

MANUAL DEL PROGRAMADOR



**Adilio de Jesús Ángel Navarro
Luis Alfredo Iglesias Hernández
Henry Alexander Candelario Barrera
Luis Miguel Mártir Martínez
José Mauricio Esquivel**

**Seminario de Especialización
Ciclo I-2022
San Salvador, El salvador**



Esta obra está bajo una [Licencia Creative Commons Atribución-No Comercial-Compartir Igual 4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/)

CONTENIDO

1. REQUISITOS PARA EL DESARROLLO DEL SISTEMA.	1
2. INTRODUCCIÓN:.....	2
3. OBJETIVOS:.....	2
4. TENER INSTALADO XAMPP CON UNA VERSIÓN DE PHP ENTRE 7.1.3 Y 7.4.X.....	3
5. DIAGRAMA ENTIDAD RELACIÓN DE LA BASE DE DATOS PARA LA MICRO EMPRESA COMPUSYSTEMTC.	4
6. DICCIONARIO DE DATOS DE LA TABLA ARTÍCULOS	5
7. DICCIONARIO DE DATOS DE LA TABLA CATEGORÍAS	6
8. DICCIONARIO DE DATOS DE LA TABLA DETALLE_INGRESOS.	6
9. DICCIONARIO DE DATOS DE LA TABLA DETALLE_VENTA.	7
10. DICCIONARIO DE DATOS DE LA TABLA INGRESOS.	8
11. DICCIONARIO DE DATOS DE LA TABLA PERSONAS.....	9
12. DICCIONARIO DE DATOS DE LA TABLA PROVEEDORES.....	10
13. DESCARGAR E INSTALAR COMPOSER IR A: HTTPS://GETCOMPOSER.ORG/DOWNLOAD/ ...	10
13.1 Vista de la página web de composer.	10
14. ABRIR UNA TERMINAL CON LA UBICACIÓN: C:/XAMPP/HTDOCS.....	11
14.1 Vista de la terminal.....	11
15. ENTRAR A LA DOCUMENTACIÓN DE LARAVEL: HTTPS://LARAVEL.COM/DOCS/5.6	11
15.1 Vista de la documentación de laravel 5.6.....	11
16. ESCRIBIR EL COMANDO EN LA TERMINAL: COMPOSER CREATE-PROJECT --PREFER-DIST LARAVEL/LARAVEL SISTEMA "5.6.*".	12
16.1 Vista de la terminal para crear un nuevo proyecto Laravel.....	12
17. VERÁ COMO SE CREÓ UNA CARPETA LLAMADA: SISTEMA, DENTRO DE HTDOCS.	12
17.1 Vista de la creación del proyecto en htdocs.....	12
18. ABRIR EL PROYECTO EN VISUAL STUDIO CODE.....	13
18.1 Vista del proyecto en visual studio code.....	13
19. NOS DIRIGIMOS A PHPMYADMIN Y CREAR SOLAMENTE LA BASE DE DATOS: DBVENTAS, SIN LAS TABLAS SOLO LA BD.....	14
19.1 Vista de la creación de la base de datos en phpmyadmin.....	14

20.	LA PRIMERA TABLA QUE CREAREMOS SERÁ: CATEGORÍA, ASÍ: TERMINAL: PHP ARTISAN MAKE:MIGRATION CREATE_CATEGORIAS_TABLE.	14
20.1	Vista de la tabla categoría que creamos.	14
21.	REVISAMOS LA CONEXIÓN EN EL ARCHIVO SISTEMA/.ENV.....	15
21.1	Vista del archivo .env a configurar.....	15
22.	EJECUTAMOS LA MIGRACIÓN EN LATERMINAL: PHP ARTISAN MÍGRATE	16
22.1	Vista de la terminal para efectuar la migración a la base de datos.	16
23.	VEREMOS LAS MIGRACIONES EJECUTADAS Y DEBEMOS REVISAR LA BD EN PHPMYADMIN16	
23.1	Vista de todas las migraciones efectuadas en phpmyadmin.....	16
24.	CREAREMOS LO MODELOS PARA EL CUAL UTILIZAMOS EL SIGUIENTE COMANDO TERMINAL: PHP ARTISAN MAKE:MODEL CATEGORÍA AL QUERER CREAR OTRO MODELO SOLO CAMBIAMOS LA PALABRA CATEGORÍA POR EL NOMBRE DEL MODELO QUE QUEREMOS ASIGNAR. 17	
24.1	Vista de la terminal para crear un modelo.	17
24.2	En la carpeta app encontraremos los modelos que creamos	17
25.	CREAMOS EL CONTROLADOR CATEGORIACONTROLLER, PARA CREAR LOS DEMÁS CONTROLADORES SOLO CAMBIAMOS LA PALABRA CATEGORIACONTROLLER POR LA QUE QUERAMOS DESIGNAR.	18
25.1	Vista de la terminal para crear los controladores.	18
26.	CREAREMOS EL COMPONENTE Y ESTA UBICADO EN LA CARPETA RESOURCES/ASSETS/JS/COMPONENTS/ EXAMPLECOMPONENT.VUE Y COPIAMOS EXAMPLECOMPONENT.VUE Y LO GUARDAMOS COMO CATEGORIA.VUE A SI MISMO SERÁ PARA LOS DEMÁS COMPONENTES A CREAR.	18
26.1	Vista de donde esta ubicados los componetes.vue	18
27.	EN LA CARPETA PUBLIC ENCONTRAREMOS LOS CSS, FONTS, IMÁGENES Y LOS JS	19
27.1	<i>Vista de la carpeta public.</i>	19
28.	EN LA CARPETA ROUTES ENCONTRAREMOS UN ARCHIVO LLAMADO WEB.PHP EN ESTE ARCHIVO ES DONDE VAN TODAS LAS RUTAS.....	19
28.1	Vista de la carpeta routes.....	19
29.	CARPETA AUTH.....	20
29.1	Vista de la carpeta auth y controladores.	20
30.	DIRECTORIO.....	21
30.1	Vista de directorio para logincontroller.php	21

