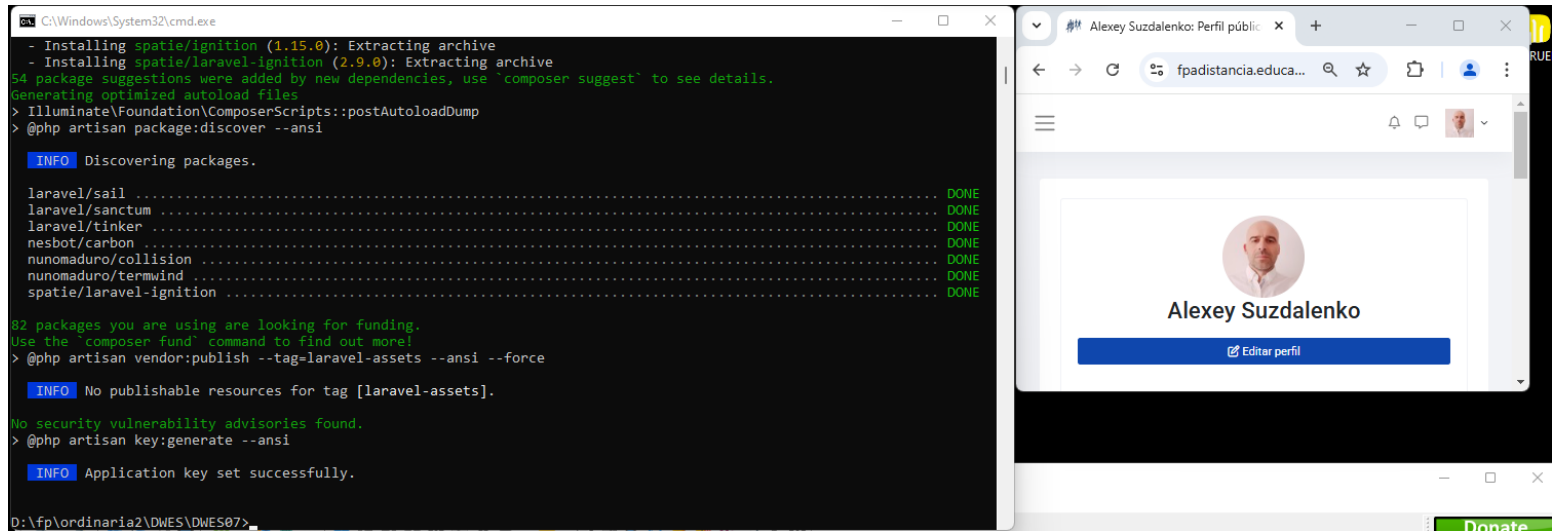
### 1. Instalar Laravel con composer, comandos:

composer self-update

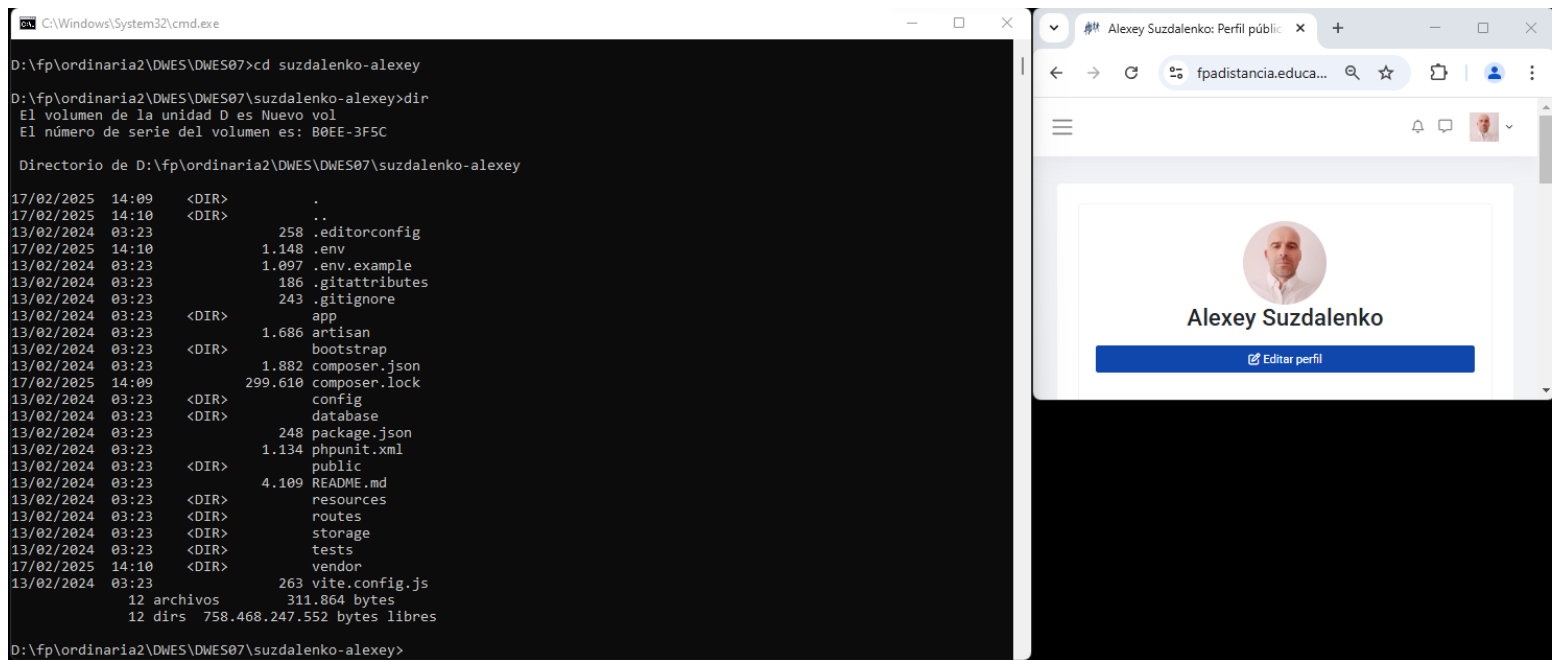
composer create-project laravel/laravel suzdalenko-alexey



