

# Controladores en Laravel 5

**Explicaciones y ejemplos sobre controladores en Laravel 5. Crear controladores, invocarlos desde las rutas.**

Aunque ya nos hemos referido anteriormente a los controladores solo fue muy de pasada. En este artículo comenzaremos a trabajar con ellos, conociéndolos un poco más a fondo.

Aunque suponemos que todos los lectores deben tener unas nociones básicas generales sobre el concepto de controlador, cabe aclarar que éstos son una de las piezas que, junto con los modelos y las vistas, forman parte del patrón MVC. En Laravel, como en cualquier otro de los frameworks PHP populares, son una importante parte de las aplicaciones. Su función es la de definir el código a ejecutar como comportamiento frente a una acción solicitada dentro de la aplicación.

Generalmente para poder desempeñar su labor se apoyan en los modelos y las vistas. El controlador sabe qué métodos del modelo debe invocar, ya sea para actualizar cierta información o para obtener ciertos datos, así como las vistas que deben presentar la información como respuesta al usuario, después de la realización de las acciones necesarias.

**Nota:** Para más información sobre MVC te referimos al artículo [Qué es MVC](#).

## Controladores en Laravel

**Los controladores están localizados en la carpeta `app/Http/Controllers`** y podemos organizarlos en subcarpetas si lo deseamos. Como otras clases de Laravel están dentro del sistema de autocarga de clases, por lo que estarán disponibles siempre que los necesitemos en la aplicación.

A continuación puedes encontrar el código de un controlador básico. Merece la pena pararse a analizarlo brevemente para comentar algunos detalles.

```
<?php

namespace App\Http\Controllers;

use App\Http\Controllers\Controller;

class ArticulosController extends Controller
{
    public function ver($id)
    {
        return view('articulos.ver', ['id' => $id]);
    }
}
```

Las dos primeras líneas son concernientes a los [espacios de nombres](#) donde estamos trabajando. Luego encontramos la propia clase que define el controlador, en este caso `ArticulosController`. Como cualquier clase, por convención, usamos la primera letra del nombre en mayúscula y además en Laravel tenemos por costumbre apellidar a las clases con el sufijo "Controller". También es una convención que te ayudará a identificar el código de los controladores y también a otras personas que puedan trabajar en el proyecto.

Dentro de la clase colocamos las acciones que deseemos. Cada acción la implementamos a partir de un método, al que podemos invocar desde el sistema de routing, como se verá a continuación.

**Nota:** Dentro del código de la acción, por el momento estamos colocando una llamada a una vista, que supuestamente mostraría un artículo. Generalmente los controladores acceden primero a los modelos para recuperar la información que se les debe pasar a las vistas para que representen. Aquí le estamos pasando como dato a la vista simplemente el id del artículo, lo que no resultaría de mucha utilidad, pero más adelante aprenderemos a acceder a la base de datos para recuperar esa información y poderla pasar completa hacia la vista.

Otro detalle que se puede observar es que los controladores son clases, de programación orientada a objetos, y como tales podríamos incluir en ellas cualquier miembro/s que consideremos oportuno para el funcionamiento interno, como métodos privados, propiedades, etc.

## Invocar un controlador desde el sistema de rutas

A los controladores los vamos a invocar normalmente desde el sistema de rutas, indicando el nombre del controlador y la acción (método) que debe ejecutarse para procesar una solicitud. En la siguiente línea de código registramos una ruta que llamaría a la acción "ver" sobre el controlador "ArticulosController".

```
Route::get('articulos/{id}', 'ArticulosController@ver');
```

Las acciones pueden recibir parámetros, tal como se explicó cuando aprendimos a crear nuestras rutas en Laravel. En esa ruta estaríamos definiendo que se debe recibir el parámetro {id}. El valor de ese parámetro será pasado al método o acción ver() de "ArticulosController". Obviamente, podemos pasar tantos parámetros como sea necesario y los recibiremos en la acción del controlador en el mismo orden como fueron definidos en el patrón de la URI registrada.

## Generar los controladores automáticamente con artisan

Crear desde cero un controlador es una tarea repetitiva dentro de Laravel, por lo que existen atajos. El ya conocido comando "artisan" nos ofrece una utilidad para crear una nueva clase controlador de una manera automática. Para ejecutarlo lanzamos en la consola este comando.

```
phpartisanmake:controllerCategoriasController
```

Dentro de la carpeta del proyecto, invocas a artisan. La operación solicitada es `make:controller` y luego le indicamos el nombre del controlador a crear, en este caso `"CategoriasController"`.

**Nota:** Si en cualquier momento queremos ver la lista de utilidades que nos facilita artisan, escribimos este comando en la consola:

```
phpartisan
```

Pero ojo, para que funcione artisan tienes que estar en la home del proyecto, donde verás que se encuentra un archivo llamado `"artisan"`.

Ahora verás, en la carpeta de los controladores (`"app/Http/Controllers"`), que ha aparecido el nuevo archivo del controlador, generado automáticamente. Te darás cuenta que tiene ya una serie de acciones definidas que son las típicas cuando quieres crear un recurso `"RESTful"`. De momento no las usaremos pero más adelante verás que es un atajo potente, ya que te dan declarados todos los métodos que necesitarías para realizar las operaciones típicas con un recurso en un API Rest.