ÍNDICE DE TABLAS

<i>Tabla 1: Diccionario de datos de la tabla artículos</i>	<i>5</i>
<i>Tabla 2: Diccionario de datos de la tabla categorías</i>	<i>6</i>
<i>Tabla 3: Diccionario de datos de la tabla detalle ingreso</i>	<i>6</i>
<i>Tabla 4: Diccionario de datos de la tabla detalle venta</i>	<i>7</i>
<i>Tabla 5: Diccionario de datos de la tabla ingresos.....</i>	<i>8</i>
<i>Tabla 6: Diccionario de datos de la tabla personas</i>	<i>9</i>
<i>Tabla 7: Diccionario de datos de la tabla proveedores</i>	<i>10</i>

ÍNDICE DE ILUSTRACIONES

<i>Ilustración 1: Vista del Panel de XAMPP</i>	<i>3</i>
<i>Ilustración 2: Diagrama de la base de datos.....</i>	<i>4</i>
<i>Ilustración 3: Composer.....</i>	<i>10</i>
<i>Ilustración 4: Vista abrir terminal.....</i>	<i>11</i>
<i>Ilustración 5: Documentación Laravel.....</i>	<i>11</i>
<i>Ilustración 6: Comando para crear el proyecto usando composer.....</i>	<i>12</i>
<i>Ilustración 7: Vista de la carpeta creada en httdocs</i>	<i>13</i>
<i>Ilustración 8: Vista de las carpetas en Visual Studio Code</i>	<i>13</i>
<i>Ilustración 9: Creación del nombre de la base de datos.....</i>	<i>14</i>
<i>Ilustración 10: Migraciones.....</i>	<i>14</i>
<i>Ilustración 11: Archivo .env cambiar el nombre de la base de datos.....</i>	<i>15</i>
<i>Ilustración 12: Comando para efectuar las migraciones.....</i>	<i>16</i>
<i>Ilustración 13: Base de datos con todas las tablas migradas</i>	<i>16</i>
<i>Ilustración 14: Creación del modelo</i>	<i>17</i>
<i>Ilustración 15: Modelos Creados</i>	<i>17</i>
<i>Ilustración 16: Crear Controladores</i>	<i>18</i>
<i>Ilustración 17: Componentes.....</i>	<i>18</i>
<i>Ilustración 18: Vista de la carpeta public</i>	<i>19</i>
<i>Ilustración 19: Rutas.....</i>	<i>19</i>
<i>Ilustración 20: Archivo Web.php donde irán las rutas</i>	<i>19</i>
<i>Ilustración 21: Carpeta de los controladores</i>	<i>20</i>
<i>Ilustración 22: LoginController</i>	<i>21</i>
<i>Ilustración 23: Vista del Login</i>	<i>21</i>

1. REQUISITOS PARA EL DESARROLLO DEL SISTEMA.

- ✓ Servidor Apache
- ✓ Servidor de MySQL
- ✓ Laravel versión 5.6
- ✓ Vue.js
- ✓ Node.js 8.9.4
- ✓ Npm 5.6
- ✓ Axios
- ✓ SweetAlert
- ✓ Npm install SweetAlert2
- ✓ Vue-barcode
- ✓ Npm install vue-barcode
- ✓ Vue-select
- ✓ Npm install vue-select2
- ✓ DomPDF
- ✓ Composer require barryvdh/laravel-dompdf":0.8.2"
- ✓ Visual Studio Code
- ✓ Navegador Google Chrome

2. INTRODUCCIÓN:

En este manual se detalla el código que se utilizó para la elaboración del **“Sistema de Compras y Ventas”** para la micro empresa CompusystemTC.

El propósito de brindar este manual de programador es para que en un futuro si quieren realizar actualizaciones al sistema, el usuario tenga una guía referente de cómo está elaborado.

3. OBJETIVOS:

- ✓ Objetivo General:
 - Ayudar a los programadores a conocer el desarrollo del sistema de compras y ventas para la micro empresa CompusystemTc.

- ✓ Objetivos Específicos:
 - Describir el código utilizado para el desarrollo del sistema.
 - Explicar de manera clara los pasos a seguir para modificar el sistema.

