

Bienvenidos

Curso de Laravel + despliegue en Ubuntu

Daniel Cruz

¿Qué voy a aprender?

- En este curso aprenderás a desarrollar una API Rest en el Framework más utilizado en PHP.
- Sabrás hacer un despliegue de tu API en un hosting.
- Sabrás la configuración básica para probar tu API en Desarrollo.



¿Qué necesito?

- Un equipo de cómputo con Ubuntu.
- Conocimientos básicos en comandos de Linux.
- Conocimientos básicos en lógica de programación.
- Conocimientos básicos en PHP.

```
realpath($_SERVER['DOCUMENT_ROOT'])  
trim(preg_replace('/\\\\\\\\/', '/', $image_src), '/');  
$_SESSION['_CAPTCHA']['config'] = serialize($captcha_config);  
array(  
    'code' => $captcha_config['code'],  
    'image_src' => $image_src  
);  
  
if ( !file_exists('hex2rgb') ) {  
    hex2rgb($hex_str, $return_string = false, $separator = ',') {  
        $str = preg_replace("/[^0-9A-Fa-f]/", '', $hex_str); // Gets a proper  
        $array = array();  
        if ( strlen($hex_str) == 6 ) {  
            $color_val = hexdec($hex_str);  
            $array['r'] = 0xFF & ($color_val >> 0x10);  
            $array['g'] = 0xFF & ($color_val >> 0x8);  
            $array['b'] = 0xFF & $color_val;  
        }  
        if ( strlen($hex_str) == 3 ) {  
            $array['r'] = hexdec(str_repeat(substr($hex_str, 0, 1), 2));  
            $array['g'] = hexdec(str_repeat(substr($hex_str, 1, 1), 2));  
            $array['b'] = hexdec(str_repeat(substr($hex_str, 2, 1), 2));  
        }  
        return $array;  
    }  
    return $return_string ? implode($separator, $rgb)  
    : $array;  
}
```

