SOFTWARE DE GESTIÓN DE MANTENIMIENTO PARA LA INMOBILIARIA DEL TURISMO EN HOLGUÍN

GUÍA DE INSTALACIÓN

Autor: Víctor Daniel Rivas Ferrer

Tutores: Ing. Adrián David Duque Iglesias

Ing. Cristhian Arlan Cecilio Tejeda

HOLGUÍN 2025

ÍNDICE

[REQUERIMIENTOS 1](#_Toc199761822)

[Verificación de instalaciones existentes 1](#_Toc199761823)

[INSTALACIÓN DE LOS REQUERIMIENTOS 2](#_Toc199761824)

[Node.js v20 2](#_Toc199761825)

[Postgres 12 2](#_Toc199761826)

[Apache 2 3](#_Toc199761827)

[Configurar Apache 3](#_Toc199761828)

[Instalaciones adicionales 4](#_Toc199761829)

[INSTALACIÓN DEL SOFTWARE 5](#_Toc199761830)

[Configurar API 5](#_Toc199761831)

[Configurar web 6](#_Toc199761832)

[EJECUCIÓN DE LA APLICACIÓN 7](#_Toc199761833)

[Levantar aplicación 7](#_Toc199761834)

[Actualización de la aplicación 7](#_Toc199761835)

[CREDENCIALES 8](#_Toc199761836)

# REQUERIMIENTOS

Para el correcto funcionamiento del software, se requieren los siguientes componentes instalados en el sistema:

1. **Node.js v20**
   * Entorno de ejecución JavaScript necesario para el *backend*
   * Versión LTS recomendada para soporte extendido
2. **PostgreSQL v12 o superior**
   * Sistema de gestión de bases de datos relacional
   * Requerido para el almacenamiento persistente de datos
3. **Apache 2.4 o superior**
   * Servidor web HTTP
   * Necesario para el despliegue del *frontend* y configuración de proxy

**Notas importantes:**

* Todas las versiones indicadas corresponden al mínimo requerido
* Se recomienda utilizar las últimas versiones estables disponibles
* En entornos de producción, utilizar exclusivamente versiones LTS

## Verificación de instalaciones existentes

Si ya alguno instalado se puede verificar con:

node *--version*

psql *--version*

apache2 *-v*

# INSTALACIÓN DE LOS REQUERIMIENTOS

A continuación se detallan los procedimientos oficiales para la instalación de cada componente requerido:

## Node.js v20

Node.js es un entorno de ejecución para JavaScript necesario para el *backend* del sistema.

Primero instalar *Node Version Mana*ger (NVM):

curl *-o-* https://raw.githubusercontent.com/nvm-sh/nvm/v0.40.3/install.sh *|* bash

source ~/.bashrc

Luego, con *nvm* instalar la versión de *node*:

nvm install 20.13.1

## Postgres 12

PostgreSQL es un sistema de gestión de bases de datos relacionales requerido para el almacenamiento persistente de datos.

Instalar el paquete:

sudo apt *-y* install postgresql

E iniciar el proceso con:

sudo systemctl start postgresql

Puede modificarse la contraseña del usuario *postgres* y la forma en el que el sistema lo trata así:

sudo *-u* postgres psql

ALTER USER postgres WITH PASSWORD 'postgres';

*# Editar pg\_hba.conf*

## Apache 2

Apache HTTP Server es el servidor web requerido para el despliegue del *frontend* y configuración de proxy inverso.

Instalar su paquete:

sudo apt *-y* install apache2

Y luego se inicia:

sudo systemctl start apache2

### Configurar Apache

Configurar Virtual Host para que acceda al *build* de la aplicación *web:*

sudo nano /etc/apache2/sites-available/[nombre-sitio].conf

Agregar la configuración*:*

*<*VirtualHost *\**:*80>*

    ServerAdmin admin@dominio.com

    ServerName dominio.com

   DocumentRoot /var/www/[directorio]

    ErrorLog ${APACHE\_LOG\_DIR}/error.log

    CustomLog ${APACHE\_LOG\_DIR}/access.log combined

*<*/VirtualHost*>*

Ejecutar la configuración y recargar apache*:*

sudo a2ensite [nombre-sitio].conf

sudo systemctl reload apache2

## Instalaciones adicionales

Se puede instalar *yarn* como versión alterna al gestor de paquetes *nvm* así:

npm install *-g* yarn

Para manejar varios procesos de *node* en segundo plano se puede instalar *pm2*:

npm install *-g* pm2

*Sequelize-cli* con *postgres* es requerido para usar los scripts del API:

npm install *-g* sequelize-cli pg pg-hstore

# INSTALACIÓN DEL SOFTWARE

Para traer el código de la aplicación se puede hacer con:

git clone <https://github.com/VictorDRF02/mangest-api>

git clone <https://github.com/VictorDRF02/mangest-web>

## Configurar API

Se accede al API:

cd mangest-api

Luego se instalan las dependencias:

yarn install

Crear la base de datos y llenarla:

yarn run db:create

yarn run db:migrate

yarn run db:seed

Crear *ecosystem.config* y configurarlo:

  module.exports *=* {

    apps: [{

      name: 'mangest-api',

      script: './src/bin/www',

      watch: *true*,

      ignore\_watch: ['node\_modules', '.git'],

      env\_production: {

        NODE\_ENV: 'production',

        DB\_HOST: "127.0.0.1",

        DB\_NAME: "db\_mangest\_development",

        DB\_USER: "postgres",

        DB\_PASSWORD: "postgres",

      }

    },

    ],

  };

Salir de la carpeta:

cd ..

## Configurar web

Se accede a la Web:

cd mangest-web

Luego se instalan las dependencias:

yarn install

Hacer el *build*:

yarn run build

Salir de la carpeta:

cd ..

# EJECUCIÓN DE LA APLICACIÓN

## Levantar aplicación

Para levantar el *api* se ejecuta lo siguiente:

    cd mangest-api

    pm2 start ecosystem.config.js

    cd ..

## Actualización de la aplicación

Se actualiza el *api*:

cd mangest-api

git pull

yarn run db:migrate

yarn run db:seed

cd ..

Se actualiza la *web*:

cd mangest-web

git pull

yarn run build

cd ..

# CREDENCIALES

Las credenciales para acceder al sistema son:

Nombre de usuario: General Administrator

Contraseña: 1234

**Se recomienda cambiarlas al desplegar el sistema.**