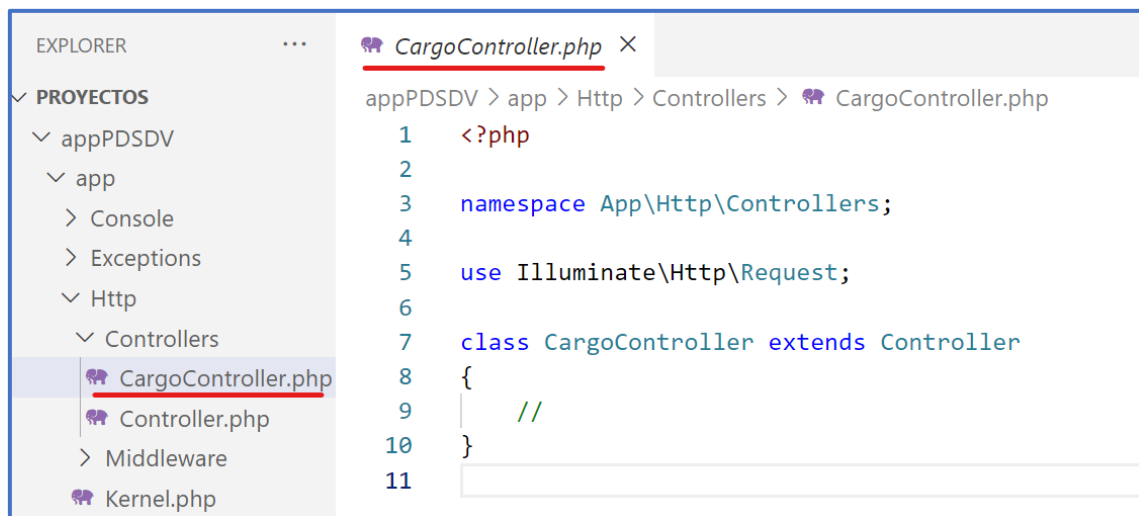


Creando controlador:

Nos dirigimos a nuestra terminal en GitBash y digitamos: **php artisan make:controller CargoController**

Verifiquemos que nos encontramos en la ruta raíz del proyecto:

```
clase@LAPTOP-CUD5QFUL MINGW64 /c:/xampp/htdocs/Desarrollo Web/Proyectos/appPDSDV
$ php artisan make:controller CargoController
Controller created successfully.
```



Modificando el Controlador:

Añadimos la función pública en el Controlador **CargoController**:

```
class CargoController extends Controller
{
    //Creamos una función pública:
    public function calcular(){
        return "Estamos calculando...";
    }
}
```

Modificamos en el archivo **cargo.php**

```
<?php

use App\Http\Controllers\CargoController;

Route::prefix('intranet')->middleware(['throttle:limite-acceso'])->group(function(){
    Route::get('cargo/index', function () {
        return 'Pagina de Cargo';
    });

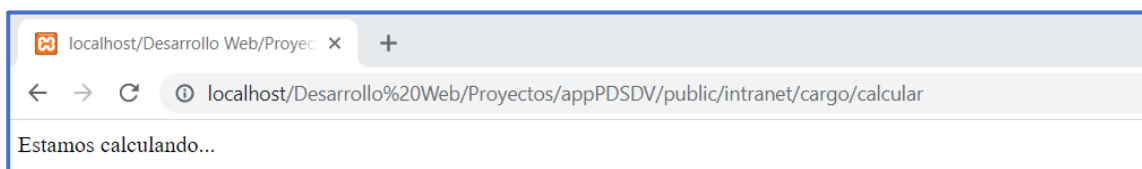
    Route::get('cargo/create', function () {
        return 'Creación de cargos';
    });

    //Agregando al Controller CargoController:
    Route::get('cargo/calcular', [CargoController::class, 'calcular'
]);

});

?>
```

El resultado en el navegador es:



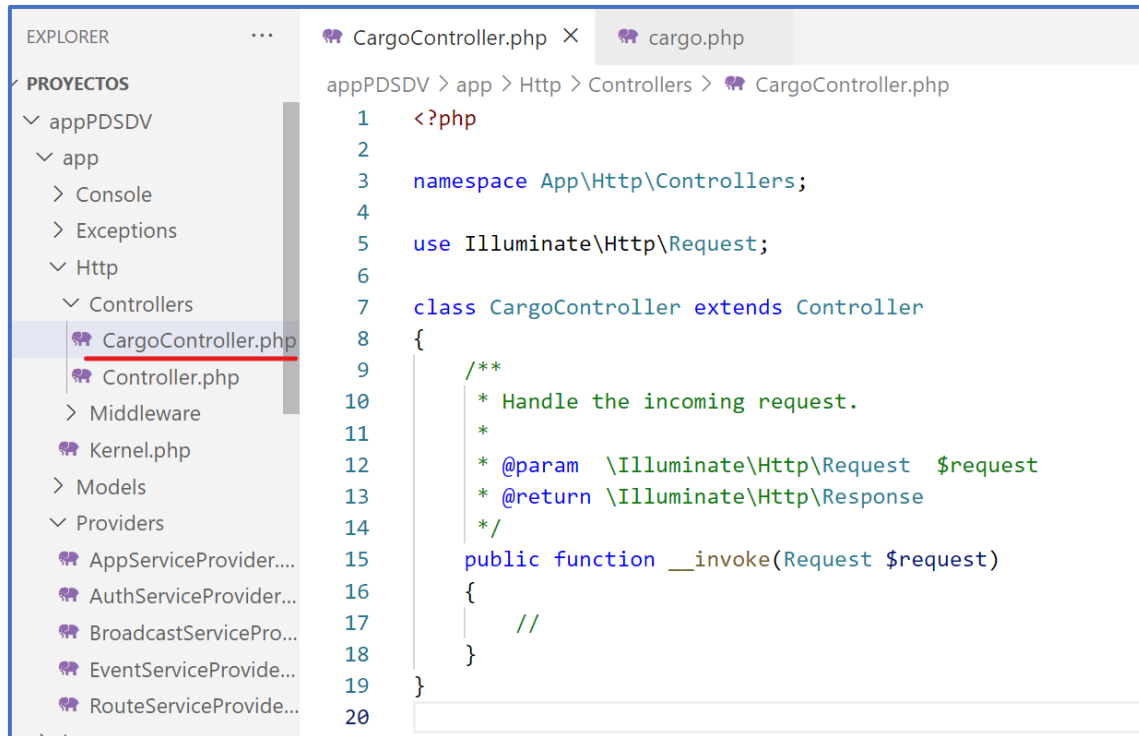
Creamos un controlador forzado:

Me permitirá llamar a la función invoke:

```
clase@LAPTOP-CUD5QFUL MINGW64 /c/xampp/htdocs/Desarrollo Web/Proyectos/appPDSDV
$ php artisan make:controller CargoController -i --force
Controller created successfully.

clase@LAPTOP-CUD5QFUL MINGW64 /c/xampp/htdocs/Desarrollo Web/Proyectos/appPDSDV
$
```

El controlador, tendrá la siguiente estructura:



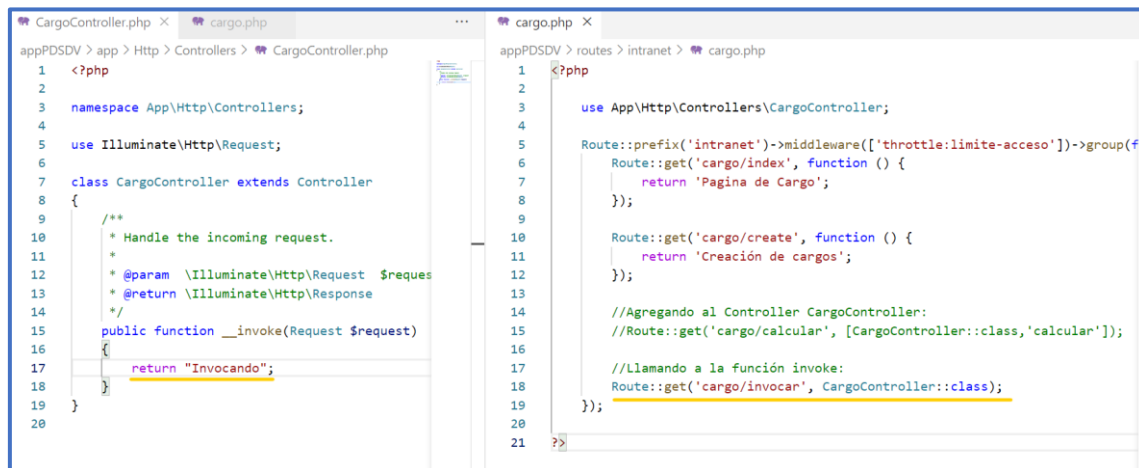
The screenshot shows the Explorer panel on the left with the project structure: **PROYECTOS** > **appPDSVDV** > **app** > **Http** > **Controllers**. The **CargoController.php** file is selected. The main editor shows the code for **CargoController.php**:

```

1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  class CargoController extends Controller
8  {
9      /**
10       * Handle the incoming request.
11       *
12       * @param  \Illuminate\Http\Request $request
13       * @return \Illuminate\Http\Response
14       */
15     public function __invoke(Request $request)
16     {
17         //
18     }
19 }
20

```

Modificamos nuestra codificación:



The screenshot shows two files in the IDE. The left file is **CargoController.php** with the following code:

```

1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  class CargoController extends Controller
8  {
9      /**
10       * Handle the incoming request.
11       *
12       * @param  \Illuminate\Http\Request $request
13       * @return \Illuminate\Http\Response
14       */
15     public function __invoke(Request $request)
16     {
17         return "Invocando";
18     }
19 }
20

```

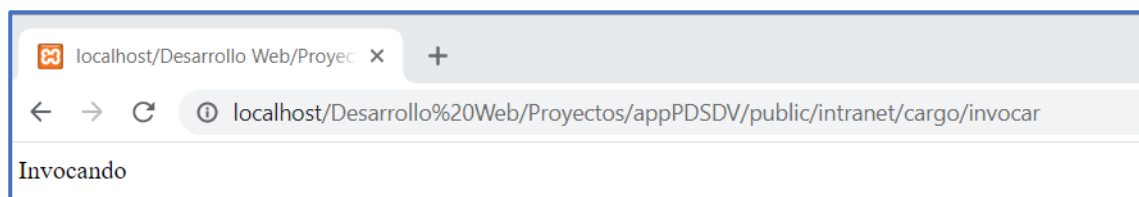
The right file is **cargo.php** with the following code:

```

1  <?php
2
3  use App\Http\Controllers\CargoController;
4
5  Route::prefix('intranet')->middleware(['throttle:limite-acceso'])->group(function () {
6      Route::get('cargo/index', function () {
7          return 'Pagina de Cargo';
8      });
9
10     Route::get('cargo/create', function () {
11         return 'Creación de cargos';
12     });
13
14     //Agregando al Controller CargoController:
15     //Route::get('cargo/calculan', [CargoController::class, 'calcular']);
16
17     //Llamando a la función invoke:
18     Route::get('cargo/invocar', CargoController::class);
19 });
20
21

```

El resultado en el navegador será:



The screenshot shows a web browser window with the address bar displaying: `localhost/Desarrollo Web/Proyectos/appPDSVDV/public/intranet/cargo/invocar`. The page content shows the text: **Invocando**.

Ahora forzaremos nuevamente para crear el Controlador **CargoController**

```
clase@LAPTOP-CUD5QFUL MINGW64 /c/xampp/htdocs/Desarrollo Web/Proyectos/appPDSDV
$ php artisan make:controller CargoController -r --force
Controller created successfully.

clase@LAPTOP-CUD5QFUL MINGW64 /c/xampp/htdocs/Desarrollo Web/Proyectos/appPDSDV
$
```

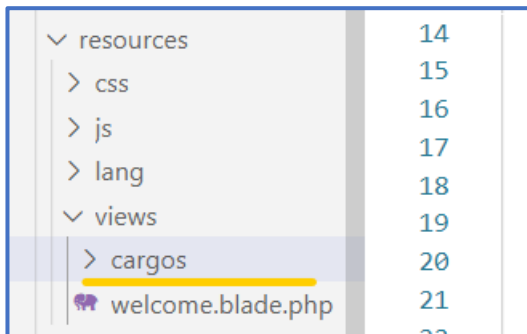
La sentencia anterior te permite crear el mismo controlador pero con los métodos del CRUD:



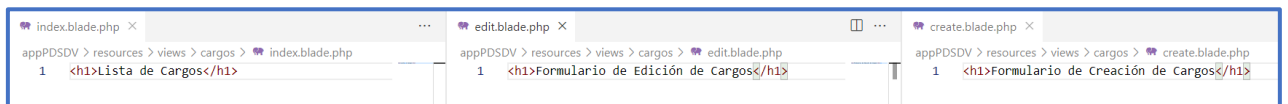
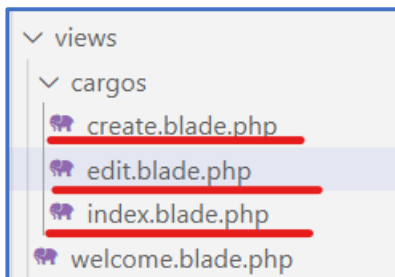
```
CargoController.php X
appPDSDV > app > Http > Controllers > CargoController.php
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  class CargoController extends Controller
8  {
9      /**
10       * Display a listing of the resource.
11       *
12       * @return \Illuminate\Http\Response
13       */
14     public function index()
15     {
16         //
17     }
18
19     /**
20      * Show the form for creating a new resource.
21      *
22      * @return \Illuminate\Http\Response
23      */
24     public function create()
25     {
26         //
27     }
28
29     /**
30      * Store a newly created resource in storage.
31      *
32      * @param \Illuminate\Http\Request $request
33      * @return \Illuminate\Http\Response
34      */
35     public function store(Request $request)
36     {
37         //
38     }
```

Trabajando con los métodos y las vistas

Crearemos la carpeta **cargos** dentro de **view/resources**



Dentro de esta carpeta **cargos**, creamos los archivos **create.blade.php** , **edit.blade.php**, **index.blade.php**



