

# Manual Técnico



***PROGYM***

# Índice

## Contenido

Introducción.....	2
Requisitos del Sistema .....	3
Software .....	3
Hardware .....	3
Estructura del Proyecto .....	4
Instalación.....	5
Clonación del repositorio:.....	5
Instalación de dependencias:.....	5
Configurar las variables de entorno:.....	5
Configuración inicial:.....	5
Usuarios predeterminados: .....	6
Componentes Principales.....	7
Backend .....	7
Frontend .....	7
Base de Datos .....	7
Desplegar en Producción: .....	7
Comando de optimización.....	8
Instalacion de Nginx.....	8
Migraciones y Seeds.....	9
Mantenimiento.....	10
Limpieza y Mantenimiento preventivo: .....	10
Seguridad .....	10
Referencias .....	11
Bibliografía .....	11



## Introducción

Gym Manager Pro es un sistema administrativo diseñado para la gestión eficiente de gimnasios, ofreciendo funcionalidades como la gestión de membresías, control de pagos y reportes. Este manual está dirigido a desarrolladores y administradores técnicos que necesiten entender, mantener o expandir el sistema.



## Requisitos del Sistema

### Software

- Servidor web: Nginx o Apache
- PHP: Versión 8.1 o superior
- Base de datos: MySQL 5.7 o superior
- Composer: Última versión
- Node.js: Versión 14 o superior (para gestionar activos del frontend)
- Git: Para control de versiones

### Hardware

- Procesador de 2 núcleos
- Memoria RAM mínimo 4 GB
- Almacenamiento mínimo de 20 GB



## Estructura del Proyecto

- **app/**: Contiene los controladores, modelos, y lógica de negocio.
- **bootstrap/**: Configuración inicial de Laravel.
- **config/**: Archivos de configuración del sistema.
- **database/**: Migraciones y seeds.
- **public/**: Archivos estáticos accesibles públicamente (CSS, JS, imágenes).
- **resources/**: Vistas Blade, archivos de idiomas y componentes del frontend.
- **routes/**: Define las rutas de la aplicación.
- **storage/**: Logs, caché y otros archivos generados.
- **tests/**: Pruebas automatizadas.



## Instalación

Clonación del repositorio:

```
git clone https://github.com/VasqCD/Gym_Manager_v2.git
```

```
cd Gym_Manager_v2
```

Instalación de dependencias:

Una vez en la carpeta se debe instalar las dependencias

```
composer install
```

```
npm install
```

Configurar las variables de entorno:

1. Se debe renombrar .env.example a .env. puede usar el comando, cp en Linux para crear una copia del mismo

Debe configurar las variables en la base de datos ejemplo:

```
APP_NAME=Gym_Manager_v2
APP_URL=http://localhost
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=gym_manager
DB_USERNAME=root
DB_PASSWORD=yourpassword
```

Configuración inicial:

Una vez teniendo configura las variables de entorno ejecutar la configuración inicial en la terminal (debe estar en la carpeta en la cual tiene el directorio Gym)

```
php artisan key:generate
```

```
php artisan migrate --seed
```



**PROGYM**

*php artisan storage:link*

### Usuarios predeterminados:

Al ejecutar esto en el sistema se crearán los roles y permisos que el sistema requiere además de los usuarios básico para acceder al sistema, puede crear nuevos usuarios y asignar roles correspondientes con una contraseña segura.

Los usuarios generados si a creado nuevos usuarios se recomienda eliminarlos o cambiar la contraseña por defecto de estos.

### Usuarios predeterminados:

*Nombre: correo / Contraseña*

*SuperAdmin: superadmin@gymflow.com / SuperAdmin2024\**

*Admin: admin@gymflow.com / Admin2024\**

*Gerente: gerente@gymflow.com / Gerente2024\**

*Recepción: recepcion@gymflow.com / Recepcion2024\**

*Entrenador: entrenador@gymflow.com / Entrenador2024\**



## Componentes Principales

### Backend

- PHP
- Framework: Laravel 10
- Controladores: Gestionan la lógica de las rutas y la interacción con el modelo.  
Ejemplo: UserController, ClienteController.
- Modelos: Manejan las interacciones con la base de datos.
- Ejemplo: User, Cliente.

### Frontend

Motor de plantillas: Blade

Tecnologías adicionales: HTML, CSS, JavaScript.