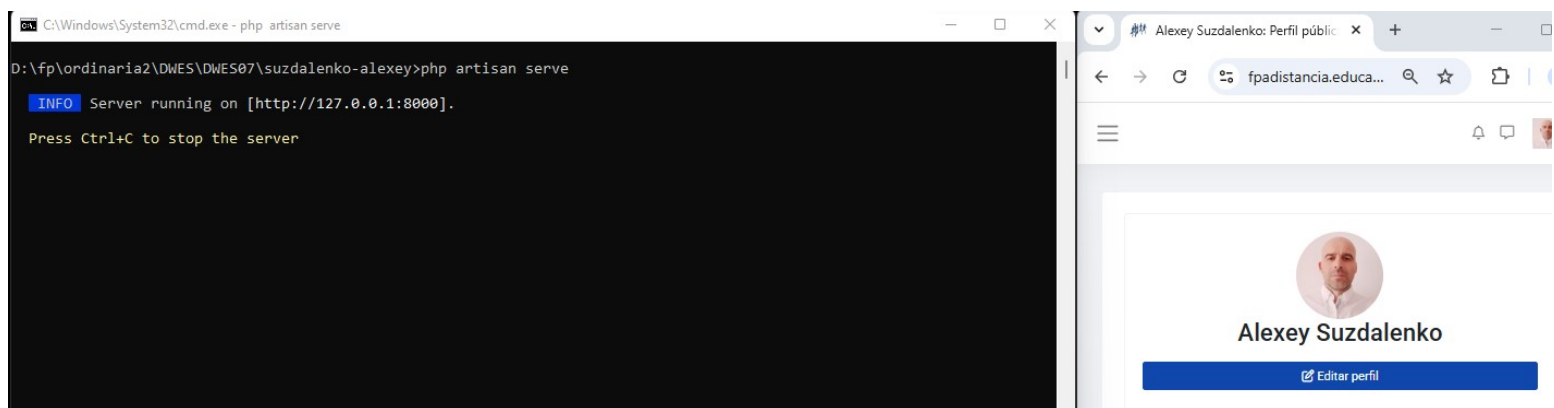
procede la instalación



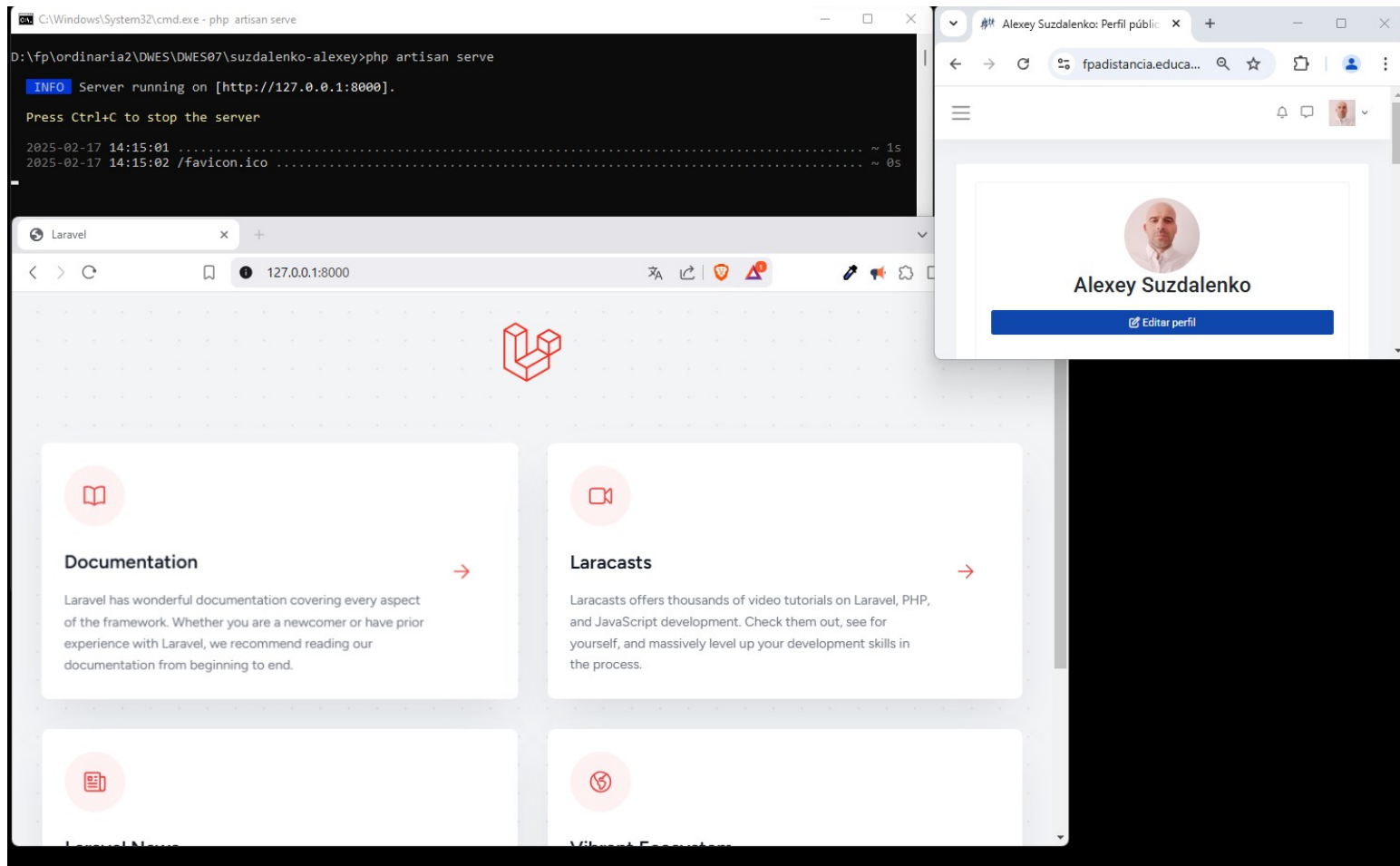
• ahora vamos a la carpeta creada suzdalenko-alexey y vemos su contenido:



• laravel instalado, intentamos iniciar la aplicación web de laravel con el comando:  
php artisan serve

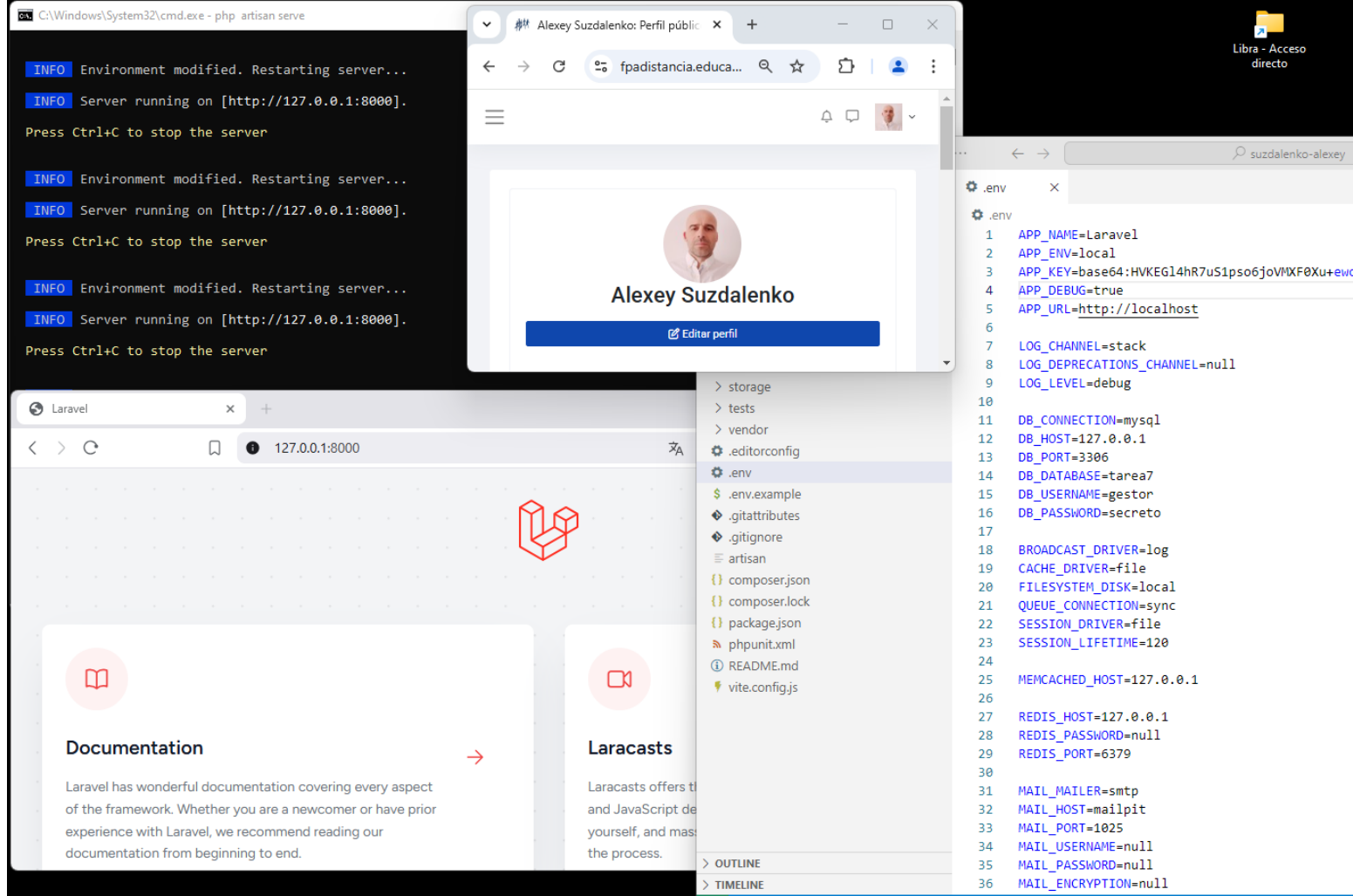


vemos que esta corriendo en la url <http://127.0.0.1:8000>, lo abrimos con el navegador y tenemos la pagina web de la bienvenida de laravel:



ahora editamos el archivo .env de laravel para indicar los credenciales de la base datos que tenemos:

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=tarea7
DB_USERNAME=gestor
DB_PASSWORD=secreto
```



tenemos el Laravel corriendo.

## 2. Visualizar los datos de la tabla productos.

Lo primero vamos a crear tabla productos con los datos necesarios en la base de datos, y esta vez en vez de crearla a mano, vamos hacerlos a través de los clases de laravel, entonces, ahora creamos el modelo (la clase) Producto, comando:

```
php artisan make:model Producto
```

este comando nos a creado la clase Producto y modelo para la migracion, añado las capturas de ambos

File Edit Selection View Go ...

suzdalenko-alexey

EXPLORER

SUZDALENKO-ALEXEY

app

Console

Exceptions

Http

Models

Producto.php

User.php

Providers

bootstrap

config

database

public

resources

routes

storage

tests

vendor

.editorconfig

.env

.env.example

.gitattributes

.gitignore

OUTLINE

TIMELINE

Producto.php

app > Models > Producto.php > ...

```
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7
8 class Producto extends Model
9 {
10     use HasFactory;
11 }
12
```

Alxey Suzdalenko: Perfil público

fpadistancia.educa...

Alexey Suzdalenko

Editar perfil

PROBLEMS OUTPUT TERMINAL PORTS COMMENTS DEBUG CONSOLE

PS D:\fp\ordinaria2\DWES\DWES07\suzdalenko-alexey> php artisan make:model Producto -m

INFO Model [D:\fp\ordinaria2\DWES\DWES07\suzdalenko-alexey\app\Models\Producto.php] created successfully.

INFO Migration [D:\fp\ordinaria2\DWES\DWES07\suzdalenko-alexey\database\migrations\2025\_02\_17\_144909\_create\_productos\_table.php] created successfully.

PS D:\fp\ordinaria2\DWES\DWES07\suzdalenko-alexey>

179 B Ln 1, Col 1 Spaces: 4 UTF-8 LF PHP Completions limit reached

The screenshot shows the Visual Studio Code editor with a file explorer on the left displaying the project structure. The main editor window shows a migration file named `2025_02_17_144909_create_productos_table.php`. The code defines a migration class that creates a `productos` table with an `id` primary key and timestamps. The terminal at the bottom shows the command `php artisan make:model Producto -m` being executed, with output indicating that the model and migration were created successfully.

```
1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 return new class extends Migration
8 {
9     /**
10      * Run the migrations.
11      */
12     public function up(): void
13     {
14         Schema::create('productos', function (Blueprint $table) {
15             $table->id();
16             $table->timestamps();
17         });
18     }
19
20     /**
21      * Reverse the migrations.
22      */
23     public function down(): void
24     {
25         Schema::dropIfExists('productos');
26     }
27 };
28
```

PS D:\fp\ordinaria2\DWES\DWES07\suzdalenko-alexy> php artisan make:model Producto -m

INFO Model [D:\fp\ordinaria2\DWES\DWES07\suzdalenko-alexy\app\Models\Producto.php] created successfully.

INFO Migration [D:\fp\ordinaria2\DWES\DWES07\suzdalenko-alexy\database\migrations\2025\_02\_17\_144909\_create\_productos\_table.php] created successfully.

PS D:\fp\ordinaria2\DWES\DWES07\suzdalenko-alexy>

ahora en la clase producto especificamos los campos que vamos a tener en la base datos, con el siguiente código:

The screenshot shows the Visual Studio Code editor with the file explorer on the left. The main editor window shows the `Producto.php` model file. The code defines a `Producto` class that extends the `Model` class and uses the `HasFactory` trait. It specifies the table name as `productos`, the primary key as `id`, and the timestamps as `false`. It also defines the fillable fields: `nombre`, `nombre_corto`, `descripcion`, `pvp`, and `familia`.

```
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7
8 class Producto extends Model
9 {
10     use HasFactory;
11
12     protected $table = 'productos'; # Nombre de la tabla en la base de datos
13     protected $primaryKey = 'id'; # Clave primaria
14     public $timestamps = false; # Deshabilitar timestamps
15
16     protected $fillable = [
17         'nombre',
18         'nombre_corto',
19         'descripcion',
20         'pvp',
21         'familia'
22     ];
23 }
24
```

y en el archivo de migración escribimos el siguiente código sobre el modelo Producto, este código nos sirve para definir tipos de campos en la futura tabla “productos”.

The image shows a Visual Studio Code editor window with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders like 'app', 'database', and 'migrations'. The code editor displays a PHP migration file named '2025\_02\_17\_144909\_create\_productos\_table.php'. The code defines a migration class that creates a 'productos' table with columns: 'id', 'nombre', 'nombre\_corto', 'descripcion', 'pvp', and 'familia'. The terminal at the bottom shows the command 'php artisan make:model Producto -m' and the successful creation of the migration file.