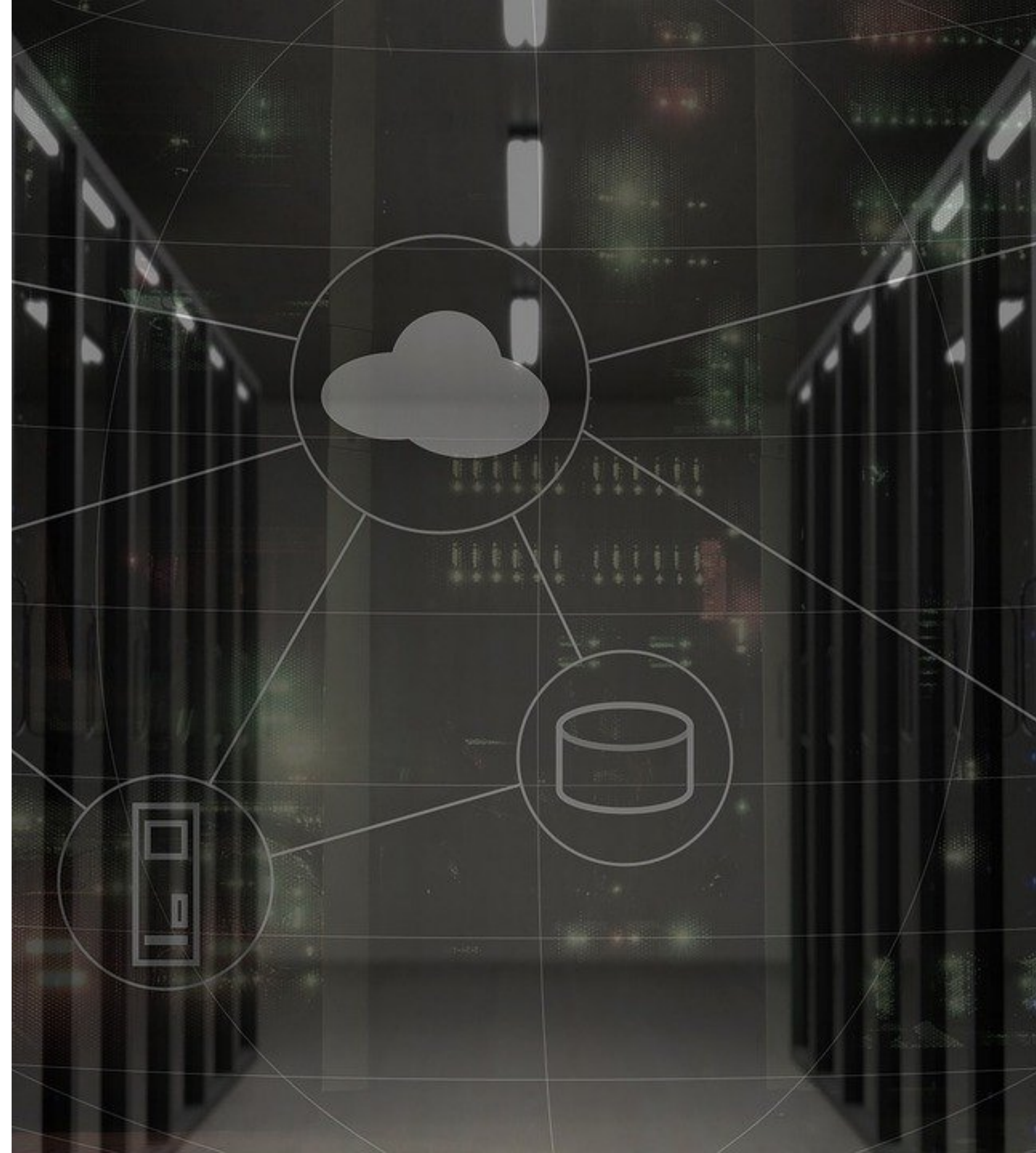

Apache

- Es un servidor HTTP de código abierto.
- Fue lanzado en 1995 y desde abril de 1996 fue el servidor web más popular hasta ahora.
- Apache se puede configurar para alojar varias aplicaciones en diversos lenguajes de programación.



HTTP

- Protocolo de Transferencia de Hipertexto. es un protocolo de la capa de aplicación para la transmisión de documentos hipermedia, como HTML.
- El protocolo cuenta con verbos que permiten hacer operaciones que veremos dentro de una API Rest.



Verbos HTTP

- POST: Regularmente se ocupa para enviar información de un registro. (CREATE)
- GET: Obtiene información de los registros almacenados (READ)
- PUT/PATH: Actualiza la información de los registros (UPDATE)
- DELETE: Eliminar un registro.

```
def property(
    self,
    name: str,
    value: Any,
    *,
    dirname: str = None,
    in_the_viewport: bool = False,
    y: int = None,
    context_selected_objects: List[Object] = None,
    port_selection: bool = False,
    setting == False:
    for i in bpy.context.scene.objects]

def l(action='DESELECT')

def list:

def H':
    path.join(folder_path, "{}.obj".format(item.name))
    scene.obj(filepath=file_path, use_selection=True,
               axis_forward=self.axis_forward_setting,
               axis_up=self.axis_up_setting,
               use_animation=self.use_animation_setting,
               use_mesh_modifiers=self.use_mesh_modifiers_setting,
               use_edges=self.use_edges_setting,
               use_smooth_groups=self.use_smooth_groups_setting,
               use_smooth_groups_bitflags=self.use_smooth_groups_bitflags_setting,
               use_normals=self.use_normals_setting,
               use_uv=self.use_uv_setting,
               use_materials=self.use_materials_setting,
```

Estados de respuesta HTTP

- 1. Respuestas informativas (100–199),
 - 2. Respuestas satisfactorias (200–299),
 - 3. Redirecciones (300–399),
 - 4. Errores de los clientes (400–499),
 - 5. Errores de los servidores (500–599).
-
- <https://developer.mozilla.org/es/docs/Web/HTTP/Status>



