

# MANUAL DE INSTALACIÓN

Joan Casanovas | Rene Oviedo  
Jerrarquixs 2024-25

# ÍNDICE

INTRODUCCIÓN	3
OBJETIVO DEL PROYECTO	4
REQUISITOS A NIVEL HARDWARE Y SOFTWARE	5
SISTEMA DE ARCHIVOS	6
PROCESO DE INSTALACIÓN	9
PUESTA A PUNTO	10

# INTRODUCCIÓN

Este manual proporciona una guía detallada para la instalación y puesta en marcha de la aplicación web JoreFIT. Su objetivo es facilitar el despliegue correcto del sistema en un entorno local, de pruebas o de producción, asegurando que todos los componentes necesarios estén correctamente configurados y operativos.

La aplicación ha sido desarrollada con tecnologías actuales, entre las que destacamos Node.js, Angular, Laravel y php. Esta, ha sido diseñada para ejecutarse dentro de contenedores Docker, lo que garantiza una instalación reproducible, escalable y fácilmente mantenible.

Este documento está dirigido a desarrolladores, administradores de sistemas y cualquier persona encargada de instalar o mantener el entorno de ejecución de la aplicación. Se asume un conocimiento básico de la línea de comandos y, en algunos casos, de herramientas como Docker y Git.

A lo largo del manual se detallan los requisitos previos, la configuración del entorno, los pasos de instalación y verificación, así como las recomendaciones para un despliegue seguro y eficiente.

## OBJETIVO DEL PROYECTO

La aplicación web JoreFIT nace de la necesidad actual de acercar el deporte y los hábitos saludables a la población mundial, promoviendo un estilo de vida activo, accesible y sostenible. Más allá de ser una simple plataforma de entrenamiento, JoreFIT ha sido diseñada con un enfoque inclusivo, pensando en la diversidad de sus usuarios: personas de todas las edades, con distintos niveles de condición física, e incluso con limitaciones físicas o de movilidad.

La aplicación ofrece una variedad de planes de entrenamiento adaptados, lo que permite a cada usuario elegir el que mejor se ajuste a sus capacidades y objetivos personales. De esta forma, JoreFIT brinda a todos la oportunidad de avanzar hacia un bienestar físico real, sin barreras ni exclusiones.

## REQUISITOS A NIVEL HARDWARE Y SOFTWARE

A continuación, se mencionan y especifican los requisitos previos a la instalación, tanto a nivel de equipos informáticos como aplicaciones y/o programas requeridos.

### Equipo necesario a nivel de hardware

Servidor para el despliegue de la aplicación mediante Docker

Componente	Requisitos mínimos	Justificación
CPU	8 núcleos	Rendimiento sostenido sin interferencias
RAM	8GB	Permite mantener el programa en funcionamiento, incluso en horas de alto tráfico
MEMORIA (Disco duro SSD)	80-120GB	Evitamos el cuello de botella al preparar el servidor para pruebas locales y gestión de peticiones

### Requisitos a nivel de software - Servidor

- ▶ **Sistema Operativo:** Ubuntu 20.04 o superior.
- ▶ **Docker Engine:** Clave para el despliegue del programa en el servidor. Es necesaria la última versión estable.
- ▶ **Docker compose:** Última versión estable.
- ▶ **Firewall:** Medida de seguridad.

### Requisitos a nivel de software - Cliente

- ▶ **Navegador web:** Permite al cliente acceder al sitio web.
- ▶ **Conexión a internet:** Da acceso a internet.

# SISTEMA DE ARCHIVOS

A continuación, encontrarán el contenido del programa, sus directorios, subdirectorios y ficheros, junto con una breve explicación de su función. El objetivo de esta sección, es permitir al equipo encargado del mantenimiento conocer los distintos ficheros que conforman la aplicación, facilitando la gestión de este.

## Carpeta raíz

Al abrir el pendrive, el primer contacto con la estructura de ficheros es una carpeta que contendrá las distintas piezas del programa, separadas en 3 carpetas principales, "Angular", "Laravel", "BD".

## Directorio Angular

En este directorio, podrán encontrar todos los ficheros relacionados a la aplicación Angular, referentes al apartado visual de este.

Dentro del directorio Angular, existen varias subcarpetas que contienen los componentes encargados de formar la aplicación así como las imágenes utilizadas.

Por un lado, encontramos el directorio 'public/assets', que contiene las imágenes utilizadas por la propia aplicación web. Esta carpeta organiza las imágenes de la siguiente manera:

**publics/assets/ASSETS\_EJERCICIOS:** Contiene los assets (imágenes) correspondientes a cada ejercicio, incluyendo formatos como .jpg, .png y .webp

**publics/assets/ASSETS\_PLANES:** Contiene los assets (imágenes) correspondientes a cada uno de los planes existentes, incluyendo formatos como .jpg, .png, .webp y .jfif

**Resto de ficheros:** Contiene todos los demás ficheros, que si bien no están relacionados a los ejercicios ni a los planes, contienen imágenes e iconos utilizados en el programa, como el icono de esta.