4. TENER INSTALADO XAMPP CON UNA VERSIÓN DE PHP ENTRE 7.1.3 Y 7.4.X.

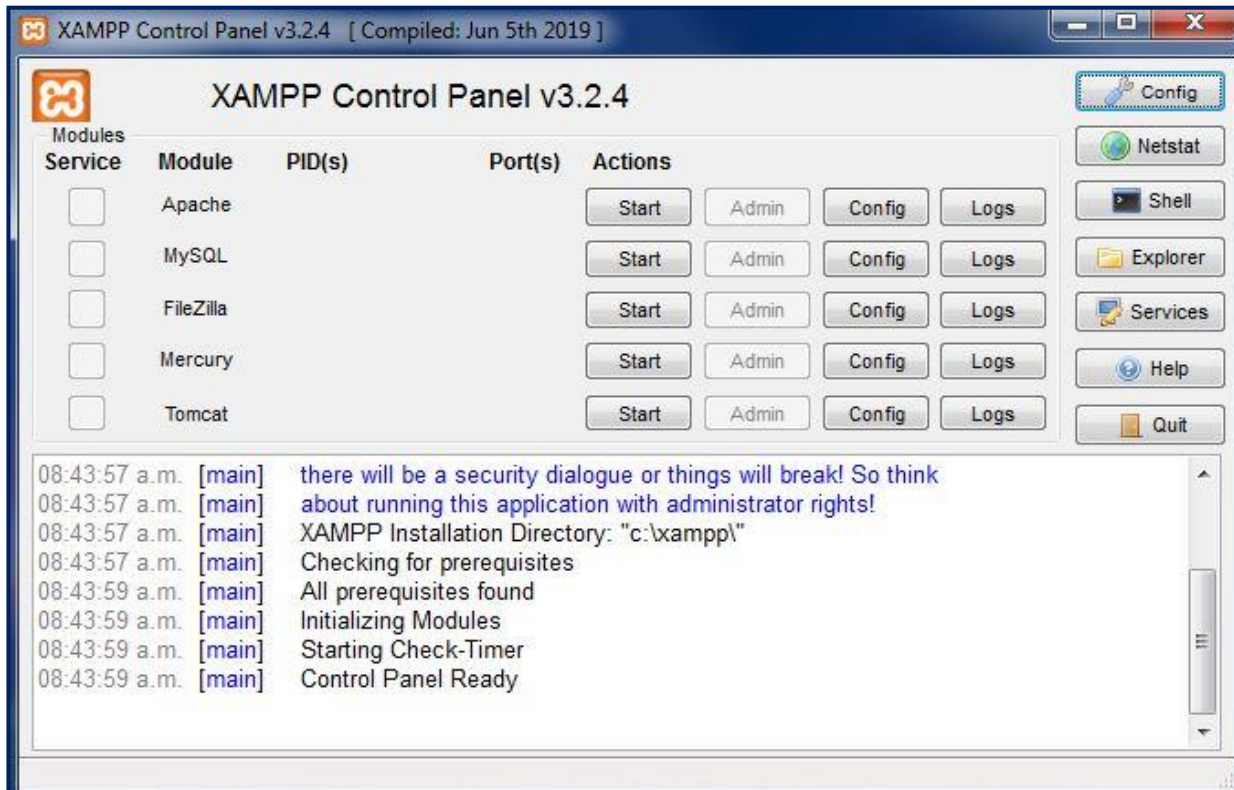


Ilustración 1: Vista del Panel de XAMPP

5. DIAGRAMA ENTIDAD RELACIÓN DE LA BASE DE DATOS PARA LA MICRO EMPRESA COMPUSYSTEMTC.

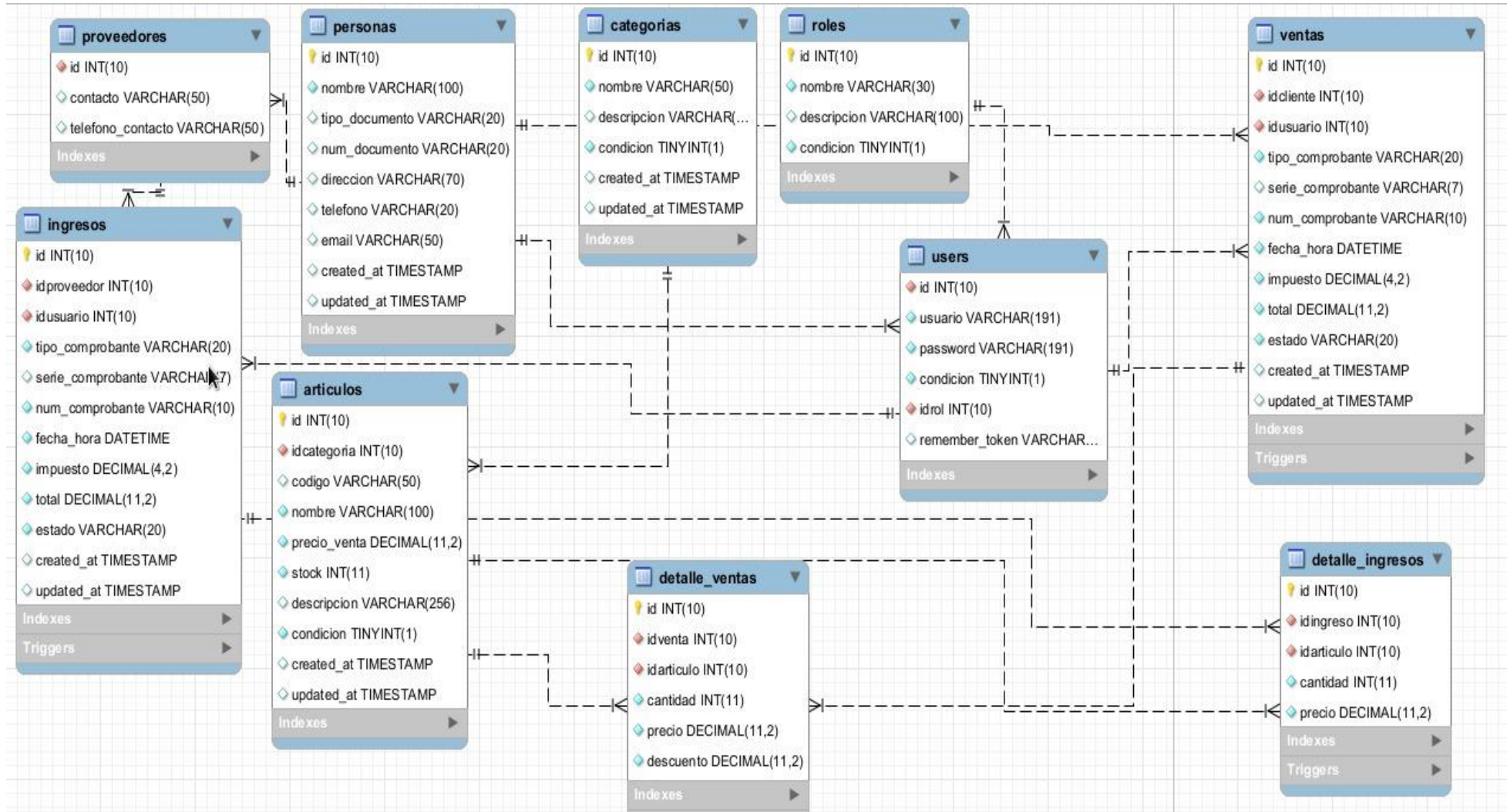


Ilustración 2: Diagrama de la base de datos

6. DICCIONARIO DE DATOS DE LA TABLA ARTÍCULOS .

Columna	Tipo	Nul o	Predeterminad o	Enlaces a	Comentario s	Medi a type
id (<i>Primaria</i>)	int(10)	No				
idcategoria	int(10)	No		categoria s -> id		
codigo	varchar(50)	Sí	<i>NULL</i>			
nombre	varchar(100)	No				
precio_venta	decimal(11,2)	No				
stock	int(11)	No				
descripcion	varchar(256)	Sí	<i>NULL</i>			
condicion	tinyint(1)	No	1			
created_at	timestamp	Sí	<i>NULL</i>			
updated_at	timestamp	Sí	<i>NULL</i>			