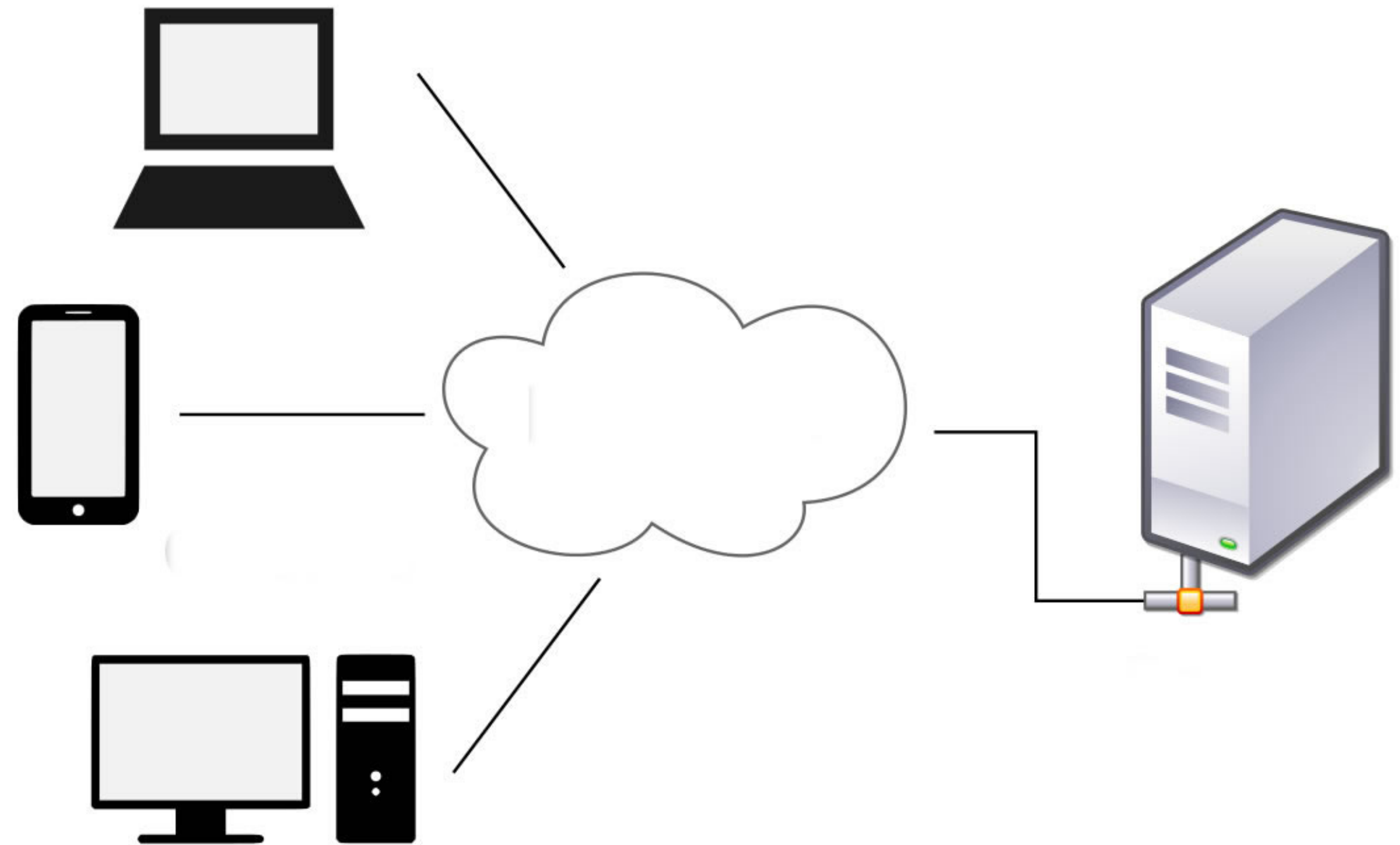
Arquitectura

- Las mayoría de las aplicaciones actuales ocupan una arquitectura dividida en al menos estos tres grupos:
 - Frontend
 - Backend
 - Database