```
1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 return new class extends Migration
8 {
9     /**
10      * Run the migrations.
11      */
12     public function up(): void
13     {
14         if (!Schema::hasTable('productos')) {
15             Schema::create('productos', function (Blueprint $table) {
16                 $table->id();
17                 $table->string('nombre');
18                 $table->string('nombre_corto', 50);
19                 $table->text('descripcion')->nullable();
20                 $table->decimal('pvp', 8, 2);
21                 $table->string('familia');
22             });
23         }
24     }
25
26     /**
27      * Reverse the migrations.
28      */
29     public function down(): void
30     {
31         Schema::dropIfExists('productos');
32     }
33 };
34
```

PS D:\fp\ordinaria2\DWES\DWES07\suzdalenko-alexey> php artisan make:model Producto -m

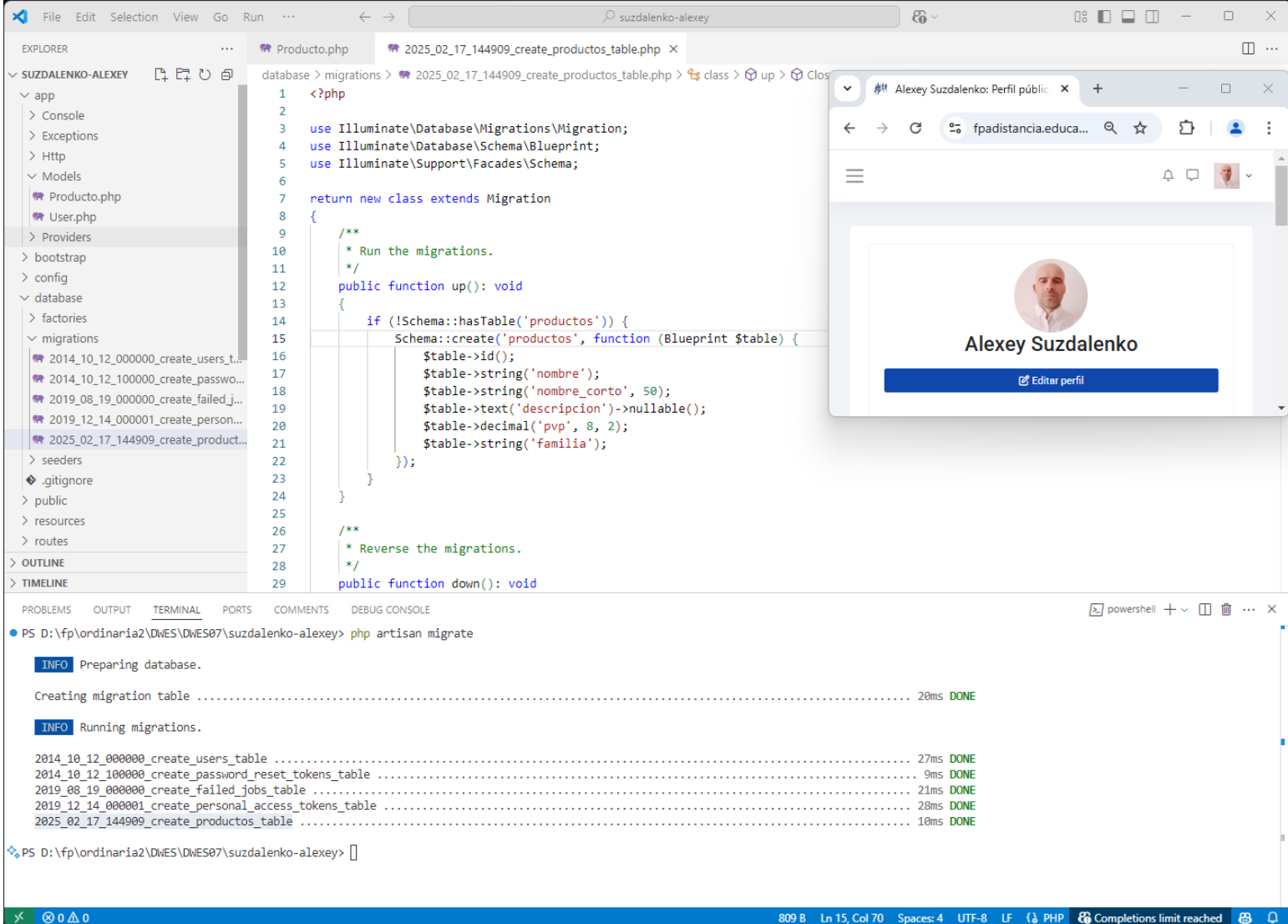
INFO Model [D:\fp\ordinaria2\DWES\DWES07\suzdalenko-alexey\app\Models\Producto.php] created successfully.

INFO Migration [D:\fp\ordinaria2\DWES\DWES07\suzdalenko-alexey\database\Migrations\2025\_02\_17\_144909\_create\_productos\_table.php] created successfully.

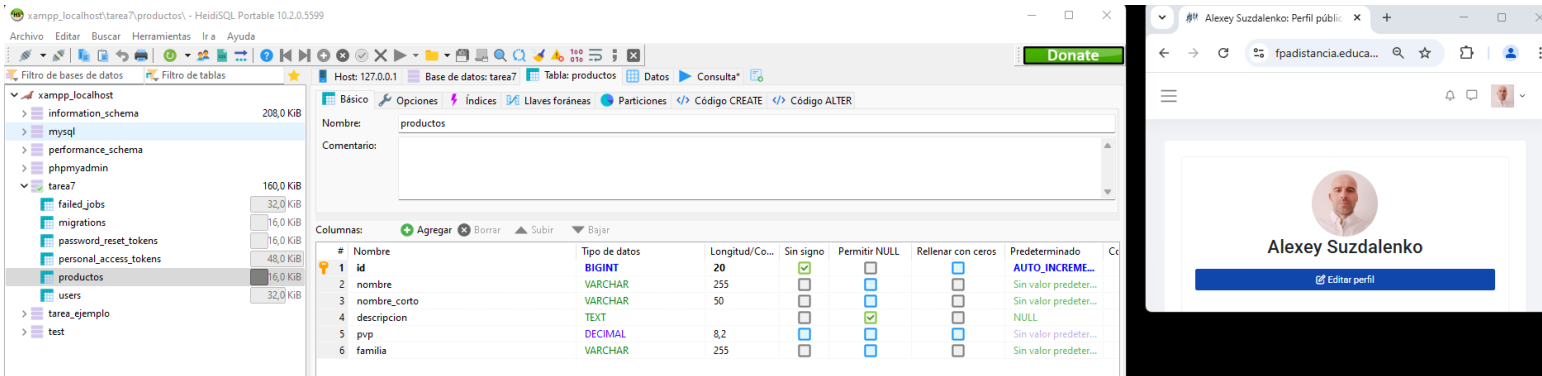
PS D:\fp\ordinaria2\DWES\DWES07\suzdalenko-alexey>

, ahora después de modificar estos dos archivos podemos proceder a hacer nuestra primera migración, que creara la tabla “productos”

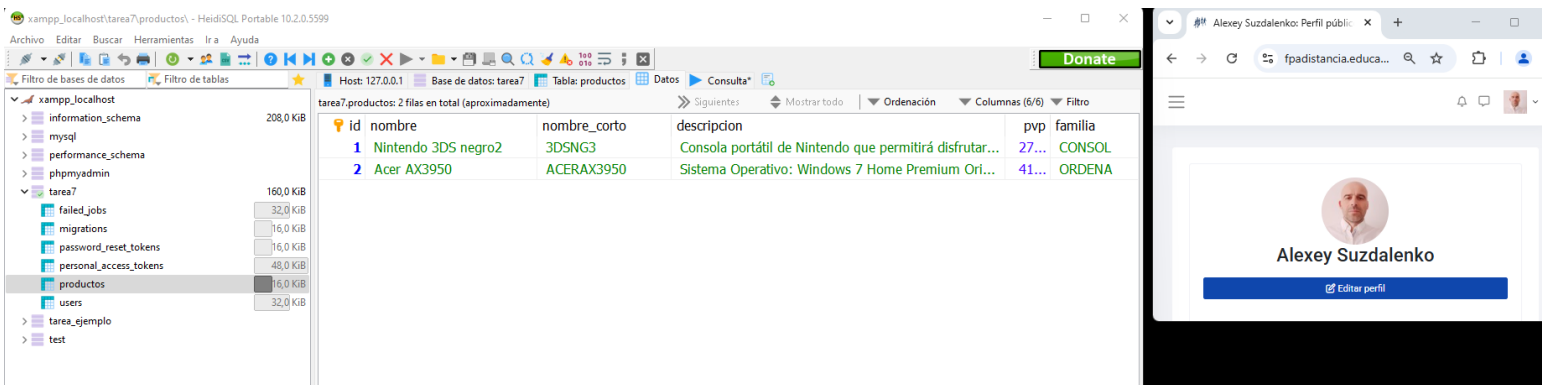
php artisan migrate



• vemos que a parte de nuestra tabla productos se han creado algunas tablas que tiene definidas por default el laravel, como tablas de gestión de usuarios, vamos a comprobar que como es la tabla productos que hemos creado:



• ahora a mano vamos a pegar los datos que necesitamos para esta tarea, pegamos datos que viene en el anuncio a la tabla productos, ya tenemos datos para trabajar:

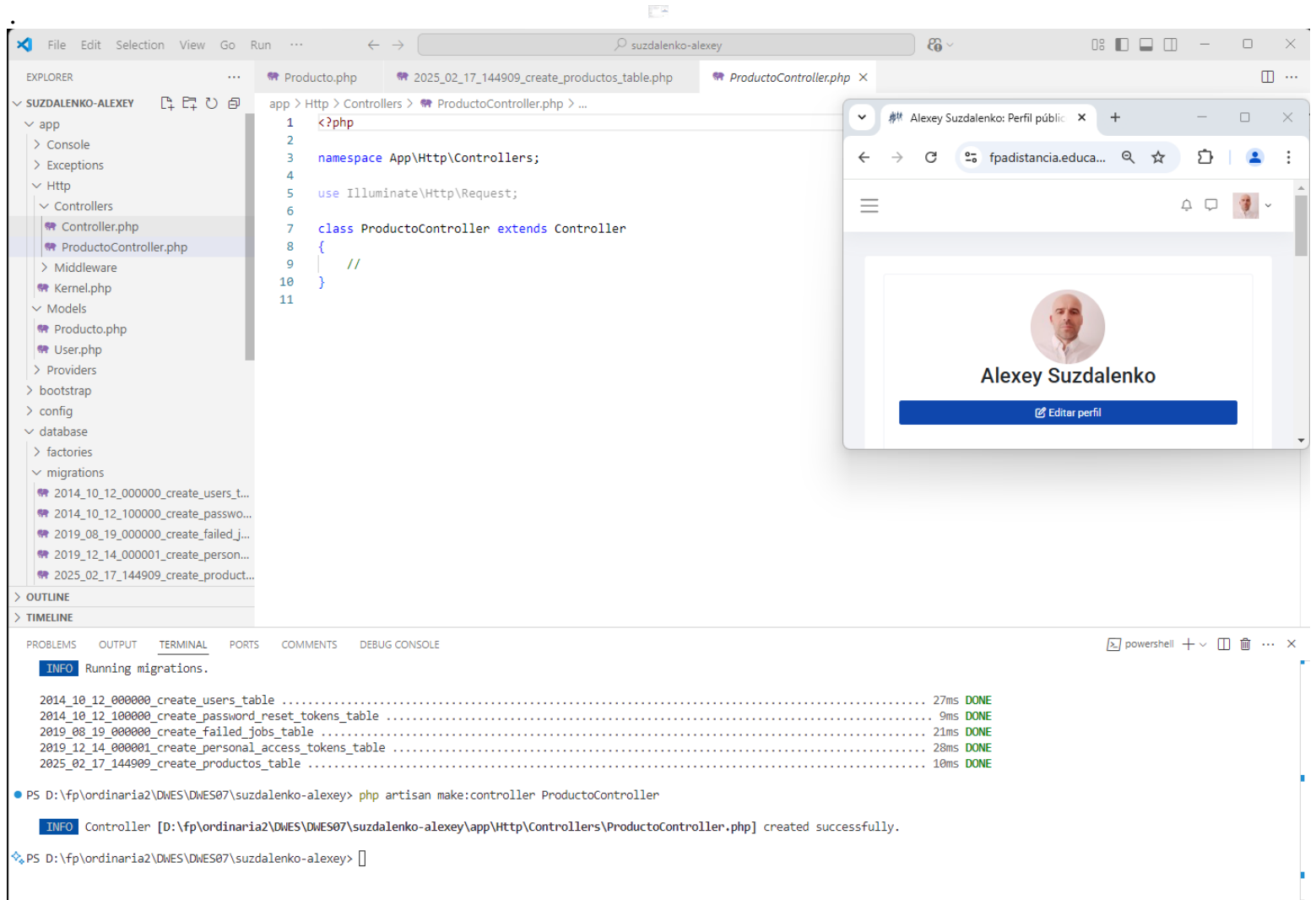


• ahora creamos un controlador para poder desde el embocar una ruta en el navegador, hacer consulta en la base de datos y mostrar los datos en pantalla a través de una vista:

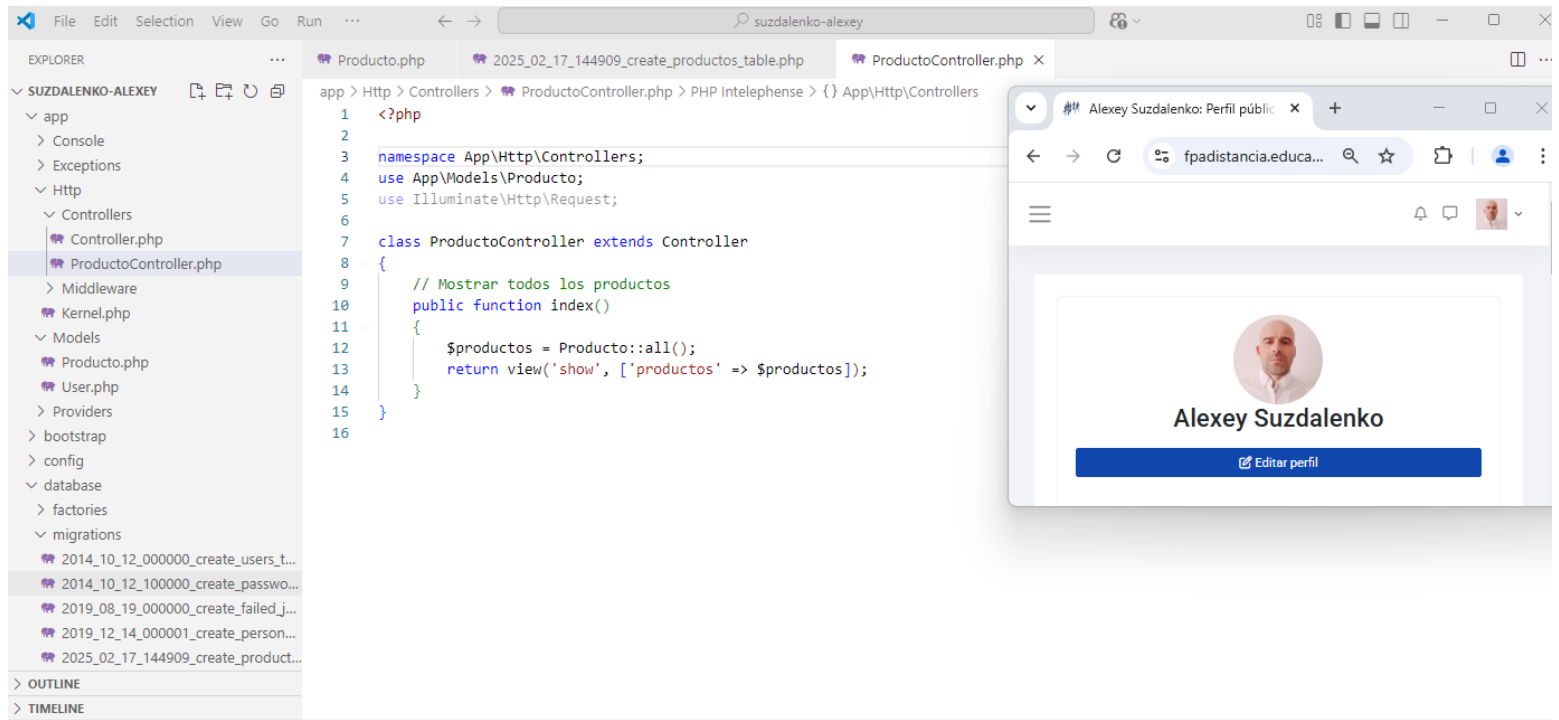


Creamos controlador:

php artisan make:controller ProductoController



se nos ha creado la clase ProductoController en la que definimos el método index() que nos sirva para mostrar la lista de productos, captura código:



y creamos una vista 'show' que se extiende de la vista 'base':

Visual Studio Code interface showing the development of a web application. The Explorer pane on the left shows the project structure, including the 'views' directory. The main editor displays the 'base.blade.php' file, which defines the layout structure with a header, main content area, and footer. The Terminal pane at the bottom shows the execution of the command `php artisan make:controller ProductoController`, which successfully creates the controller.

