**3. CONTROLADORES Y RUTAS**

Vamos a crear los controladores que en principio tendrá nuestra aplicación y haremos la rutas básicas para que funciones correctamente. Por si se nos ha olvidado, un controlador se encarga de recibir datos desde las vistas y hacer la lógica de procesamiento pidiendo datos a los modelos(SQL) y lo que necesite para devolver de nuevo a la vista los datos procesados. En resumen, controla y gestiona lógica de la aplicación.

Para crea un controlador vamos al CMD y en la ruta del proyecto🡪 **php artisan make:controller nombre controlador**. Para este proyecto necesitaremos los controladores:

1. Controlador usuario, encargado de procesar información de los usuarios como el registro perfiles…
2. Controlador post, encargado de gestionar la nuevas entradas, listar …
3. Controlador categorías.

Una vez creado crearemos una función de prueba en cada uno de ellos para ver que funcionan correctamente.

**PROBANDO CONTROLADORES:**

Crearemos una función de prueba que imprimirá en el navegador una frase para poder observar que la ruta del controlador funciona correctamente. En la función será necesario pasar como argumento el tipo request que será el encargado de almacenar toda la información que contenga la petición.

La función de prueba quedaría de la siguiente manera:

public function pruebas(Request $request){

        return "Accion de prueba de CategoryController";

}

Haremos lo mismo para cada uno de los controladores, y luego para usarlos deberemos crear una ruta, por lo tanto, iremos al archivo web.php proporcionado por Laravel y escribiremos la ruta para el uso del controlador.

Para laravel9 es necesario importar estos archivos al archivo web.php con la ruta completa:

use App\Http\Controllers\UserController;

Luego de igual manera que hicimos en los capítulos anteriores crearemos una ruta e invocaremos el método de pruebas que hemos declarado:

Route::get('/usuario/pruebas' , 'App\Http\Controllers\UserController@pruebas');

En usuario podremos crear los métodos de registro y login y más adelante crearemos su lógica, por ahora solo los declararemos y crearemos sus rutas.

**METODOS HTTP IMPORTANTES:**

1. GET : Conseguir datos o recursos --> petición si rellenar un formulario por lo general
2. POST: Guardar datos o recursos , más seguro que un get --> petición rellenando un formulario para guardar información, o el propio login
3. PUT : Actualizar datos o recursos
4. DELETE: Eliminar datos o recursos

Los métodos post obviamente no podrán ser accedidos mediante el navegador dado que se necesitare rellenar una serie de campos para poder usarlos. Como decíamos se usa sobre todo en formularios.

Route::post('/usuario/login' , 'App\Http\Controllers\UserController@login');

**HERRAMIENTA: CLIENTE RESTFUL**

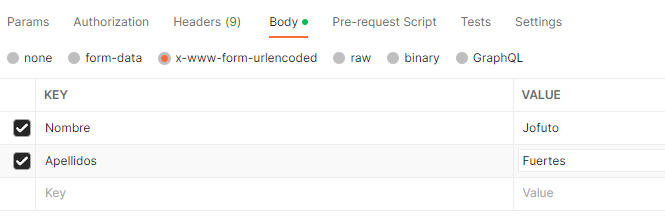
Postman aplicación que actúa como un cliente restfull y que utilizaremos para testear todas las rutas y métodos que vayamos desarrollando.

Laravel incluye una protección para las formularios que por ahora vamos a omitir, por ellos en el archivo kernel.php deberemos comentar la siguiente línea de la siguiente manera:

//\App\Http\Middleware\VerifyCsrfToken::class,

Una vez hecho esto, podremos mediante el cliente observar que mediante un método post nos devolverá el return que habíamos programado.

Por lo general los métodos post como habíamos dicho se usan para formularios, para poder gastar los datos que nos envíen utilizaremos request. En el cliente podemos ir a la pestaña body y seleccionar la pestaña x-www-… donde podremos rellenar parámetros como si se tratase de un formulario html.



Ahora para poder utilizar estos parámetros que hemos enviado mediante el método post deberemos utilizar request de la siguiente manera:

public function login(Request $request){

   $name = $request ->input('Nombre');

   $surname = $request ->input('Apellido');

   return "Accion de prueba de login UserController enviada por: $name $surname";

    }

**ORGANIZACIÓN DE RUTAS**

A medida que desarrollemos nuestro Backend deberemos tener una manera de saber y organizar cada una de la rutas que habremos ido desarrollando. Para ello Laravel nos proporciona un método para listar todas las rutas que hemos ido desarrollando:

