

# LARAVEL

CLASE 09



# SESSION

Uso de session en Laravel



**HTTP es en sí mismo “stateless”, es decir, sin estado.**

**¿Cómo mantengo información en una secuencia de requests y responses?**

## Formas de implementar Session:

- **Archivo** (*en storage/framework/sessions*)
- **Cookies**
- **Base de datos relacional** → `php artisan session:table`
- **Memcached / Redis**
- **(Array: solo para pruebas)**

## Veamos cómo obtener estos datos en el controlador:

```
public function listado(Request $request, $id)
{
    $valor = $request->session()->get('nombre_clave');
    //
}
```



# LOCALIZATION

Aplicaciones multi-idioma en Laravel



Laravel nos provee de un mecanismo muy sencillo para guardar las traducciones. Las palabras o frases van en arrays dentro de archivos, agrupadas según el contexto o dominio de cada una.

```
/resources  
  /lang  
    /en  
      messages.php  
    /es  
      messages.php
```

Luego en blade...

```
{{ __('messages.welcome') }}
```

```
@lang('messages.welcome')
```



# APIs JSON

The background features abstract, overlapping geometric shapes in various colors including green, blue, orange, red, and purple, creating a modern, layered effect on a light gray background.

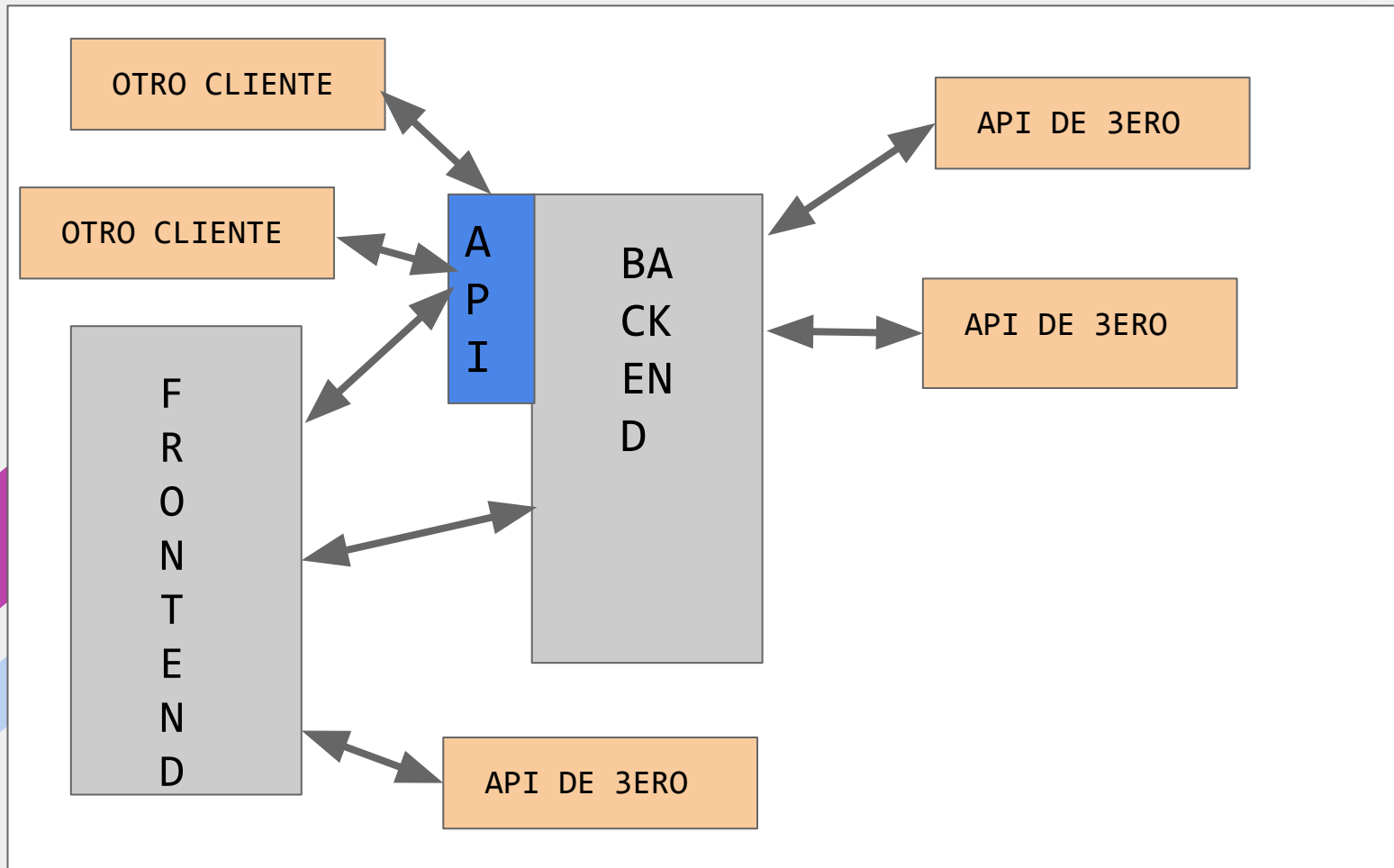
***Una API puede considerarse como un conjunto de funciones y rutinas diseñadas para ser utilizadas por otro software, lo que ofrece una gran ventaja a la hora del desarrollo de aplicaciones multiplataforma pues de esta forma podemos concentrar toda la lógica del negocio en un solo lugar accesible desde muchas otras aplicaciones o programas.***