Tabla 1: Diccionario de datos de la tabla artículos

7. DICCIONARIO DE DATOS DE LA TABLA CATEGORÍAS .

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	Media type
id (Primaria)	int(10)	No				
nombre	varchar(50)	No				
descripcion	varchar(256)	Sí	NULL			
condicion	tinyint(1)	No	1			
created_at	timestamp	Sí	NULL			
updated_at	timestamp	Sí	NULL			

Tabla 2: Diccionario de datos de la tabla categorías

8. DICCIONARIO DE DATOS DE LA TABLA DETALLE_INGRESOS.

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	Media type
id (Primaria)	int(10)	No				
idingreso	int(10)	No		ingresos -> id		
idarticulo	int(10)	No		articulos -> id		
cantidad	int(11)	No				
precio	decimal(11,2)	No				

Tabla 3: Diccionario de datos de la tabla detalle ingreso

9. DICCIONARIO DE DATOS DE LA TABLA DETALLE_VENTA.

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	Media type
id (Primaria)	int(10)	No				
idventa	int(10)	No		ventas - > id		
idarticulo	int(10)	No		articulos -> id		
cantidad	int(11)	No				
precio	decimal(11,2)	No				
descuento	decimal(11,2)	No				

Tabla 4: Diccionario de datos de la tabla detalle venta

10. DICCIONARIO DE DATOS DE LA TABLA INGRESOS.

Columna	Tipo	Nul o	Predetermina do	Enlaces a	Comentari os	Medi a type
id (Primaria)	int(10)	No				
idproveedor	int(10)	No		proveedor es -> id		
idusuario	int(10)	No		users -> id		
tipo_comprobante	varchar(20)	No				
serie_comprobante	varchar(7)	Sí	NULL			
num_comprobante	varchar(10)	No				
fecha_hora	datetime	No				
impuesto	decimal(4,2)	No				
total	decimal(11,2)	No				
estado	varchar(20)	No				
created_at	timestamp	Sí	NULL			
updated_at	timestamp	Sí	NULL			

Tabla 5: Diccionario de datos de la tabla ingresos

11. DICCIONARIO DE DATOS DE LA TABLA PERSONAS.

Columna	Tipo	Nul o	Predeterminad o	Enlace s a	Comentarios	Medi a type
id (Primaria)	int(10)	No				
nombre	varchar(100)	No				
tipo_documento	varchar(20)	Sí	NULL			
num_documento	varchar(20)	Sí	NULL			
direccion	varchar(70)	Sí	NULL			
telefono	varchar(20)	Sí	NULL			
email	varchar(50)	Sí	NULL			
created_at	timestamp	Sí	NULL			
updated_at	timestamp	Sí	NULL			

Tabla 6: Diccionario de datos de la tabla personas

12. DICCIONARIO DE DATOS DE LA TABLA PROVEEDORES.

Columna	Tipo	Nul o	Predeterminad o	Enlaces a	Comentarios	Medi a type
id	int(10)	No		personas -> id		
contacto	varchar(50)	Sí	NULL			
telefono_contacto	varchar(50)	Sí	NULL			

