

Rutas en Laravel

Básicamente el trabajo con rutas consiste en lo siguiente:

- Laravel comprueba a qué ruta se hizo la petición.
- Si la ruta está definida en el sistema de rutas, se ejecuta el código vinculado a dicha ruta.

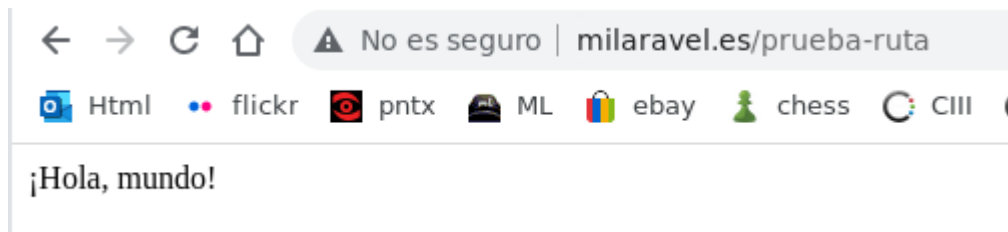
Es un sistema bastante intuitivo y fácil de manejar.

Laravel destina un directorio (*routes*) para ubicar todas las rutas de la aplicación. Por defecto tiene dos archivos de rutas: *web.php* y *api.php*. Inicialmente trabajaremos con *web.php*.

Dentro de *web.php* podemos definir una ruta de la siguiente forma:

```
Route::get('/prueba-ruta', function () {  
    return '¡Hola, mundo!';  
});
```

Siguiendo con nuestro ejemplo, para ver el funcionamiento de esta ruta tendremos que poner en el navegador <http://milaravel.es/prueba-ruta>



Además del método “get” de la clase Route, tendremos los siguientes métodos HTTP:

```
Route::post();  
Route::put();  
Route::delete();  
Route::any();
```

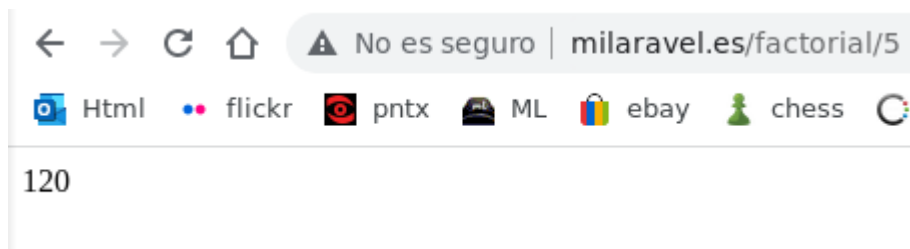
Por ahora sólo usaremos los métodos GET (Acceso normal a la web) y POST (Petición con parámetros para el servidor).

El primer parámetro del método get (‘/prueba-ruta’) especifica la ruta relativa con la que se ejecutará el método. El segundo parámetro sirve para definir la lógica de la petición. En este caso mandamos como parámetro una función que devuelve un texto plano.

Con el sistema de rutas también se pueden crear rutas más complejas que necesiten parámetros dinámicos, para lo que tendremos que hacer dos modificaciones en la definición

de la ruta. Primero tendremos que indicarle que en la ruta relativa irá un parámetro, poniéndolo entre llaves; después habrá que especificar que la función recibirá una variable de PHP.

```
Route::get('/factorial/{n}', function ($n) {  
    for ($i=1,$a=1;$i<=$n;$i++)  
        $a*=$i;  
  
    return $a;  
});
```



Referencias:

<https://styde.net/rutas-con-laravel/>