```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>@yield('title', 'Mi Aplicación')</title>
7 </head>
8 <body>
9     <header>
10        <h1>Mi Aplicación</h1>
11        <nav>
12            <ul>
13                <li><a href="/productos">Productos</a></li>
14            </ul>
15        </nav>
16    </header>
17    <main>
18        @yield('content')
19    </main>
20 </body>
21 </html>
22
```

Terminal Output:

```
INFO Running migrations.
2014_10_12_000000_create_users_table ..... 27ms DONE
2014_10_12_100000_create_password_reset_tokens_table ..... 9ms DONE
2019_08_19_000000_create_failed_jobs_table ..... 21ms DONE
2019_12_14_000001_create_personal_access_tokens_table ..... 28ms DONE
2025_02_17_144909_create_productos_table ..... 10ms DONE

PS D:\fp\ordinaria2\DWES\DWES07\suzdalenko-alexy> php artisan make:controller ProductoController

INFO Controller [D:\fp\ordinaria2\DWES\DWES07\suzdalenko-alexy\app\Http\Controllers\ProductoController.php] created successfully.

PS D:\fp\ordinaria2\DWES\DWES07\suzdalenko-alexy>
```

y vista 'show':

Visual Studio Code interface showing the development of the 'show' view. The Explorer pane on the left shows the project structure, including the 'views' directory. The main editor displays the 'show.blade.php' file, which defines the layout structure for the 'show' view, including a table to display the list of products. The Terminal pane at the bottom shows the execution of the command `php artisan make:controller ProductoController`, which successfully creates the controller.

