

Slim es un micro-framework para PHP que nos ayuda a desarrollar de forma más rápida aplicaciones web y APIs. Sus principales características son las siguientes:

- Es muy sencillo.
- Sistema de rutas potente.
- Rutas Restful.
- Tiene mensajes flash.
- Renderizado de vistas.

Veamos como instalar Slim Framework en Windows. En Linux el proceso es más sencillo si cabe.

Primero que todo debemos instalar **php**, **composer** y **git** en nuestro sistema y añadir sus variables de entorno al Path para que estén accesibles desde la consola.

En el directorio de nuestro webserver

```
mkdir slim-nombre-del-proyecto
```

→ El **composer.json** lo podemos agregar de dos formas.

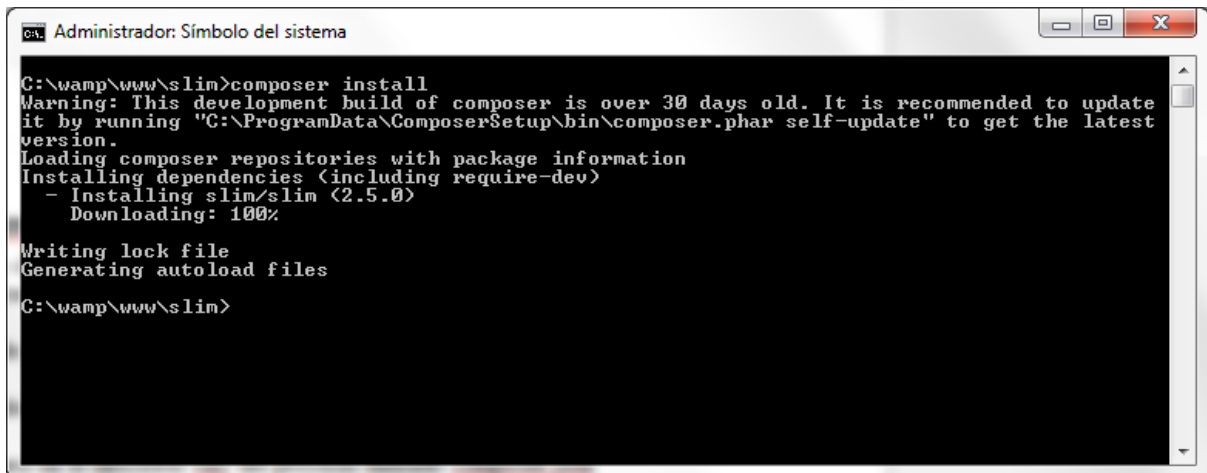
→ → **La primera:**

Creamos un fichero en el directorio raiz del proyecto llamado composer.json

```
{  
    "require": {  
        "slim/slim": "^4.0",  
        "slim/psr7": "^1.3",  
    }  
}
```

Ahora lanzamos el comando:

```
composer install
```



```
Administrador: Símbolo del sistema
C:\wamp\www\slim>composer install
Warning: This development build of composer is over 30 days old. It is recommended to update
it by running "C:\ProgramData\ComposerSetup\bin\composer.phar self-update" to get the latest
version.
Loading composer repositories with package information
Installing dependencies (including require-dev)
- Installing slim/slim (2.5.0)
  Downloading: 100%
Writing lock file
Generating autoload files
C:\wamp\www\slim>
```

Con esto instalamos los paquetes que especifica el json y se generó la carpeta vendor.

→ → **La segunda forma:**

Añadir los paquetes uno a uno. Para estos dos paquetes, podemos usar estos comandos:

```
composer require slim/slim
```

```
composer require slim/psr7
```

Podremos ver en el output de la consola como el primero que escribamos creará el composer.json y el segundo le hará un update.

*\*Tanto si utilizamos la primera forma o la segunda, podemos usar*

**composer require paquete**  
para instalar más paquetes.

→ Una vez creado el composer.json e instalados los paquetes (carpeta vendor):

**Creamos un index.php (preferentemente, en la carpeta app)**

```
<?php

require_once 'vendor/autoload.php';

$app = new \Slim\Slim();

$app->get('/hello/:name', function ($request, $response,
array $args) {
    $response->getBody()->write("Funciona!");

    return $response
});

$app->run();
?>
```

→ Faltaría agregar un archivo **.htaccess** en la **misma carpeta que el index** con el siguiente código:

```
RewriteEngine On

RewriteCond %{REQUEST_FILENAME} !-f

RewriteCond %{REQUEST_FILENAME} !-d

RewriteRule ^ index.php [QSA,L]
```

→ Ahí tenemos un hola mundo al cual podremos acceder.

Si el index.php está en la carpeta app:

```
php -S localhost:666 -t app
```

Sinó

```
php -S localhost:666
```

Otra carpeta?

```
php -S localhost:666 -t otraCarpeta/subCarpeta
```

→ ¿Y si lo quiero subir a Railway?

Dos cosas son necesarias:

1. Que el index esté en la carpeta app
2. Crear el archivo **nginx.template.conf** en la **raíz del proyecto** con el siguiente código:

<https://github.com/AgustinFrich/slim-php-deployment/blob/main/nginx.template.conf>

\*(El .htaccess no está en ese proyecto porque no es necesario en Railway, por eso no está en ese repositorio).

**Ante cualquier duda pueden avisar de cualquier error en el archivo por slack, classroom o meet. Si hubo algun error que quieran que quede documentado también.**

**¡Suerte!**