Tabla 7: Diccionario de datos de la tabla proveedores

13. DESCARGAR E INSTALAR COMPOSER IR A: [HTTPS://GETCOMPOSER.ORG/DOWNLOAD/](https://getcomposer.org/download/).

13.1 Vista de la página web de composer.



Ilustración 3: Composer

14. ABRIR UNA TERMINAL CON LA UBICACIÓN: C:/XAMPP/HTDOCS.

14.1 Vista de la terminal.

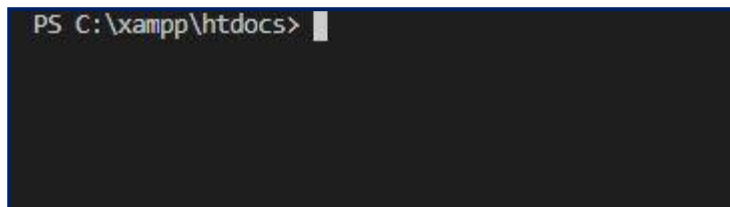


Ilustración 4: Vista abrir terminal

15. ENTRAR A LA DOCUMENTACIÓN DE LARAVEL:
[HTTPS://LARAVEL.COM/DOCS/5.6.](https://laravel.com/docs/5.6)

15.1 Vista de la documentación de laravel 5.6.

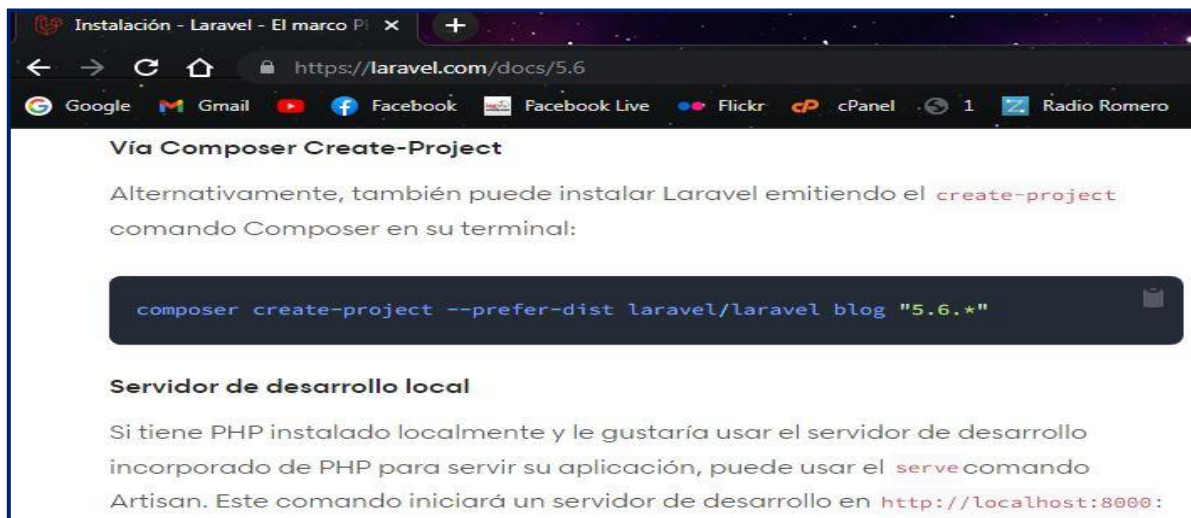


Ilustración 5: Documentación Laravel

16. ESCRIBIR EL COMANDO EN LA TERMINAL: COMPOSER CREATE-PROJECT --PREFER-DIST LARAVEL/LARAVEL SISTEMA "5.6.*".

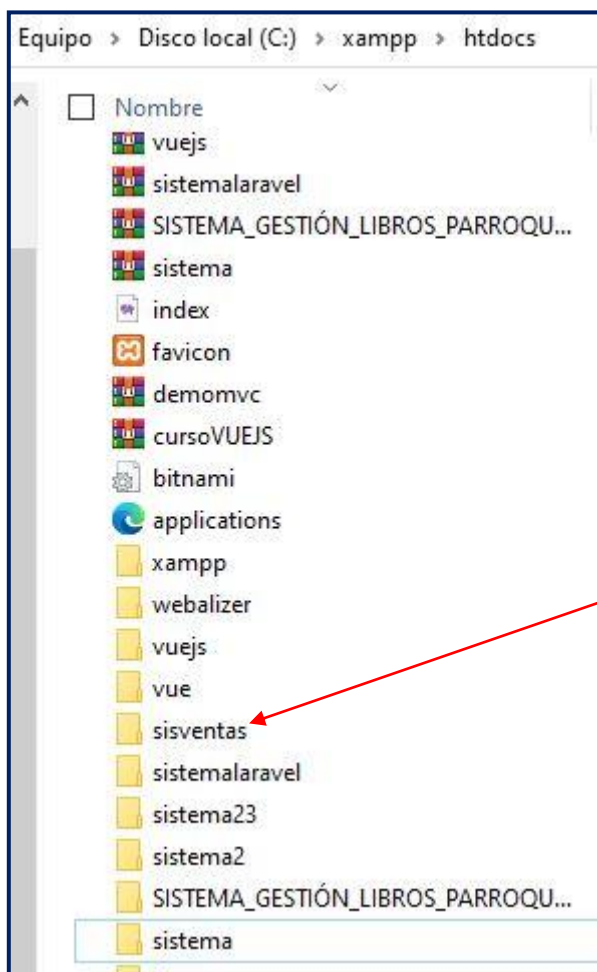
16.1 Vista de la terminal para crear un nuevo proyecto Laravel

```
PS C:\xampp\htdocs\sisventas> cd /xampp/htdocs  
PS C:\xampp\htdocs> composer create-project --prefer-dist laravel/laravel sistema "5.6.*"
```

Ilustración 6: Comando para crear el proyecto usando composer

17. VERÁ COMO SE CREÓ UNA CARPETA LLAMADA: SISTEMA, DENTRO DE HTDOCS.

17.1 Vista de la creación del proyecto en htdocs.



Nos dirigimos a la carpeta htdocs de nuestra pc para corroborar que se creó el proyecto Laravel.

Ilustración 7: Vista de la carpeta creada en htdocs

18. ABRIR EL PROYECTO EN VISUAL STUDIO CODE.

18.1 Vista del proyecto en visual studio code.

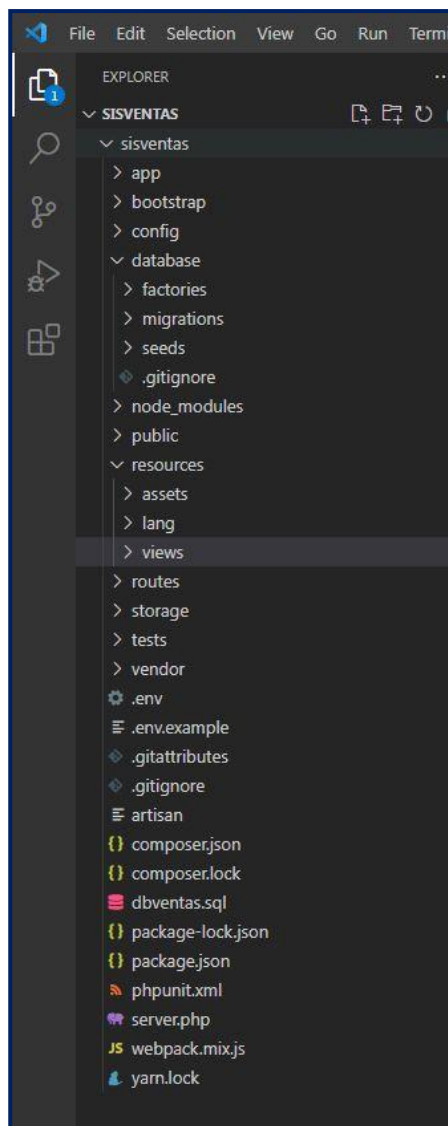


Ilustración 8: Vista de las carpetas en Visual Studio Code

19. NOS DIRIGIMOS A PHPMYADMIN Y CREAR SOLAMENTE LA BASE DE DATOS: DBVENTAS, SIN LAS TABLAS SOLO LA BD.

19.1 Vista de la creación de la base de datos en phpmyadmin.



Ilustración 9: Creación del nombre de la base de datos

20. LA PRIMERA TABLA QUE CREAREMOS SERÁ: CATEGORÍA, ASÍ: TERMINAL: PHP ARTISAN MAKE:MIGRATION CREATE_CATEGORIAS_TABLE.

20.1 Vista de la tabla categoría que creamos.

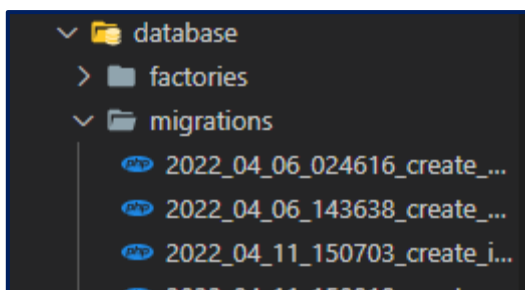


Ilustración 10: Migraciones

21. REVISAMOS LA CONEXIÓN EN EL ARCHIVO SISTEMA/.ENV.

🌈 Cambiamos los parámetros de la conexión así:

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=dbventas
DB_USERNAME=root
DB_PASSWORD=
```

21.1 Vista del archivo .env a configurar.

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=dbventas
DB_USERNAME=root
DB_PASSWORD=
```

Ilustración 11: Archivo .env cambiar el nombre de la base de datos

22. EJECUTAMOS LA MIGRACIÓN EN LATERMINAL: PHP ARTISAN MÍGRATE

22.1 Vista de la terminal para efectuar la migración a la base de datos.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. Reservados todos los derechos.

PS C:\xampp\htdocs\sisventas> php artisan migrate
```

Ilustración 12: Comando para efectuar las migraciones

23. VEREMOS LAS MIGRACIONES EJECUTADAS Y DEBEMOS REVISAR LA BD EN PHPMYADMIN

23.1 Vista de todas las migraciones efectuadas en phpmyadmin.

Filtros

Que contengan la palabra:

<input type="checkbox"/>	Tabla	Acción	Filas	Tipo	Cotejamiento	Tamaño	Residuo a depurar	
<input type="checkbox"/>	articulos	Examinar Estructura Buscar Insertar Vaciar Eliminar	3	InnoDB	utf8mb4_unicode_ci	48.0 KB	-	
<input type="checkbox"/>	categorias	Examinar Estructura Buscar Insertar Vaciar Eliminar	5	InnoDB	utf8mb4_unicode_ci	16.0 KB	-	
<input type="checkbox"/>	detalle_ingresos	Examinar Estructura Buscar Insertar Vaciar Eliminar	10	InnoDB	utf8mb4_unicode_ci	48.0 KB	-	
<input type="checkbox"/>	detalle_ventas	Examinar Estructura Buscar Insertar Vaciar Eliminar	13	InnoDB	utf8mb4_unicode_ci	48.0 KB	-	
<input type="checkbox"/>	ingresos	Examinar Estructura Buscar Insertar Vaciar Eliminar	6	InnoDB	utf8mb4_unicode_ci	48.0 KB	-	
<input type="checkbox"/>	migrations	Examinar Estructura Buscar Insertar Vaciar Eliminar	12	InnoDB	utf8mb4_unicode_ci	16.0 KB	-	
<input type="checkbox"/>	notifications	Examinar Estructura Buscar Insertar Vaciar Eliminar	27	InnoDB	utf8mb4_unicode_ci	32.0 KB	-	
<input type="checkbox"/>	password_resets	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8mb4_unicode_ci	32.0 KB	-	
<input type="checkbox"/>	personas	Examinar Estructura Buscar Insertar Vaciar Eliminar	7	InnoDB	utf8mb4_unicode_ci	32.0 KB	-	
<input type="checkbox"/>	proveedores	Examinar Estructura Buscar Insertar Vaciar Eliminar	2	InnoDB	utf8mb4_unicode_ci	32.0 KB	-	
<input type="checkbox"/>	roles	Examinar Estructura Buscar Insertar Vaciar Eliminar	3	InnoDB	utf8mb4_unicode_ci	32.0 KB	-	
<input type="checkbox"/>	users	Examinar Estructura Buscar Insertar Vaciar Eliminar	3	InnoDB	utf8mb4_unicode_ci	48.0 KB	-	
<input type="checkbox"/>	ventas	Examinar Estructura Buscar Insertar Vaciar Eliminar	9	InnoDB	utf8mb4_unicode_ci	48.0 KB	-	
13 tablas			Número de filas	100	InnoDB	utf8mb4_general_ci	480.0 KB	0 B

Ilustración 13: Base de datos con todas las tablas migradas

24. CREAREMOS LOS MODELOS PARA EL CUAL UTILIZAMOS EL SIGUIENTE COMANDO TERMINAL: `PHP ARTISAN MAKE:MODEL CATEGORÍA` AL QUERER CREAR OTRO MODELO SOLO CAMBIAMOS LA PALABRA CATEGORÍA POR EL NOMBRE DEL MODELO QUE QUEREMOS ASIGNAR.

24.1 Vista de la terminal para crear un modelo.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\xampp\htdocs\sistema> php artisan make:model Categoria
```

Ilustración 14: Creación del modelo

24.2 En la carpeta app encontraremos los modelos que creemos

Al desplegar la carpeta app encontraremos los modelos.

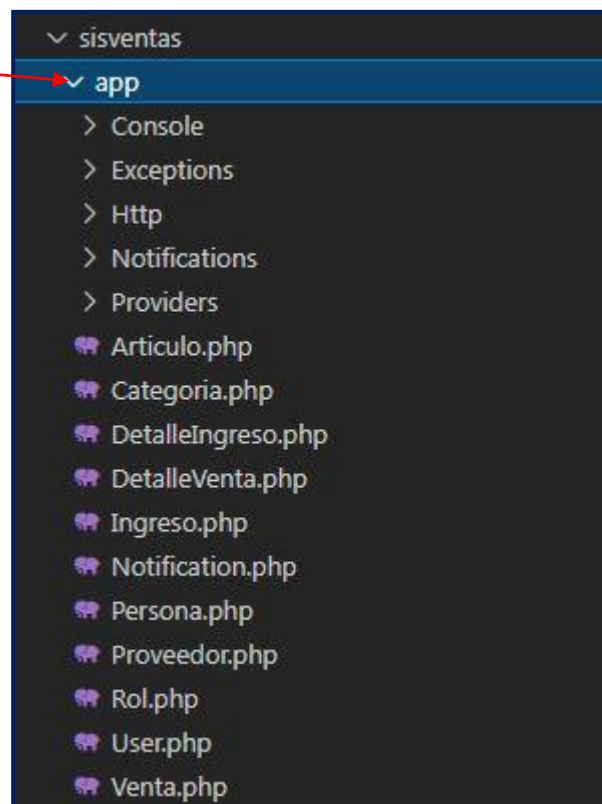


Ilustración 15: Modelos Creados

25. CREAMOS EL CONTROLADOR CATEGORIACONTROLLER, PARA CREAR LOS DEMÁS CONTROLADORES SOLO CAMBIAMOS LA PALABRA CATEGORIACONTROLLER POR LA QUE QUERAMOS DESIGNAR.