Ahora, trabajamos en el archivo **CargoController**:



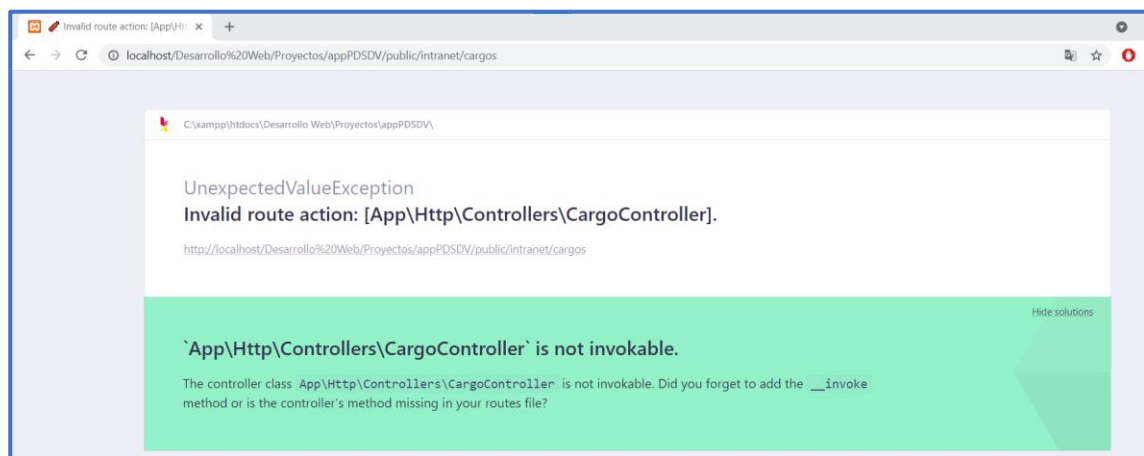
Asociamos ahora las rutas con el controlador:

```

CargoController.php cargo.php X
appPDSDV > routes > intranet > cargo.php
1  <?php
2
3      use App\Http\Controllers\CargoController;
4
5      Route::prefix('intranet')->middleware(['throttle:limite-acceso'])->gro
6      Route::get('cargo/index', function () {
7          return 'Pagina de Cargo';
8      });
9
10     Route::get('cargo/create', function () {
11         return 'Creación de cargos';
12     });
13
14     //Agregando al Controller CargoController:
15     //Route::get('cargo/calcular', [CargoController::class,'calcular'])
16
17     //Llamando a la función invoke:
18     Route::get('cargo/invocar', CargoController::class);
19
20     //Agregamos las rutas:
21     Route::get('cargos', [CargoController::class,'index']);
22     Route::get('cargos/create', [CargoController::class,'create']);
23     Route::get('cargos/edit/{id}', [CargoController::class,'edit']);
24 });
25
26

```

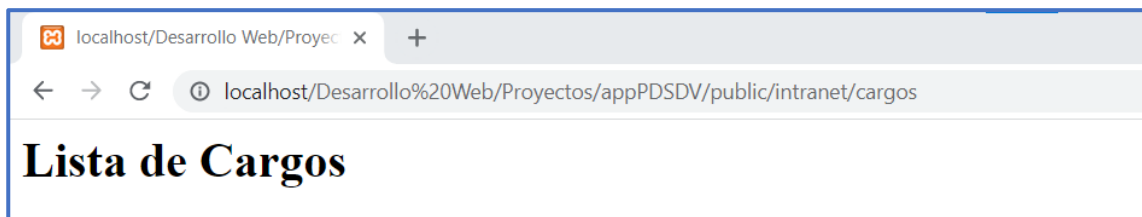
Si hacemos que nuestro navegador interprete veremos el siguiente error:



Ello se soluciona eliminando las rutas creadas anteriormente por el invoke():

```
return 'Creación de Cargos';  
});  
  
//Agregando al Controller CargoController:  
//Route::get('cargo/calcular', [CargoController::class, 'calcular']  
  
// //Llamando a la función invoke:  
// Route::get('cargo/invocar', CargoController::class);  
  
//Agregamos las rutas:  
Route::get('cargos', [CargoController::class, 'index']);  
Route::get('cargos/create', [CargoController::class, 'create']);  
Route::get('cargos/edit/{id}', [CargoController::class, 'edit']);  
});
```

Recargamos la página y ya podemos visualizar los resultados en el navegador:



Lo mismo realizaremos con create():

```
* @return \Illuminate\Http\Response  
*/  
public function create()  
{  
    return view('cargos/create');  
}
```



Finalmente codificamos para el método editar():

```
* @param int $id
* @return \Illuminate\Http\Response
*/
public function edit($id)
{
    return view('cargos/edit');
}
```



Podemos también retirar la validación de los 3 intentos.