**¿Y QUÉ TIENE QUE VER ESTO**





**CON MIS APLICACIONES EN LARAVEL?**




# URLs RESTful





¿Qué es un buen diseño?  
¿Qué implica buen diseño en una  
aplicación?  
¿Qué implica buen diseño en el código?



Mejor comunicación  
(respetando estandar)



Menos documentación



Más felicidad



## Estandar RESTful

- Organizamos URLs por recurso
- Cada recurso se identifica con un sustantivo (no verbos)  
Ejemplo: /peliculas, /actores, /carreras
- Los sub recursos se identifican dentro del recurso principal. Ejemplo /carreras/{id}/tramos
- Se recomienda usar siempre en plural (por más que obtenga sólo uno)
- Las acciones se identifican con los métodos HTTP (GET, POST, PUT, PATCH, DELETE)



## ¿Qué NO hacer en las URLs?

- Indicar acción en la URL:  
`/verUsuario`, `/modificarUsuario`
- Cambiar cardinalidad del sustantivo:  
`/usuario/{id}`, `/usuarios`

## Ejemplo: ¿cómo son las URLs del CRUD para películas?

<b>Nueva película</b>	<b>POST /peliculas</b>
<b>Modificar toda la película</b>	<b>PUT /peliculas/{id}</b>
<b>Modificar parte de la película</b>	<b>PATCH /peliculas/{id}</b>
<b>Listar todas las películas</b>	<b>GET /peliculas</b>
<b>Obtener una película</b>	<b>GET /peliculas/{id}</b>
<b>Eliminar una película</b>	<b>DELETE /peliculas/{id}</b>

Ejemplo: ¿qué pasa si tengo que hacer una acción  
que no está en los métodos HTTP?

Ejemplo: marcar una serie con una estrella

GET /series/{id}

Obtengo detalles de una serie

PUT /series/{id}/star

Agrego estrella

DELETE /series/{id}/star

Saco estrella

**¡Importante!**

**Si bien intentamos seguir el estándar,  
podemos apoyarnos en comentarios y  
documentación del código si por algún  
motivo vemos más claro no seguirlo  
(no muere un gatito si una URL no es  
RESTful)**

**ES MOMENTO DE  
PRACTICAR !**

### **RESTful**

- Corregir las rutas de la aplicación actual de películas.
- Diseñar las rutas del proyecto grupal.

The background features abstract, overlapping geometric shapes in various colors including light blue, green, cyan, magenta, orange, and red. These shapes are arranged in a way that creates a sense of depth and movement, with some shapes appearing to be layered on top of others. The overall composition is modern and vibrant.

**¡ HASTA LA  
PROXIMA !**