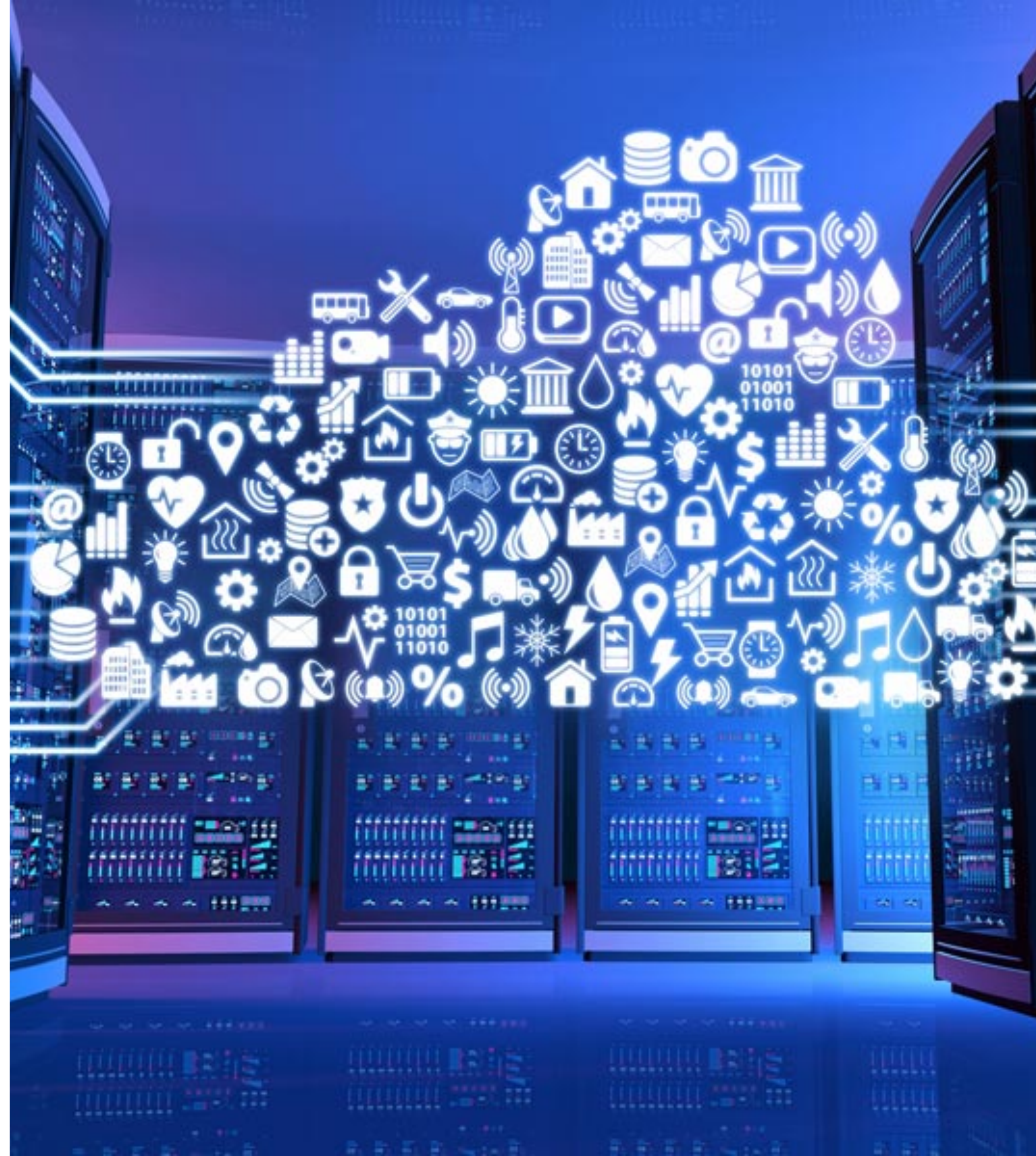
Arquitectura

- La arquitectura de cliente servidor se puede observar en todos los sistemas web aunque en sí mismo existen diferentes escalamientos y arquitecturas que pueden ser útiles para ciertos proyectos.
- Durante nuestro proyecto nos enfocaremos en el trabajo del lado del servidor por lo que utilizaremos Laravel como servidor aunque también puede usarse para renderizar contenido de nuestro sitio.



Ventajas del render en el servidor

- Mejora la indexación de los buscadores para nuestro sitio.
- Mejora el tiempo de la primera carga.