- Terminal: `php artisan make:controller CategoriaController --resource`
- Se crea en `app/Http/Controllers`

25.1 Vista de la terminal para crear los controladores.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. Reservados todos los derechos.

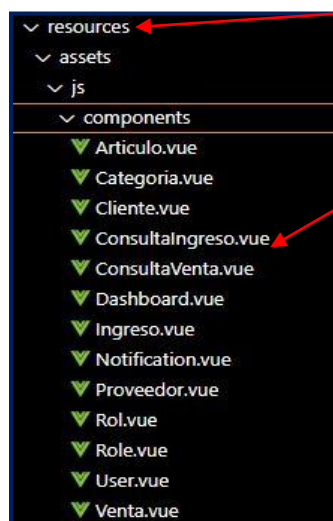
PS C:\xampp\htdocs\sisventas> php artisan make:controller CategoriaController --resource
```

Ilustración 16: Crear Controladores

26. CREAREMOS EL COMPONENTE Y ESTA UBICADO EN LA CARPETA `RESOURCES/ASSETS/JS/COMPONENTS/` `EXAMPLECOMPONENT.VUE` Y COPIAMOS `EXAMPLECOMPONENT.VUE` Y LO GUARDAMOS COMO `CATEGORIA.VUE` A SI MISMO SERÁ PARA LOS DEMÁS COMPONENTES A CREAR.

- Si VSC sugiere la instalación de una extensión, se instalan.
- Copiar todo el código de `ExampleComponent.vue` y pegarlo en `Categoria.vue`

26.1 Vista de donde esta ubicados los componetes.vue

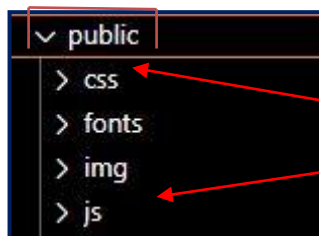


En la carpeta `resources`, `assets`, `js`, componentes encontraremos los compotentes.vue

Ilustración 17: Componentes

27. EN LA CARPETA PUBLIC ENCONTRAREMOS LOS CSS, FONTS, IMÁGENES Y LOS JS

27.1 Vista de la carpeta public.

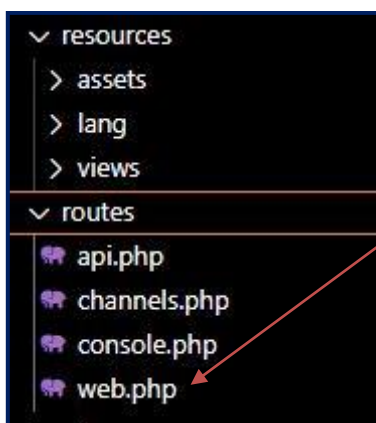


En la carpeta public encontraremos los css, fonts, img y js de nuestro sistema.

Ilustración 18: Vista de la carpeta public

28. EN LA CARPETA ROUTES ENCONTRAREMOS UN ARCHIVO LLAMADO WEB.PHP EN ESTE ARCHIVO ES DONDE VAN TODAS LAS RUTAS.

28.1 Vista de la carpeta routes.



```
Route::group(['middleware' => ['Almacenero']], function () {
    Route::get('/categoria', 'CategoriaController@index');
    Route::post('/categoria/registran', 'CategoriaController@store');
    Route::put('/categoria/actualizar', 'CategoriaController@update');
    Route::put('/categoria/desactivar', 'CategoriaController@desactivar');
    Route::put('/categoria/activar', 'CategoriaController@activar');
    Route::get('/categoria/selectCategoria', 'CategoriaController@selectCategoria');