Por otro lado, encontramos el directorio 'src', que contiene todos los componentes que construyen la aplicación y que permiten el correcto funcionamiento de esta.

**src/app:** Contiene todos los directorios con los diferentes componentes

▶ **src/app/add-admin:** Componente encargado de registrar un administrador

▶ **src/app/add-ejer-admin:** Componente encargado de registrar un nuevo ejercicio

- ▶ **/src/app/add-list-plan-ejer:** Componente encargado de mostrar los ejercicios de un plan
- ▶ **/src/app/add-plan:** Componente que permite al administrador agregar un plan
- ▶ **/src/app/cambiar-ejercicio:** Componente que permite modificar un ejercicio asignado
- ▶ **/src/app/ejer-asig:** Componente que muestra al usuario los ejercicios asignados correspondientes al plan que ha escogido.
- ▶ **/src/app/footer:** Pie de página de la página inicio.
- ▶ **/src/app/footer2:** Pie de página del resto de páginas.
- ▶ **/src/app/index-content:** Componente que contiene el contenido de la página de inicio.
- ▶ **/src/app/list-plans:** Muestra el listado de planes.
- ▶ **/src/app/list-plans-ejer:** Muestra los ejercicios correspondientes a un plan (los ejercicios por defecto de ese plan).
- ▶ **/src/app/login:** Muestra el formulario de inicio de sesión
- ▶ **/src/app/pago:** Muestra el formulario de inicio de sesión
- ▶ **/src/app/plam-admin:** Muestra la página que verá el administrador al administrar planes.
- ▶ **/src/app/plan-asig-admin:** Muestra la página que verá el administrador al administrar planes de un usuario.
- ▶ **/src/app/profil:** Muestra la página que verá el usuario al iniciar sesión
- ▶ **/src/app/profil-admin:** Muestra la página que verá el administrador al iniciar sesión
- ▶ **/src/app/recover-pwd:** Muestra el formulario de recuperar contraseña
- ▶ **/src/app/register:** Muestra el formulario para registrarse como cliente
- ▶ **/src/app/send:** Muestra la página encargada de informar al usuario que ha comprado un plan correctamente.
- ▶ **/src/app/shopping/add-card:** Muestra el formulario para agregar una tarjeta en el proceso de pago
- ▶ **/src/app/upd-admin:** Permitirá al administrador actualizar sus datos, mostrando los actuales dentro de los campos antes de rellenarlos.
- ▶ **/src/app/udp-usr:** Permitirá al cliente actualizar sus datos, mostrando los actuales dentro de los campos antes de rellenarlos.

- ▶ **/src/app/update-ejercicio:** Permitirá al administrador actualizar los datos de un ejercicio.
- ▶ **/src/app/update-plan:** Permitirá al administrador actualizar los datos de un plan.
- ▶ **/src/app/user-admin:** Muestra al administrador un listado con todos los usuarios actuales, permitiendo su gestión.
- ▶ **/src/app/user-admin-update:** Permitirá al administrador actualizar los datos de un usuario en su pantalla de administración.
- ▶ **/src/app/app.component.html:** Contiene la vista del header principal.
- ▶ **/src/app/app.component.css:** Contiene el estilo del header principal.
- ▶ **/src/app/app.routes.ts:** Contiene las url que enlazan a las distintas secciones de la web.
- ▶ **/src/app/validators:** Contiene los distintos ficheros que aseguran que los campos de los formularios sean correctos.
- ▶ **/src/app/services:** Contiene todos los servicios usados por el programa.
- ▶ **/src/app/guards:** Contiene todos los guards usados por el programa.
- ▶ **/src/app/interface:** Contiene todas las interfaces usadas por el programa.

## Directorio Laravel

En este directorio, podrán encontrar todos los ficheros relacionados a la aplicación Laravel, referentes a la interacción con la base de datos por parte de la aplicación web y la lógica de este.

- ▶ **/app/Controllers:** Contiene los diferentes controladores que gestionan la lógica
- ▶ **/app/Controllers/EjercicioController.php:** Realiza todas las funciones que afectan a los ejercicios, ya sea crear, modificar, borrar o mostrar.
- ▶ **/app/Controllers/PlanController.php:** Realiza todas las funciones que afectan a los planes, ya sea crear, modificar, borrar o mostrar.
- ▶ **/app/Controllers/UsuarioController.php:** Realiza todas las funciones que afectan a los usuarios, ya sea crear, modificar, borrar o mostrar.
- ▶ **/app/Middleware:** Se encarga de evitar que el usuario acceda a rutas desde la url.
- ▶ **/app/Models:** Contiene los diferentes modelos encargados de comunicarse con el Controlador correspondiente.



▶ **/app/Models/Ejercicio.php:** Define los atributos de un ejercicio y las relaciones con ejerciciosAsignados

▶ **/app/Models/EjercicioAsignado.php:** Define los atributos de un ejercicio asignado y las relaciones con plan, ejercicio y usuario

▶ **/app/Models/Plan.php:** Define los atributos de un plan y las relaciones con planAsignados y ejercicioAsignados

▶ **/app/Models/PlanAsignado.php:** Define los atributos de un plan asignado y las relaciones con plan y usuario

▶ **/app/Models/Usuario.php:** Define los atributos de un usuario y las relaciones con planesAsignados y ejerciciosAsignados

▶ **/database:** Contiene los directorios con información correspondientes a la base de datos.