```
1 @extends('base')
2 @section('title', 'Listado de Productos')
3 @section('content')
4     <h1>Listado de Productos</h1>
5
6     <table border="1">
7         <thead>
8             <tr>
9                 <th>ID</th>
10                <th>Nombre</th>
11                <th>Nombre Corto</th>
12                <th>Descripción</th>
13                <th>Precio</th>
14                <th>Familia</th>
15                <th>Editar</th>
16            </tr>
17        </thead>
18        <tbody>
19            @foreach ($productos as $producto)
20                <tr>
21                    <td>{{ $producto->id }}</td>
22                    <td>{{ $producto->nombre }}</td>
23                    <td>{{ $producto->nombre_corto }}</td>
24                    <td>{{ $producto->descripcion }}</td>
25                    <td>{{ $producto->pvp }} €</td>
26                    <td>{{ $producto->familia }}</td>
27                    <td>
28                        <a href="{{ url('/productos/' . $producto->id . '/edit') }}">Editar</a>
29                    </td>
30                </tr>
31            @endforeach
32        </tbody>
33    </table>
34 @endsection
```

Terminal Output:

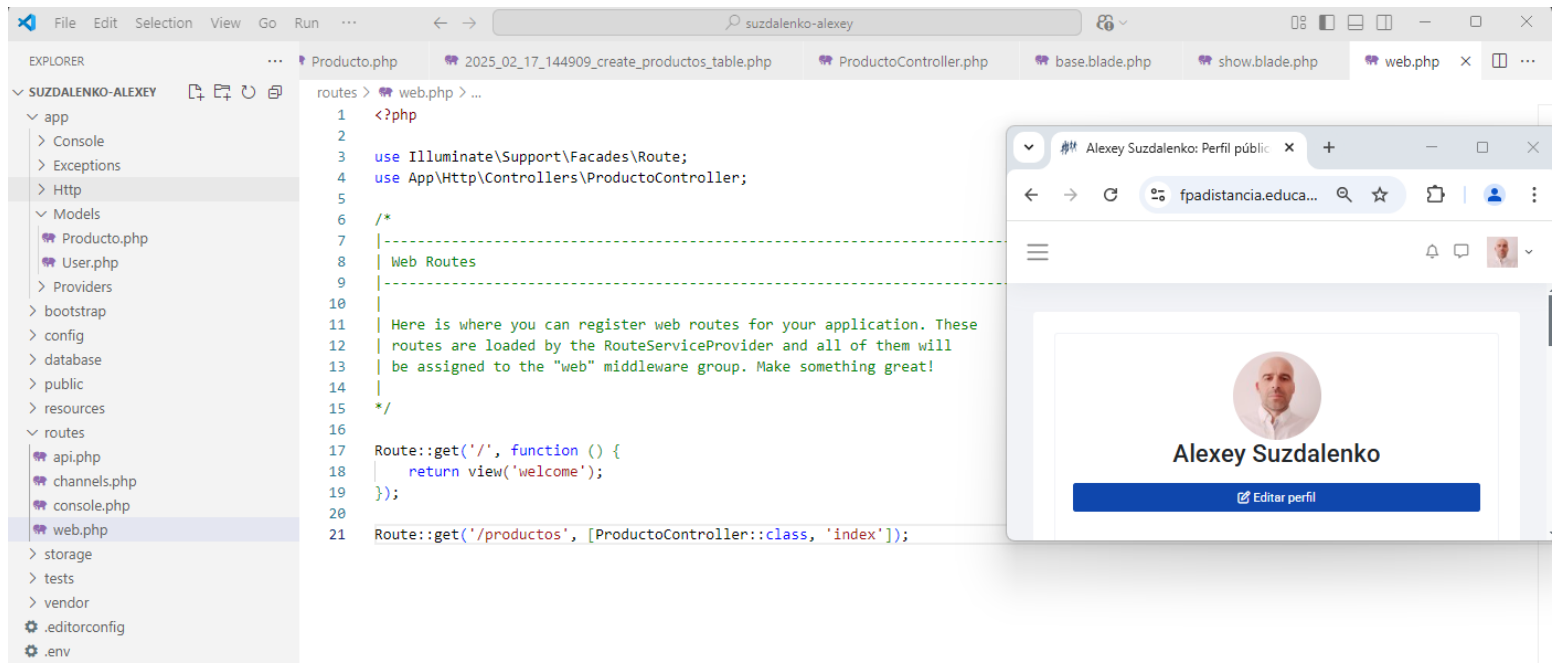
```
INFO Running migrations.
2014_10_12_000000_create_users_table ..... 27ms DONE
2014_10_12_100000_create_password_reset_tokens_table ..... 9ms DONE
2019_08_19_000000_create_failed_jobs_table ..... 21ms DONE
2019_12_14_000001_create_personal_access_tokens_table ..... 28ms DONE
2025_02_17_144909_create_productos_table ..... 10ms DONE

PS D:\fp\ordinaria2\DWES\DWES07\suzdalenko-alexy> php artisan make:controller ProductoController

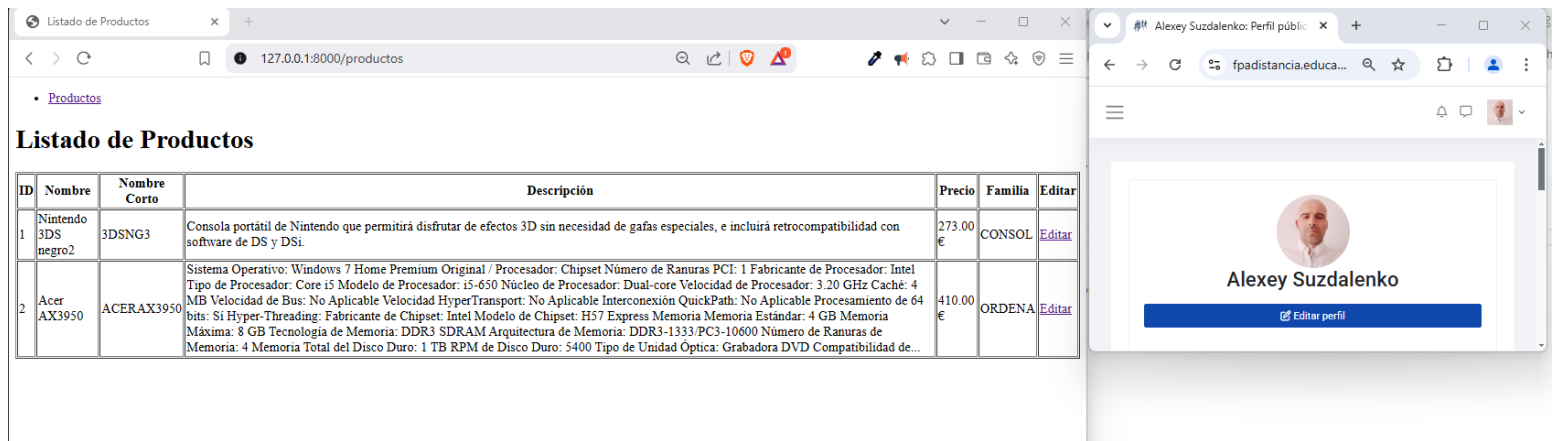
INFO Controller [D:\fp\ordinaria2\DWES\DWES07\suzdalenko-alexy\app\Http\Controllers\ProductoController.php] created successfully.