    Route::get('/articulo', 'ArticuloController@index');
    Route::post('/articulo/registran', 'ArticuloController@store');
    Route::put('/articulo/actualizar', 'ArticuloController@update');
    Route::put('/articulo/desactivar', 'ArticuloController@desactivar');
    Route::put('/articulo/activar', 'ArticuloController@activar');
    Route::get('/articulo/buscarArticulo', 'ArticuloController@buscarArticulo');
    Route::get('/articulo/listarArticulo', 'ArticuloController@listarArticulo');
    Route::get('/articulo/listarPdf', 'ArticuloController@listarPdf->name('articulos_pdf');
    Route::get('/proveedor', 'ProveedorController@index');
}
```

Ilustración 19: Rutas

Ilustración 20: Archivo Web.php donde irán las rutas

29. CARPETA AUTH.

29.1 Vista de la carpeta auth y controladores.

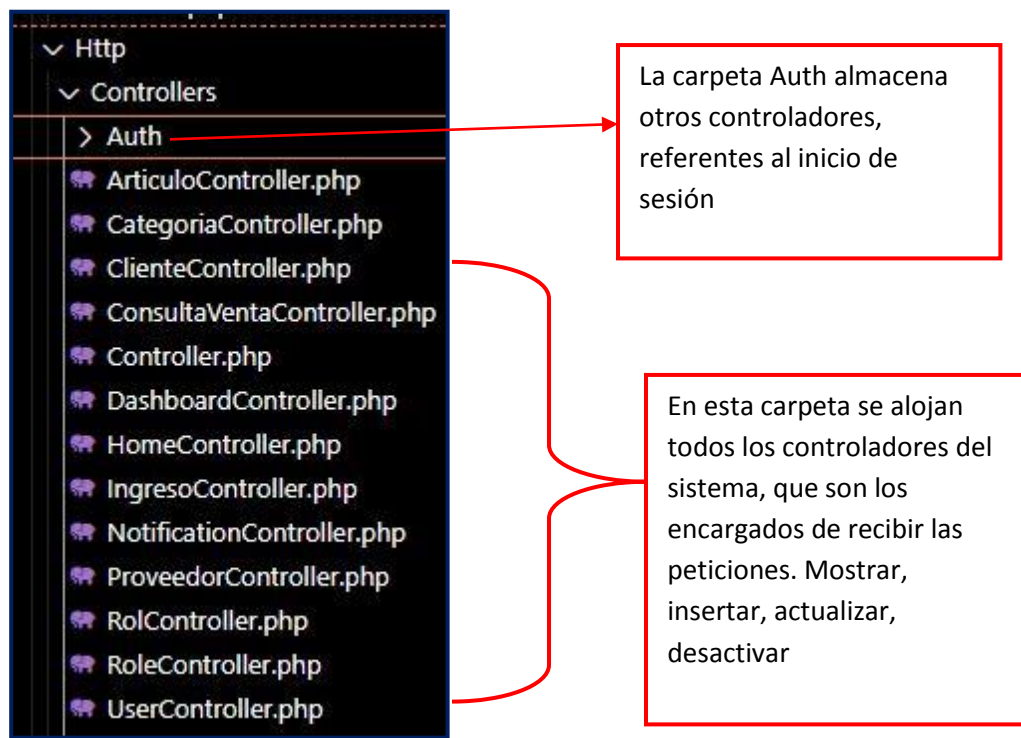


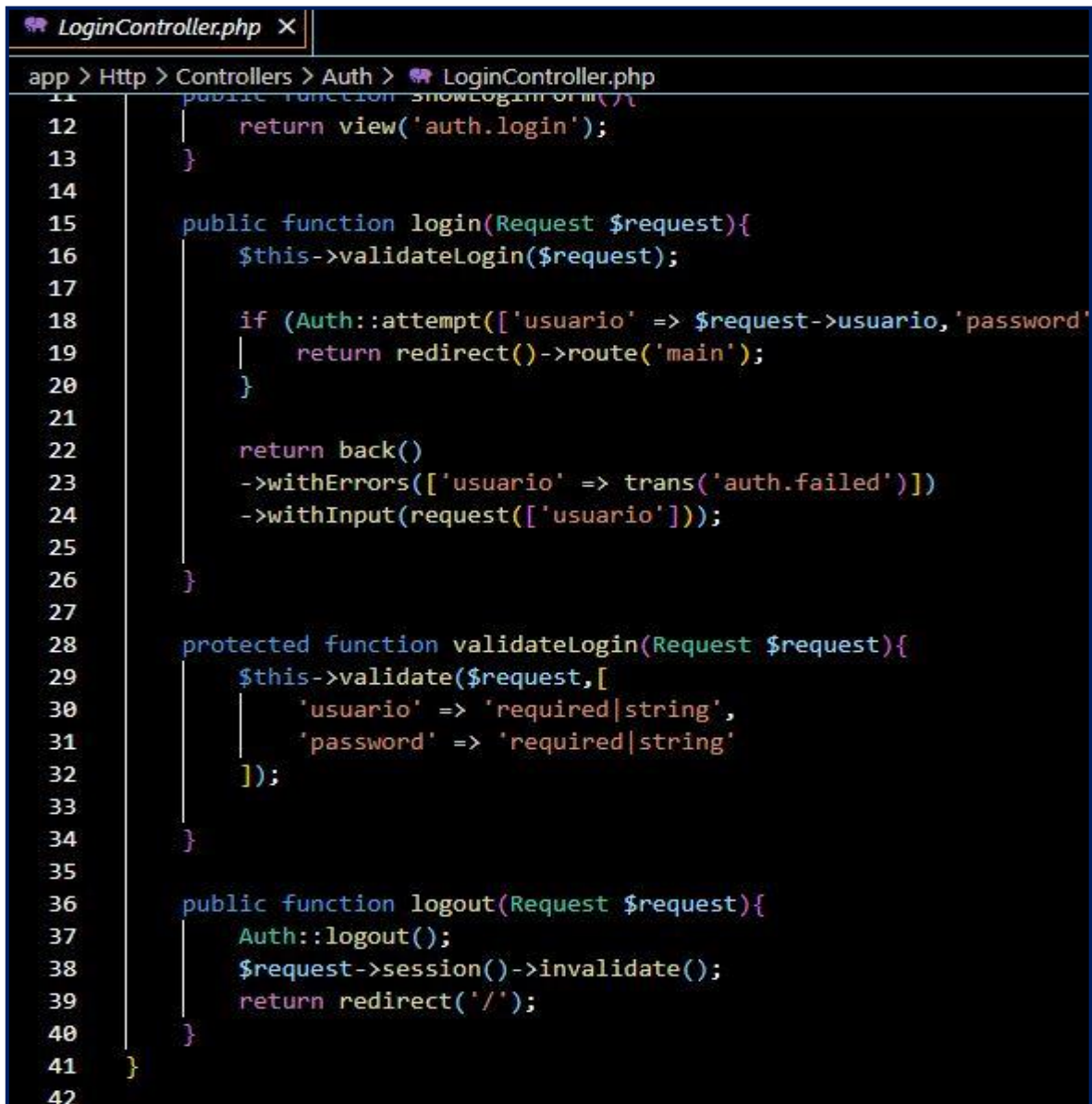
Ilustración 21: Carpeta de los controladores

30.1 Vista de directorio para logincontroller.php

app > Http > Controllers > Auth > LoginController.php

Ilustración 22: LoginController

En el directorio app/http/controllers/Auth encontramos LoginController.php aquí es donde se crea la vista del login para el sistema.



```

11 public function showLoginForm()
12 |     return view('auth.login');
13 | }
14
15 public function login(Request $request){
16 |     $this->validateLogin($request);
17
18 |     if (Auth::attempt(['usuario' => $request->usuario, 'password'
19 |         |     return redirect()->route('main');
20 |     }
21
22 |     return back()
23 |         ->withErrors(['usuario' => trans('auth.failed')])
24 |         ->withInput(request(['usuario']));
25 |
26 | }
27
28 protected function validateLogin(Request $request){
29 |     $this->validate($request,[
30 |         'usuario' => 'required|string',
31 |         'password' => 'required|string'
32 |     ]);
33 |
34 | }
35
36 public function logout(Request $request){
37 |     Auth::logout();
38 |     $request->session()->invalidate();
39 |     return redirect('/');
40 | }
41 }
42

```

Ilustración 23: Vista del Login