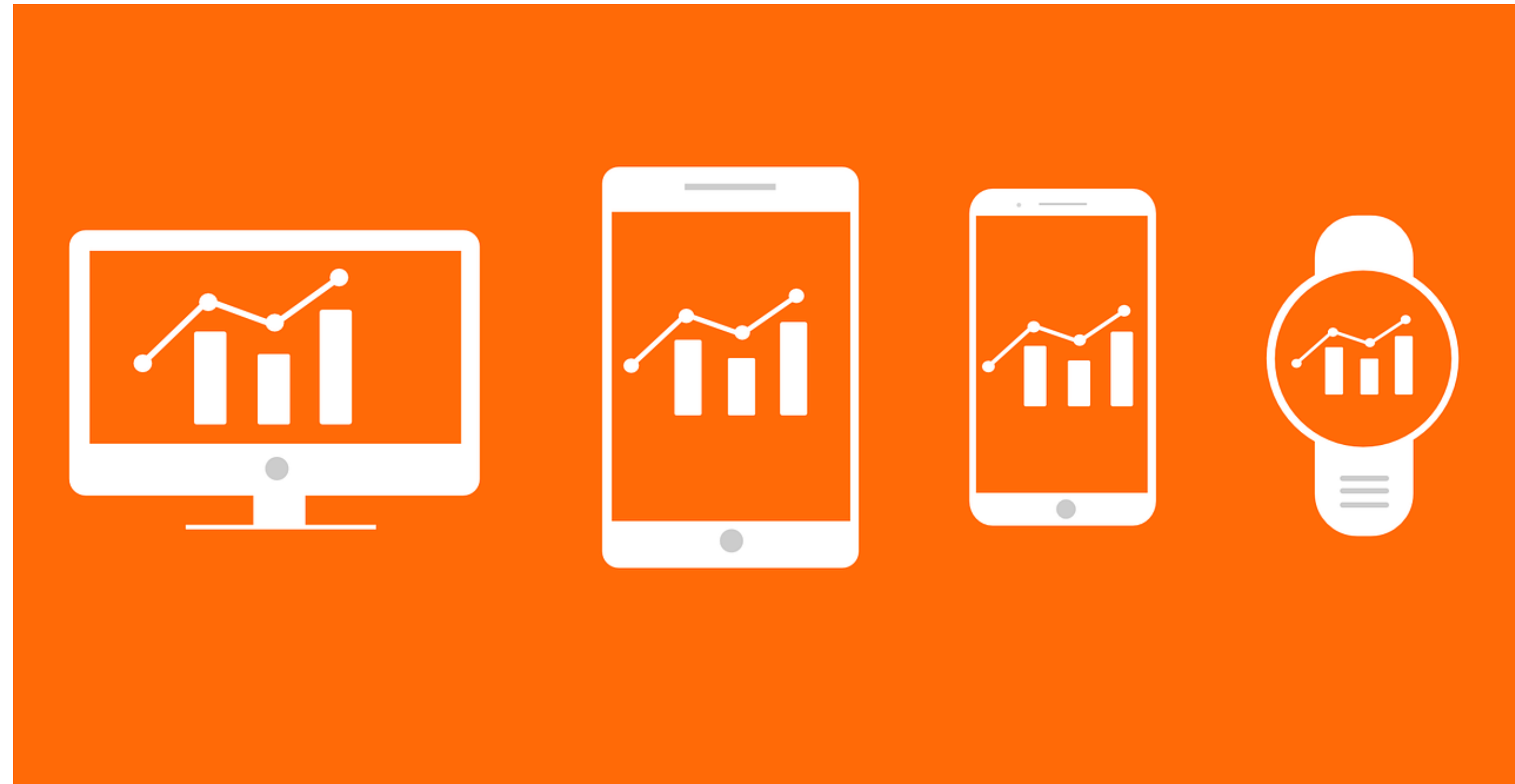
Desventajas del render en el servidor

- Seguridad
- Configuración de caché
- Sobrecarga de trabajo
- Mayor tiempo de carga entre páginas