▶ **/database/migrations/2025\_01\_22\_153557\_create\_jorefit\_m\_s\_table.php:** Establece cómo se crean las tablas de la base de datos

▶ **/database/seeder:** Contiene los ficheros que poblarán la base de datos con datos de prueba.

▶ **/database/seeder/EjerciciosSeeder.php:** En este fichero se definen los valores que poblarán la tabla de ejercicios en la base de datos.

▶ **/database/seeder/PlanesSeeder.php:** En este fichero se definen los valores que poblarán la tabla de planes en la base de datos.

▶ **/database/seeder/UsuariosSeeder.php:** En este fichero se definen los valores que poblarán la tabla de usuarios en la base de datos.

▶ **/database/seeder/DatabaseSeeder.php:** En este fichero se llaman a los seeder anteriores para de esta manera insertar los datos en la base de datos.

▶ **/routes/api.php:** En este fichero se definen las distintas rutas junto con su método (GET, POST, PATCH,DELETE)

▶ **.env:** En este fichero se establecen el nombre de la base de datos, la ip y las credenciales de acceso. En el programa, se establece *DB\_DATABASE*, con el valor *jorefit* (el nombre de la base de datos)

**Directorio BD**

En este directorio, podrán encontrar el fichero con la estructura de la base de datos. Es importante mencionar, que para poder hacer uso de esta, es necesario un gestor de bases de datos PhpMyAdmin, por tanto, programas como XAMPP o WAMPP son clave.

# PROCESO DE INSTALACIÓN

Con el objetivo de instalar la aplicación web JoreFIT, a continuación se explica detalladamente el proceso a seguir para poder instalar este programa. Para empezar, partimos de la base de que ya cuenta con un equipo informático.

## Proceso de instalación de Angular

### Instalación de Node.js

Diríjase a su navegador de preferencia y entre en el sitio web <https://nodejs.org/en/download> y pulse en el botón Windows Installer. Una vez descargado, descomprima en el directorio que usted desee.

### Instalación de Angular CLI

Pulse el botón con el icono de Windows en su teclado o en la barra de tareas y escriba cmd, luego pulse en el programa mostrado o pulse enter.

Dentro de la consola de windows (cmd) escriba el siguiente comando para instalar Angular CLI

```
npm install -g @angular/cli
```

Tras esto, ya tendrá instalado Angular CLI en su sistema

## Proceso de instalación de Laravel

### Instalación de php

Ejecute el siguiente comando en la consola de Linux, que instale php en su sistema Ubuntu/Debian

```
sudo apt update  
sudo apt install php php-cli php-mbstring php-xml php-bcmath php-curl unzip curl git -y
```

Una vez completado el proceso, puede validar si se ha instalado correctamente mediante el siguiente comando

```
php -v
```

Laravel depende de Composer para su correcto funcionamiento, por lo que deberemos instalarlo. A continuación se explica como instalar este en un sistema Linux basado en Ubuntu o Debian.

## Instalación de Composer

Ejecute el siguiente comando en su terminal de Ubuntu/Debian. Este instalara Composer en su sistema.

```
curl -sS https://getcomposer.org/installer | php  
sudo mv composer.phar /usr/local/bin/composer
```

Una vez completado, puede validar si se ha instalado correctamente mediante el siguiente comando

```
composer -V
```

Finalmente, deberemos instalar Laravel. Podemos hacerlo en Ubuntu/Debian de la siguiente manera

```
composer global require laravel/installer
```

Una vez instalado, deberemos agregarlo al PATH. Podemos hacerlo de la siguiente manera

```
echo 'export PATH="$HOME/.config/composer/vendor/bin:$PATH"' >> ~/.bashrc  
source ~/.bashrc
```

## PUESTA A PUNTO

A continuación, mencionaremos y explicaremos el proceso a seguir para desplegar tanto la aplicación Laravel en el servidor como la aplicación Angular en los equipos de los administradores.

### Despliegue de la aplicación Laravel en el servidor

En la máquina servidor, acceda al directorio donde se instaló el servicio de gestión de base de datos (ej. XAMPP) y en la consola de Ubuntu/Debian, escriba el siguiente comando

```
php artisan serve
```

Esto levantará el servidor PHP, permitiendo a este atender solicitudes por parte de los clientes.

### Despliegue de la aplicación Angular en el servidor

En una máquina cliente abra un nuevo terminal (Windows o Linux) y diríjase al directorio que contiene los ficheros pertenecientes a la aplicación Angular mediante el comando "cd"

Una vez en el directorio, escriba el siguiente comando para lanzar la aplicación Angular

```
ng serve --open
```

Con esto, habrá desplegado el programa correctamente. Por último, abra su gestor de bases de datos, en este caso XAMPP, agregue una nueva base de datos con el nombre jorefit e importe el fichero *jorefit.sql*. Esto creará las tablas necesarias y las poblara con datos de ejemplo.