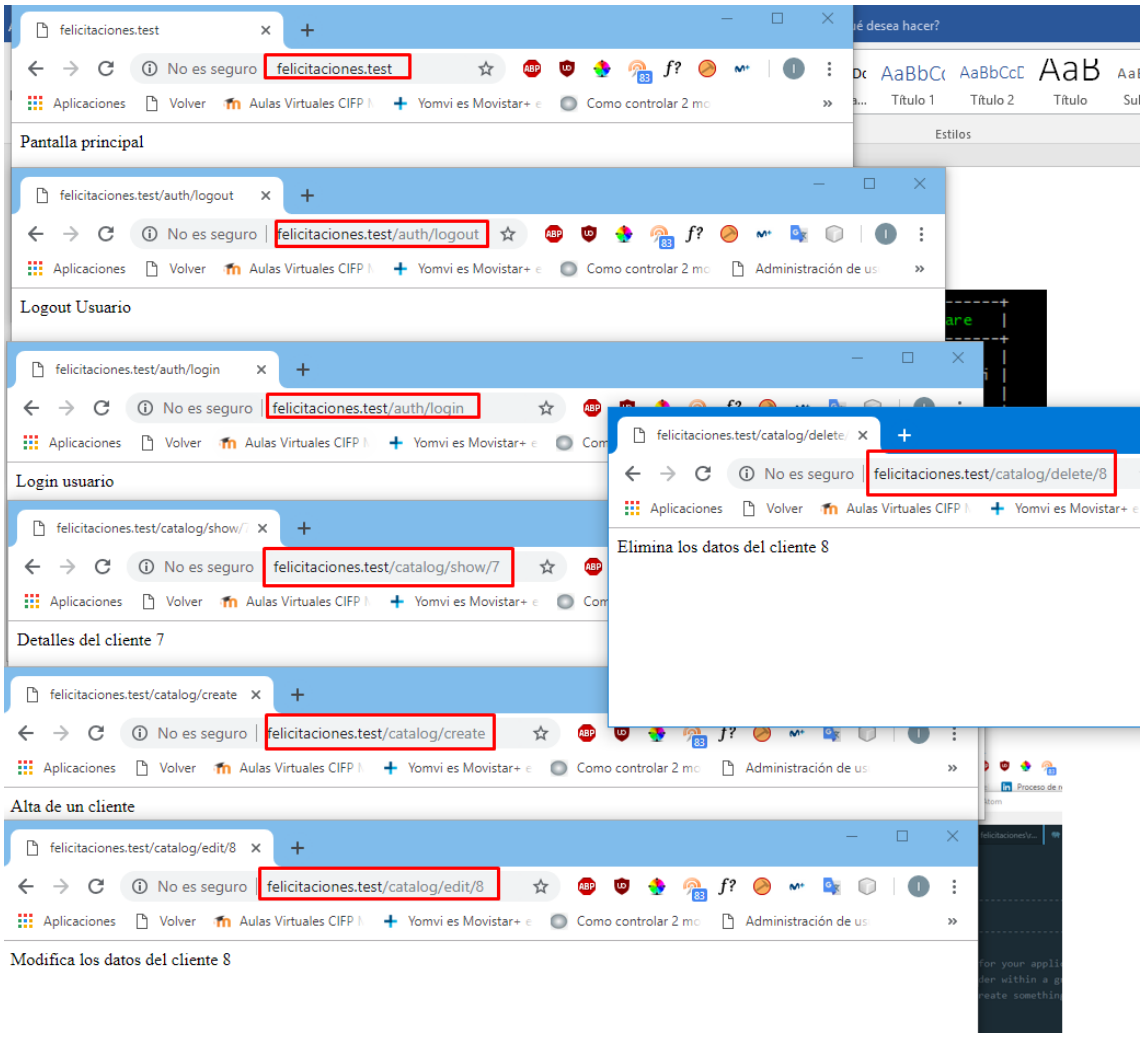
Ejercicio 1

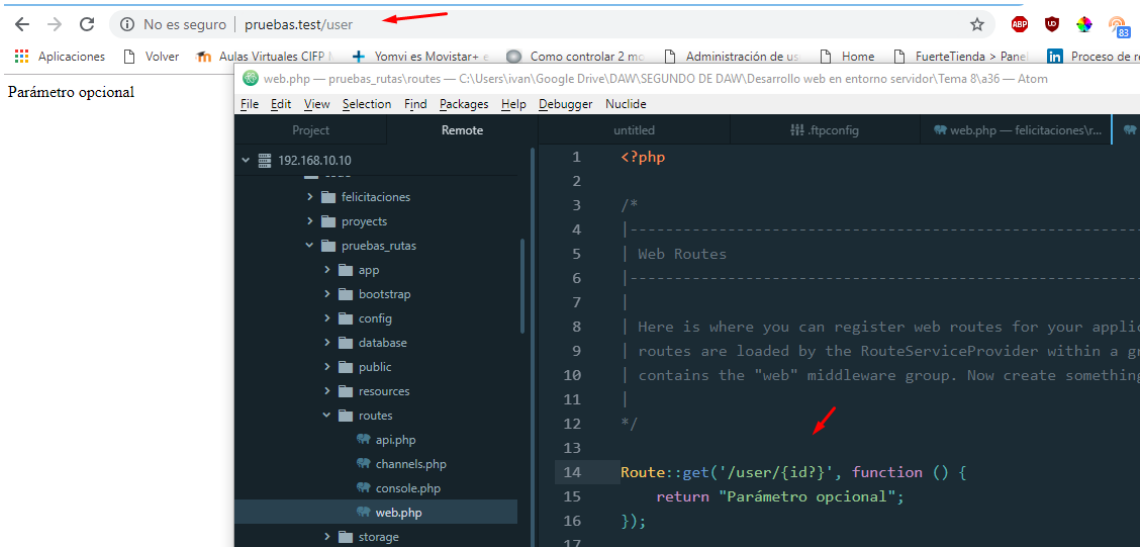
```
+--------+----------+-----------------------+------+---------+--------------+
| Domain | Method   | URI                   | Name | Action  | Middleware   |
+--------+----------+-----------------------+------+---------+--------------+
|        | GET|HEAD | /                     |      | Closure | web          |
|        | GET|HEAD | api/user              |      | Closure | api,auth:api |
|        | GET|HEAD | auth/login            |      | Closure | web          |
|        | GET|HEAD | auth/logout           |      | Closure | web          |
|        | GET|HEAD | catalog               |      | Closure | web          |
|        | GET|HEAD | catalog/create        |      | Closure | web          |
|        | GET|HEAD | catalog/delete/{id}   |      | Closure | web          |
|        | GET|HEAD | catalog/edit/{id}     |      | Closure | web          |
|        | GET|HEAD | catalog/show/{id}     |      | Closure | web          |
+--------+----------+-----------------------+------+---------+--------------+
vagrant@homestead:~/code/felicitaciones$
```