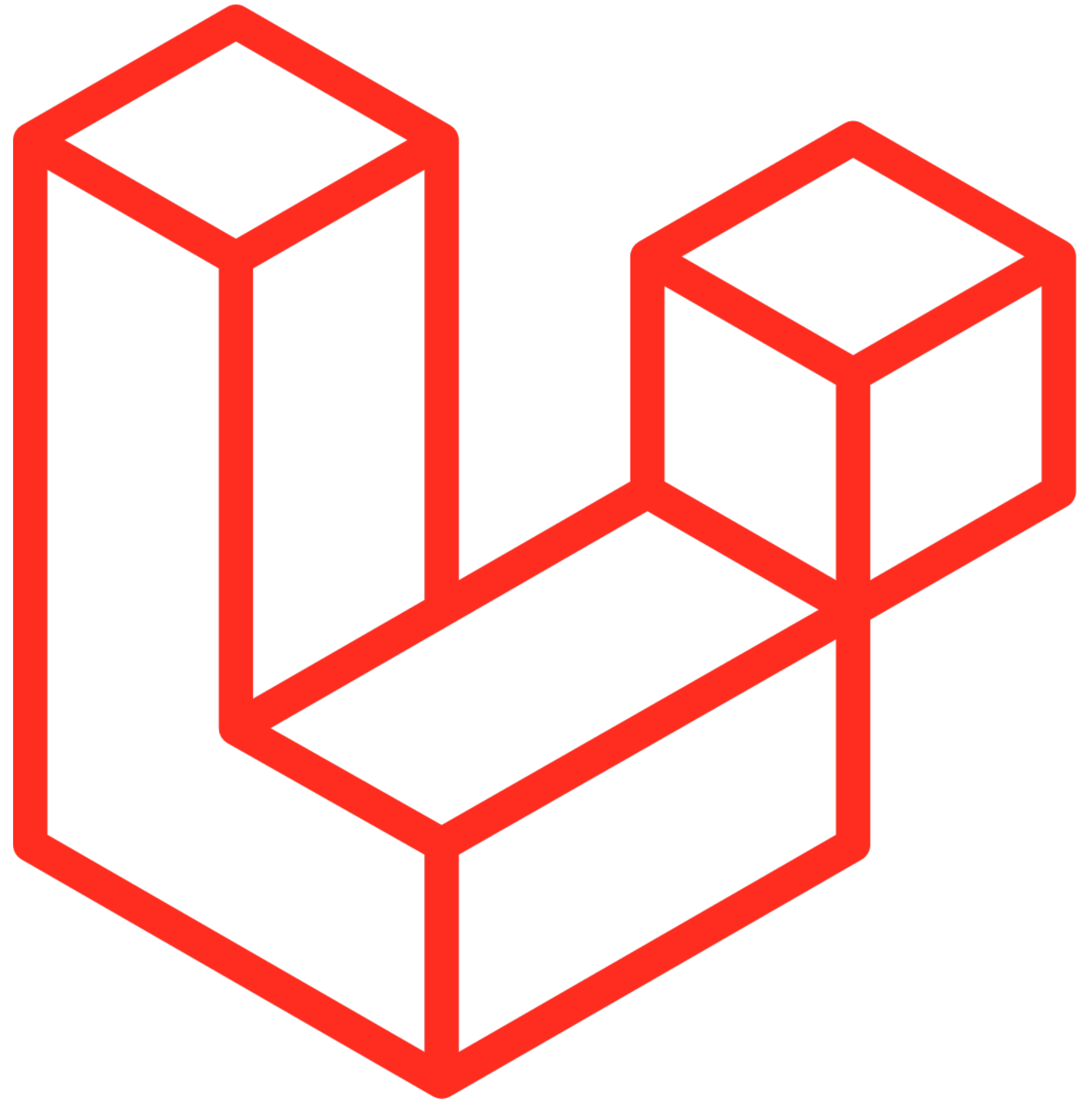
API Rest

- Interfaz de programación de aplicaciones.
- Representational State Transfer- Transferencia de Estado Representacional



Recursos

- Fiddler
- Postman
- PHPStorm/Visual Studio Code
- Git
- Composer



Proyectos

- App para el control de comensales en un restaurante
- App para el control de una línea de autobuses
- App para el control de una escuela
- App para el control de una renta de autos
- App para el control de un estacionamiento

```
dirname(self.filepath))

in the viewport
y.context.selected_objects

ort_selection
etting == False:
for i in bpy.context.scene.objects]

l(action='DESELECT')

ist:

H':
ath.join(folder_path, "{}.obj".format(item.name))
cene.obj(filepath=file_path, use_selection=True,
axis_forward=self.axis_forward_setting,
axis_up=self.axis_up_setting,
use_animation=self.use_animation_setting,
use_mesh_modifiers=self.use_mesh_modifiers_setting,
use_edges=self.use_edges_setting,
use_smooth_groups=self.use_smooth_groups_setting,
use_smooth_groups_bitflags=self.use_smooth_groups_bitflags_setting,
use_normals=self.use_normals_setting,
use_uv=self.use_uv_setting,
use_materials=self.use_materials_setting,
```


Plan de trabajo

- Instalación
- Apache
- Repaso de PHP
- API Rest PHP sin Framework
- API Rest con Laravel

```
realpath($_SERVER['DOCUMENT_ROOT'])  
[trim(preg_replace('/\\\\\\\\/', '/', $image_src), '/')];  
SION['_CAPTCHA']['config'] = serialize($captcha_config);  
array(  
ode' => $captcha_config['code'],  
age_src' => $image_src  
_exists('hex2rgb') ) {  
hex2rgb($hex_str, $return_string = false, $separator = ',') {  
str = preg_replace("/[^0-9A-Fa-f]/", '', $hex_str); // Gets a proper  
array = array();  
rlen($hex_str) == 6 ) {  
olor_val = hexdec($hex_str);  
b_array['r'] = 0xFF & ($color_val >> 0x10);  
b_array['g'] = 0xFF & ($color_val >> 0x8);  
b_array['b'] = 0xFF & $color_val;  
f( strlen($hex_str) == 3 ) {  
_array['r'] = hexdec(str_repeat(substr($hex_str, 0, 1), 2));  
_array['g'] = hexdec(str_repeat(substr($hex_str, 1, 1), 2));  
_array['b'] = hexdec(str_repeat(substr($hex_str, 2, 1), 2));  
n false;  
return_string ? implode($separator, $rgb
```