Gestión de paquetes: Laravel Mix (Webpack).

### Base de Datos

- Gestor: MySQL
- Migraciones: Define la estructura de las tablas.
- Seeds: Inserta datos iniciales para pruebas, incluyendo usuarios predeterminados.

## Desplegar en Producción:

En este caso de uso se a desplegado en una instancia de EC2 en Amazon web services, con la distribución de Ubuntu server.

Teniendo el sistema listo se debe instalar el servidor web deseado, en este caso la configuración siguiente será nginx:

### Configuración del Servidor





Servidor web: Configurar Nginx o Apache para servir la aplicación.

## Comando de optimización

```
sudo apt update && sudo apt upgrade -y
```

```
php artisan config:cache
```

```
php artisan route:cache
```

## Instalacion de Nginx

```
sudo apt install nginx -y
```

Habilitar una vez instalado con:

```
sudo systemctl start nginx
```

```
sudo systemctl enable nginx
```

## Instalar MySQL

```
sudo apt install mysql-server -y
```

Instalar PHP ademas de extensiones necesarias

```
sudo apt install php-fpm php-mysql php-cli php-curl php-gd php-mbstring php-xml php-zip -y
```

Una vez instalado php se debe **configurar Nginx para Laravel**: Creando un nuevo bloque de servidor para el proyecto

```
sudo nano /etc/nginx/sites-available/gym_manager
```

Dentro del archivo se debe colocar la siguiente configuración básica,

```
server {  
    listen 80;  
    server_name tu_dominio.com;  
    root /ruta/a/tu/proyecto/public;  
  
    index index.php index.html index.htm;  
  
    location / {  
        try_files $uri $uri/ /index.php?$query_string;  
    }  
}
```



```
location ~ /\.php$ {  
    include snippets/fastcgi-php.conf;  
    fastcgi_pass unix:/var/run/php/php7.4-fpm.sock;  
    fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;  
    include fastcgi_params;  
}  
  
location ~ /\.ht {  
    deny all;  
}  
}
```

Esta configuración se puede adaptar a los requerimientos como ser la necesidad de certificados subdominios etc.

Una vez realizado los cambios en el archivo se debe guardar y crear un enlace simbólico para habilitar el sitio.

Se puede hacer con el siguiente comando:

```
sudo ln -s /etc/nginx/sites-available/gym_manager /etc/nginx/sites-enabled/
```

Una vez realizado todos los cambios necesarios y verificando que no hallan errores se debe reiniciar nginx con `sudo systemctl restart nginx`. Posteriormente se debe instalar composer.

```
sudo apt install composer -y
```

Luego dirigirse a la ruta donde está instalado el proyecto o carpeta según el nombre que el allá colocado, dentro de la misma ejecutar el comando

```
composer install
```

Aquí puede continuar y aplicar lo visto anteriormente en el capítulo Instalación, finalizado el proceso deberá tener funcionando correctamente la aplicación web.

Instalará todas las dependencias necesarias para el buen funcionamiento.

## Migraciones y Seeds

```
php artisan migrate --force
```

```
php artisan db:seed --force
```



## Mantenimiento

- **Backup de la base de datos:** Configurar un cron job.
- **Actualización de dependencias**
- **Verificar regularmente el servidor**

Limpieza y Mantenimiento preventivo:

### Archivos Temporales:

- Limpia archivos temporales de Laravel
  - php artisan cache:clear
  - php artisan config:clear
  - php artisan route:clear
  - php artisan view:clear

## Seguridad

**Contraseñas y Roles:** Cambia las contraseñas de las cuentas críticas periódicamente. Revisa y ajusta los roles y permisos de usuarios en el sistema.

**Actualizaciones de Seguridad:** Instala actualizaciones de seguridad para el sistema operativo y los paquetes de servidor.



## Referencias

### Bibliografía

LARAVEL. (s.f.). *LARAVEL*. Obtenido de LARAVEL: <https://laravel.com/docs/11.x>

Vasquez, C. (2024). *GitHub*. Obtenido de Gym Manager V2:  
[https://github.com/VasqCD/Gym\\_Manager\\_v2](https://github.com/VasqCD/Gym_Manager_v2)

Vasquez, C. (s.f.). *Documentacion Gym Manager Pro*. Obtenido de Progym:  
<https://documentacion.gymmanager.icu>