Comprobando rutas

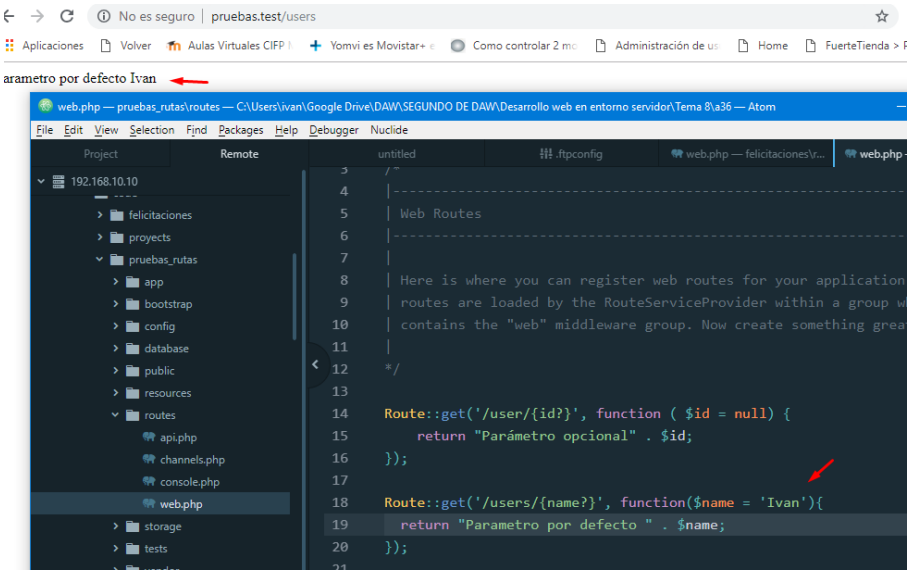


Ejercicio 2

1. Crea una ruta que tenga un parámetro que sea opcional y comprueba que funciona.

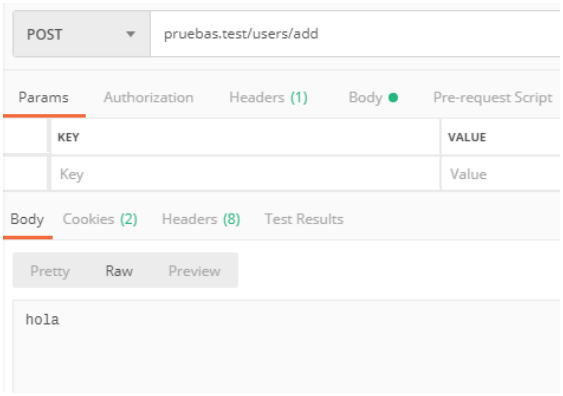


2. Crea una ruta que tenga un parámetro que sea opcional y tenga un valor por defecto en caso de que no se especifique.



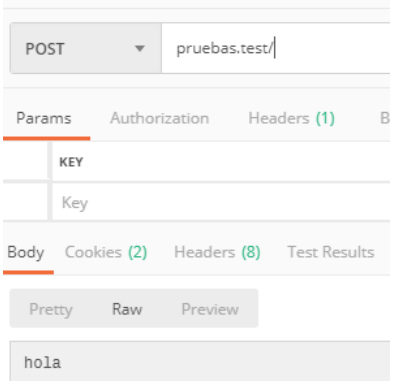
3. Crea una ruta que atienda por POST y compruébala con Postman. Si obtienes un error de tipo VerifyCsrfToken comenta la línea correspondiente en el fichero kernel.php (carpeta Http). Esto es un filtro para evitar ataques XSS (Cross-Site Scripting)

```
Route::post('/users/add', function(){
  return 'hola ';
});
```



4. Crea una ruta que atienda por GET y por POST (en un único método) y compruébalas.

```
Route::match(array('GET','POST'), '/', function(){
  return "hola";
});
```

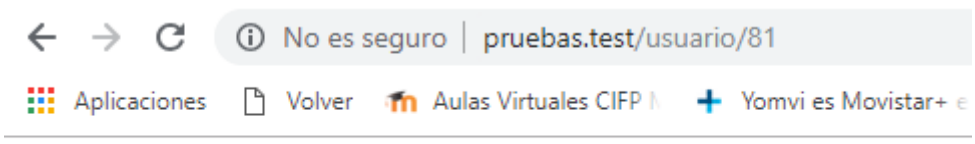


5. Crea una ruta que compruebe que un parámetro está formado sólo por números.

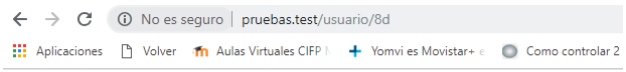
```
Route::get('usuario/{id}', function($id){

})->where('id', '[0-9]+');
```

Prueba con parámetros numéricos



Prueba con parámetros tipo texto.



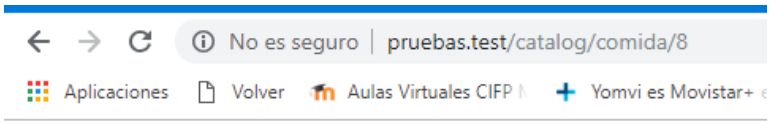
404

6. Crea una ruta con dos parámetros que compruebe que el primero está formado sólo por letras y el segundo sólo por números.

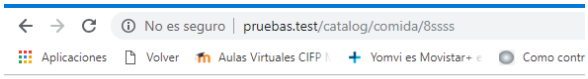
```
Route::get('catalog/{name}/{id}', function($id,$name){

})->where(array('name' => '[A-Za-z]+' , 'id' => '[0-9]+'));
```

Prueba con parámetros correctos



Prueba con un parámetro incorrecto



404

7. Ejecuta el siguiente comando y explica qué es lo que hace: `artisan make:controller MiControlador`

Crea la estructura para un controlador