

Práctica UD-5 Sesión-5

Objetivos	<ul style="list-style-type: none">• Aprender el uso de requests, responses, cookies y sesiones para el desarrollo de aplicaciones webs con Laravel.
Instrucciones de entrega	<ul style="list-style-type: none">• Subid el proyecto o ficheros con el código que resuelve los ejercicios a Edmodo (no es imprescindible haber resuelto todos para entregar).

- Abre una línea de comandos desde **VS Code** y ve a la ruta donde está el directorio **htdocs** de Xampp (**c:\xampp\htdocs**) después crea un nuevo proyecto con nombre **PHPPracticaUD5Sesion5**:

```
C:\xampp\htdocs>laravel new PHPPracticaUD5Sesion5
```

- Carga el proyecto desde **VS Code** simplemente abriendo la carpeta.

Petición HTTP (Request)

- Documentación oficial
 - <https://laravel.com/docs/8.x/requests>
- En Laravel a través de la clase **Illuminate\Http\Request** podemos interactuar con la solicitud HTTP actual que recibe la aplicación por ejemplo para recuperar los parámetros de entrada, las cookies y los archivos que se enviaron con la solicitud.

Acceso desde un método del controlador

- Crea un nuevo controlador con nombre **TestRequestController**.
 - **C:\xampp\htdocs\PHPPracticaUD5Sesion5> php artisan make:controller TestRequestController**
- Añade el siguiente código:

```
public function index(Request $request){  
  
}
```

Acceso desde una ruta

- Edita el fichero de rutas **web.php** con el siguiente contenido:

```
use Illuminate\Http\Request;
use App\Http\Controllers\TestRequestController;
...

Route::get('/testRequestEnRuta', function (Request $request) {

});

Route::get('/testRequestEnControlador', [TestRequestController::class,
'index']);
```

- Comprueba que las rutas se ha registrado bien:
 - `php artisan route:list`
- Ahora vamos a acceder a información de la request, para ello edita el contenido del método `index` del controlador para añadir:

```
public function index(Request $request) {
    // Recuperar información de la petición HTTP
    // Recuperar parámetro del request payload (tanto post como
    querystring)
    $input = 'no está presente';
    if ($request->has('nombre')){
        $input = $request->input('nombre');
    }
    // Recuperar parámetro sólo de QueryString
    $query = $request->query('apellidos');
    // Recuperar el Path
    $uri = $request->path();
    // Recuperar la URL
    $urlWithQueryString = $request->fullUrl();
    // Recuperar el método
    $method = $request->method();
    // Recuperar cabeceras
    $value = $request->header('User-Agent');
    // Recuperar la ip
    $ipAddress = $request->ip();
```

```

    $array = [$input, $query, $uri, $urlWithQueryString, $method, $value,
$ipAddress];

    return join('<br>', $array);
}

```

- Arranca el servidor de pruebas:
 - `php artisan serve`
- Prueba el resultado navegando a la URL:
 - <http://127.0.0.1:8000/testRequestEnControlador>
 - <http://127.0.0.1:8000/testRequestEnControlador?nombre=juan>
 - 127.0.0.1:8000/testRequestEnControlador?nombre=juan&apellidos=lopez

Respuesta HTTP (Response)

- Documentación oficial
 - <https://laravel.com/docs/8.x/responses>
- Por lo general, no solo devolverá cadenas o arrays simples en las funciones de las rutas o en los controladores sino que podemos devolver instancias de ***Illuminate\Http\Response*** o bien vistas completas (*views*).
- Devolver una instancia de ***Illuminate\Http\Response*** permite personalizar el código de estado HTTP y las cabeceras de HTTP.
- Añade una nueva entrada en el fichero de rutas con lo siguiente código:

```

Route::get('/home', function () {
    return response('Hello World', 200)
        ->header('Content-Type', 'text/plain');
});

```

Sesión

- Documentación oficial
 - <https://laravel.com/docs/8.x/session>
- Hay dos formas principales de trabajar con datos de sesión en Laravel:
 - El helper global de sesión: ***session***
 - Un método de la instancia de ***Request***
- Crea un nuevo controlador con nombre ***TestSessionController*** y añade el siguiente código:

```

public function escribirDatosDESesion(Request $request){

    // Via a request instance...
    $request->session()->put('nombre', 'Enrique');

    // Via the global "session" helper...
    session(['apellidos' => 'García']);

    return 'Datos escritos en sesión';
}