PS D:\fp\ordinaria2\DWES\DWES07\suzdalenko-alexy>
```

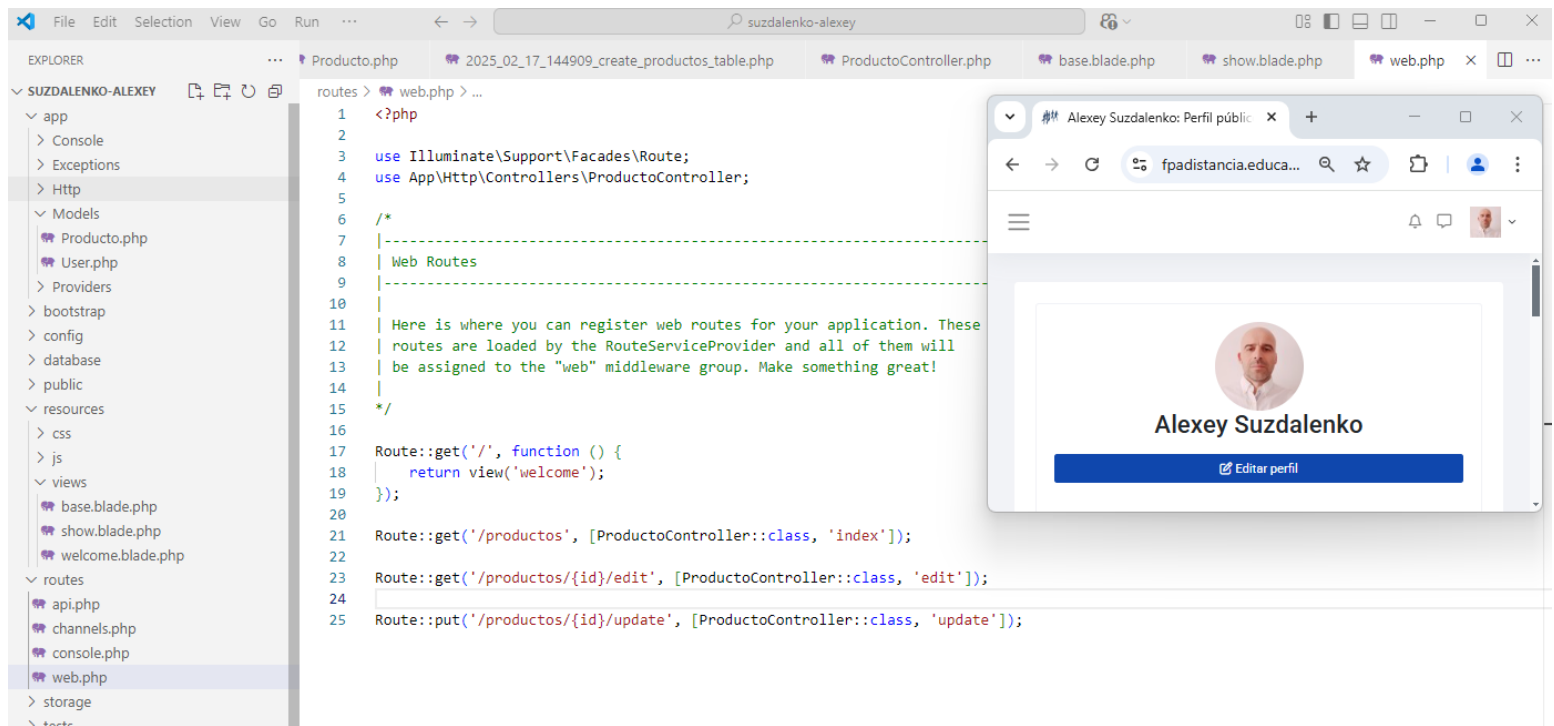
Ahora escribimos el código que va hacer la llamada el método 'index' del controlador ProductoController en el archivo web.php, código:



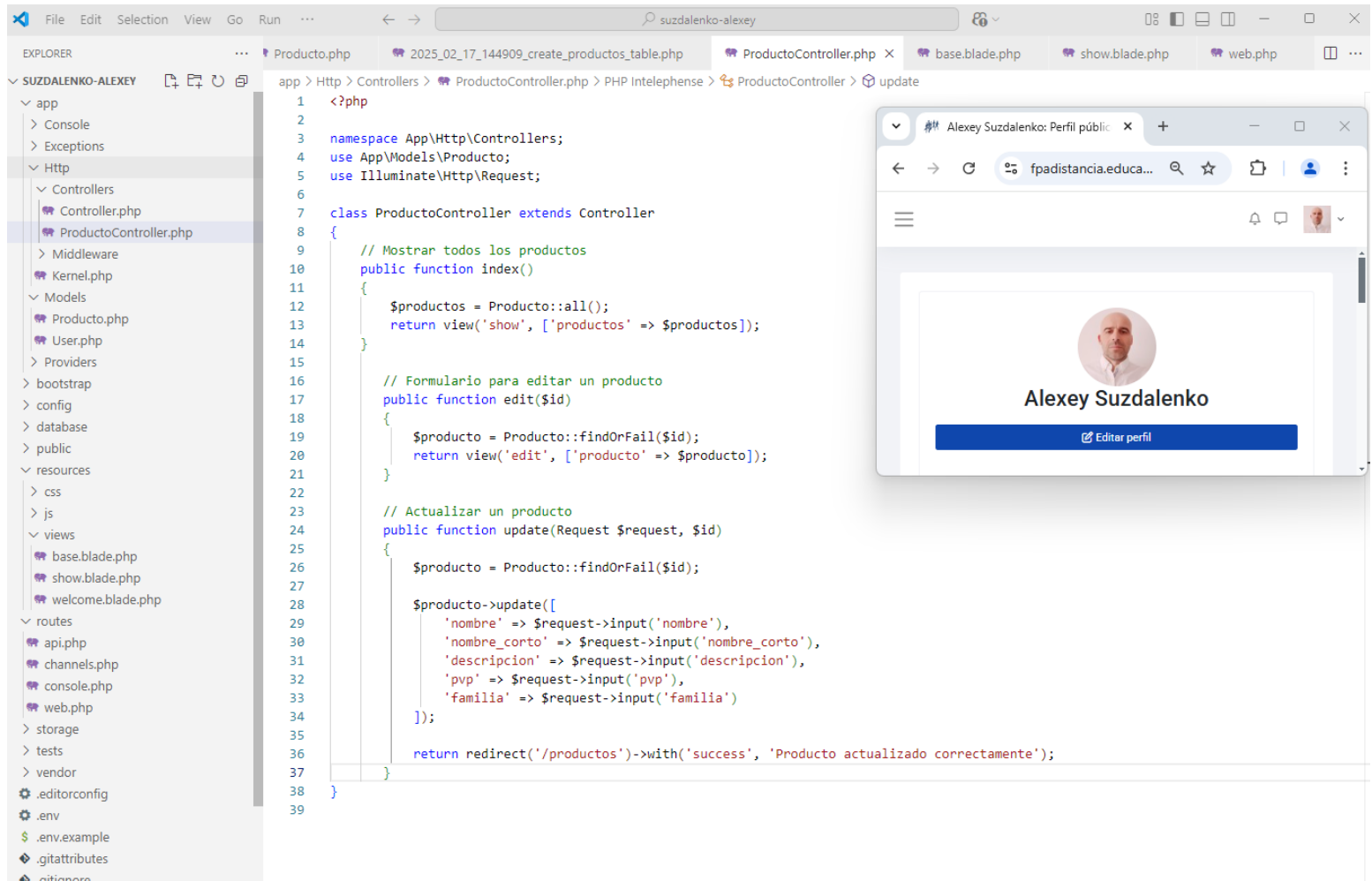
ahora abrimos navegador con esta ruta:



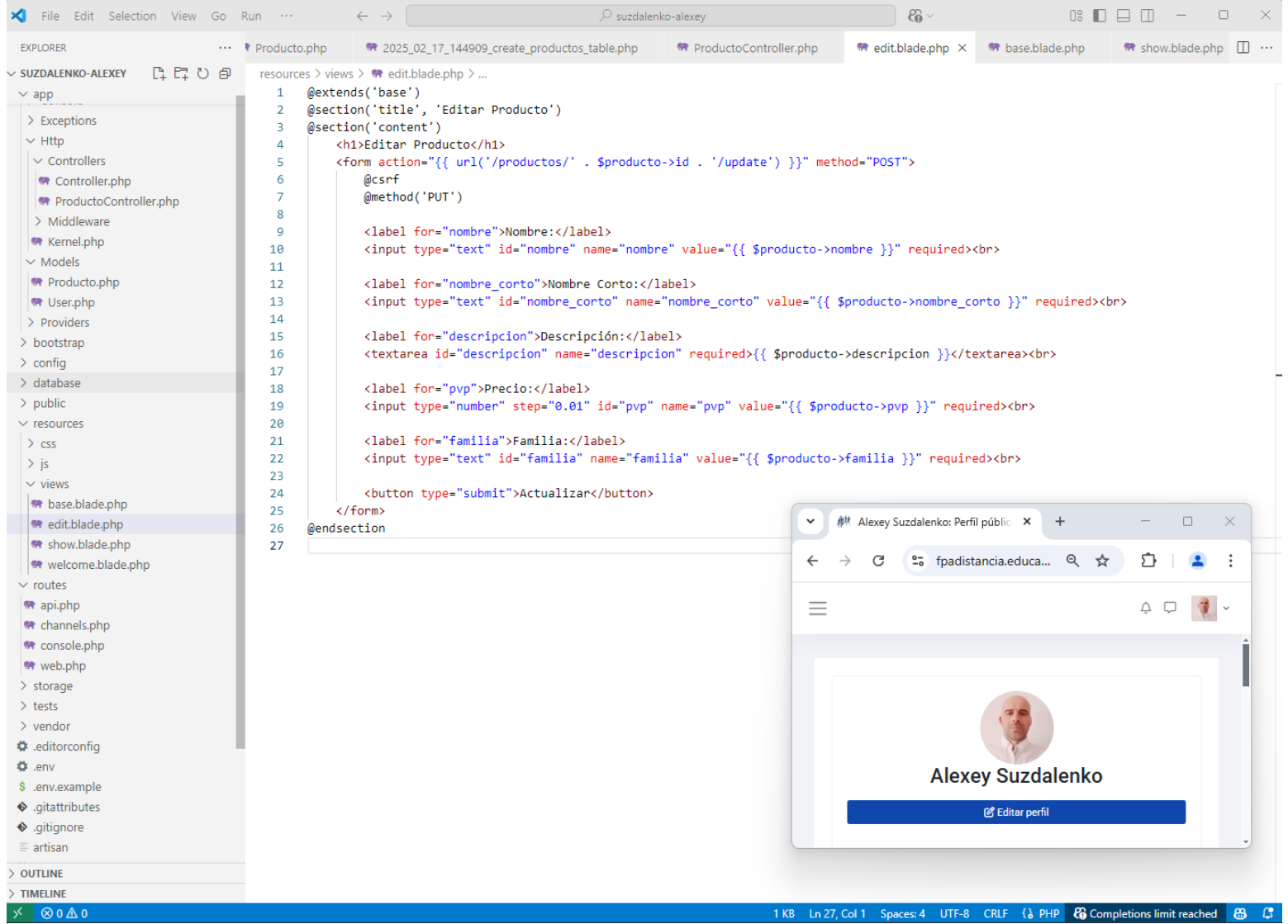
ahora creamos 2 rutas mas la de editar y de actualizar el producto, archivo web.php:



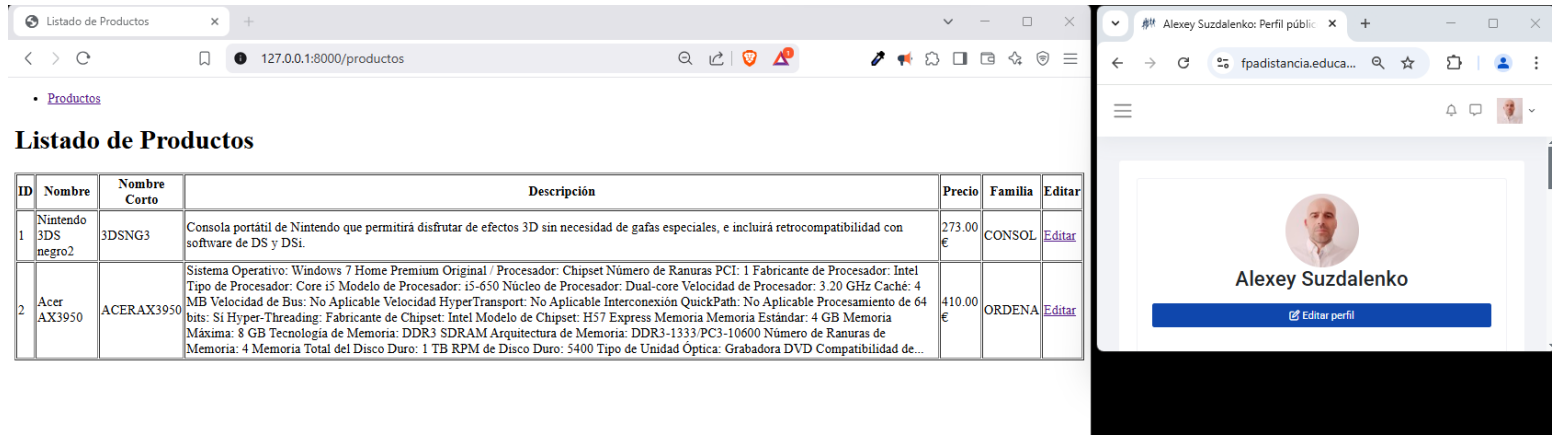
Modificamos el archivo ProductoController.php:



también necesitamos crear una vista para editar el producto, el código:



Con esto al presionar el enlace editar en la vista 'show' se nos abrirá la vista 'edit' donde podemos editar los datos del producto y actualizarlos en la base datos, vamos a comprobarlos, presionamos el enlace editar cuando estamos viendo todos los productos:



presionamos editar:

Editar Producto

127.0.0.1:8000/productos/1/edit

Productos

### Editar Producto

Nombre:

Nombre Corto:

Descripción:


Precio:

Familia:

Actualizar

Alexey Suzdalenko: Perfil público

fpadistancia.educa...



Alexey Suzdalenko

Editar perfil

Modificamos el nombre del producto por “suzdalenko prod”

Editar Producto

127.0.0.1:8000/productos/1/edit

Productos

### Editar Producto

Nombre:

Nombre Corto:

Descripción:


Precio:

Familia:

Actualizar

Alexey Suzdalenko: Perfil público

fpadistancia.educa...



Alexey Suzdalenko

Editar perfil

y presionamos actualizar:

Listado de Productos

127.0.0.1:8000/productos


Productos

### Listado de Productos

ID	Nombre	Nombre Corto	Descripción	Precio	Familia	Editar
1	suzdalenko prod	3DSNG3	Consola portátil de Nintendo que permitirá disfrutar de efectos 3D sin necesidad de gafas especiales, e incluirá retrocompatibilidad con software de DS y DSi.	273.00 €	CONSOL	Editar
2	Acer AX3950	ACERAX3950	Sistema Operativo: Windows 7 Home Premium Original / Procesador: Chipset Número de Ranuras PCI: 1 Fabricante de Procesador: Intel Tipo de Procesador: Core i5 Modelo de Procesador: i5-650 Núcleo de Procesador: Dual-core Velocidad de Procesador: 3.20 GHz Caché: 4 MB Velocidad de Bus: No Aplicable Velocidad HyperTransport: No Aplicable Interconexión QuickPath: No Aplicable Procesamiento de 64 bits: Si Hyper-Threading: Fabricante de Chipset: Intel Modelo de Chipset: H57 Express Memoria Memoria Estándar: 4 GB Memoria Máxima: 8 GB Tecnología de Memoria: DDR3 SDRAM Arquitectura de Memoria: DDR3-1333/PC3-10600 Número de Ranuras de Memoria: 4 Memoria Total del Disco Duro: 1 TB RPM de Disco Duro: 5400 Tipo de Unidad Óptica: Grabadora DVD Compatibilidad de...	410.00 €	ORDENA	Editar

Alexey Suzdalenko: Perfil público

fpadistancia.educa...



Alexey Suzdalenko

Editar perfil

Como podemos ver el nombre del primer producto se ha cambiado por el “suzdalenko prod”.

Revisamos los criterios de la tarea:

- Instalar Laravel con composer SI.
- Programar Controlador SI.
- Programar Modelo SI.
- Vista de productos SI.
- Vista de edición SI.
- Actualización del producto SI.