```

- Si la variable de sesión es un array y se quiere simplemente añadir un nuevo elemento se puede hacer directamente con el método **push**:

```

$products = array(1,2,3,4);
$request->session()->put('products', $products);
$request->session()->push('products', 5);

```

- Para recuperar datos de la sesión se utiliza el método **put** de la instancia Request, añade un nuevo método al controlador:

```

public function leerDatosDeSesion(Request $request){

    // Via a request instance...
    $nombre = $request->session()->get('nombre', 'uno por defecto');
    $apellidos = $request->session()->get('apellidos');

    // Via the global "session" helper...
    $apellidosDesdeHelper = session('apellidos');

    $array = [$nombre, $apellidos, $apellidosDesdeHelper];
    return join('<br>', $array);
}

```

- Edita el fichero de rutas **web.php** con el siguiente contenido:

```

Route::get('/testEscribirDatosDeSesion', [TestSessionController::class,
'escribirDatosDeSesion']);
Route::get('/testLeerDatosDeSesion', [TestSessionController::class,
'leerDatosDeSesion']);

```

- Prueba el resultado navegando a la URL:
 - <http://127.0.0.1:8000/testEscribirDatosDeSesion>
 - <http://127.0.0.1:8000/testLeerDatosDeSesion>

Cookies

- Documentación oficial
 - <https://laravel.com/docs/8.x/requests#cookies>
 - <https://laravel.com/docs/8.x/responses#attaching-cookies-to-responses>
- Todas las cookies creadas por Laravel están encriptadas y firmadas con un código de autenticación, lo que significa que se considerarán inválidas si el cliente las ha cambiado.
- Para crear una cookie lo podemos hacer de la siguiente manera, añade una nueva ruta en el fichero de rutas:

```
Route::get('/testCrearCookie', function () {
    $minutes = 2;
    return response('Creando una cookie')->withCookie(cookie('cookie_dws', 'en
clase de Laravel', $minutes));
});
```

- Para recuperar un valor de una cookie de la solicitud, se usa el método **cookie** de la clase **Illuminate\Http\Request** que hemos visto antes, para ello añade una nueva ruta con el siguiente contenido:

```
Route::get('/testLeerCookie', function (Request $request) {
    $value = $request->cookie('cookie_dws');
    return response("La cookie contiene: $value");
});
```

- Prueba el resultado navegando a la URL:
 - <http://127.0.0.1:8000/testCrearCookie>
 - <http://127.0.0.1:8000/testLeerCookie>

Ejercicios

Página de MicroBlogging

- Este ejercicio consiste en crear una página para la publicación de comentarios por parte de cualquier usuario. Constará de un formulario en su parte superior y a continuación los diferentes comentarios que los usuarios hayan ido añadiendo.
- Para su realización utilice el mismo proyecto **PHPPracticaUD5Sesion5** pero crea una carpeta específica microblogging para el controlador, la vista y el modelo. El resultado deberá incluir un formulario como el siguiente:

Este campo es obligatorio.

Este campo es obligatorio.

Este campo es obligatorio.

Enviar datos

- La validación deberá ser hecha en el servidor.
- Debajo del formulario crea una tabla dinámica con una sola columna donde tengamos por cada fila el contenido de la forma: Fecha, Nombre, Correo, Título y Comentario, por ejemplo:

09/04/2019 Rafeal Acosta micorreo@gmail.com

Lorem ipsum dolor sit amet, consectetur adipiscing elit

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed id magna sit amet nibh pharetra volutpat. Aliquam erat volutpat. Quisque id vulputate neque. Pellentesque vitae ultricies tortor. Mauris maximus quis diam id ultricies. In hac habitasse platea dictumst. Donec laoreet sapien sit amet euismod pharetra. Ut mi lorem, commodo sed ornare sit amet, tempor vitae ligula.

09/04/2019 Rafael Acosta micorreo@gmail.com

Suspendisse lorem justo, rhoncus faucibus efficitur sit amet

Sed molestie est sed magna condimentum pellentesque. Ut varius erat vitae facilisis tincidunt. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Cras ornare aliquam sem, et ullamcorper diam efficitur sit amet. Duis tincidunt ipsum sem, sed porttitor sem malesuada id. Aliquam efficitur pulvinar ipsum sed varius. Sed consequat erat dapibus cursus commodo. Donec nunc massa, imperdiet ac aliquet eu, blandit sit amet ante. Proin lorem ipsum, facilisis ac condimentum id, ultricies sed dui.

- Los comentarios no se deben perder en los sucesivos envíos (se guardarán en la sesión) y se mostrarán a todos los usuarios de la misma sesión.

