

**Manual despliegue**

Versión 1.0

Febrero – 2023

Desarrollado por **Angel Loor**

# Introducción

Para la puesta en producción del sistema de facturación se ha utilizado un entorno de UNIX, utilizando la distro Ubuntu 20.04, la cual permitió instalar el entorno de ejecución de NODEJS, a continuación, se listan los pasos a seguir para replicar el despliegue del API, y portal del sistema de facturación denominado WSFE.

# Pasos a seguir

1. Primero hay que levantar una instancia de Linux con Ubuntu 20.04.
2. Actualizar la instancia con los siguientes comandos.

sudo apt update

1. Instalar dependencias (wget curl ca-certificates)

sudo apt install wget curl ca-certificates

1. Instalar PostgreSQL 13

sudo sh -c 'echo "deb http://apt.postgresql.org/pub/repos/apt $(lsb\_release -cs)-pgdg main" > /etc/apt/sources.list.d/pgdg.list'

wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo apt-key add -

sudo apt-get update

sudo apt-get -y install postgresql-13

1. Definir contraseña al usuario postgres

su postgres

psql

alter user postgres with password 'nuevaContraseña';

1. Reiniciar los servicios de PostgreSQL

systemctl restart postgresql

1. Configurar white list de PostgreSQL para que pueda ser accesible solo por direcciones pertinentes.

sudo vi /etc/postgresql/13/main/pg\_hba.conf

IPV4

host    all             all             localhost/32   md5

host    all             all             ip\_instancia/32   md5

host    all             all             ip\_conexion\_remota/32   md5

1. Habilitar listen\_addresses en **postgresql.conf**

sudo vi /etc/postgresql/13/main/postgresql.conf

listen\_addresses = 'ip\_instancia'

1. Instalar NodeJS

sudo apt install nodejs

## Comprobar version

nodejs -v

1. Instalar npm

sudo apt install npm

## Comprobar version

npm -v

1. Instalar pm2

sudo apt install pm2

1. Instalar node-gyp node-pre-gyp

sudo npm i -g node-gyp

sudo npm I -g node-pre-gyp

1. Después de instalar todas las dependencias necesarias, se tiene que crear la base de datos, para mayor comodidad es necesario hacer una conexión remota desde PgAdmin 4 hacia el servidor, (Es importante mencionar que para que se pueda hacer la conexión se tiene que habilitar la Ip del equipo de donde se piensa hacer la conexión en **pg\_hba.conf** (**Paso 7**)).
2. Una vez realizado la conexión remota al servidor se tiene que crear la base de datos con el script **create.sql** que se encuentra en la siguiente ruta.

**Ruta:** Municipio\Proyectos\nashor\nashor-backend\resource\database\create.sql

1. Una vez creado la base de datos, se tiene que pasar los archivos del backend al servidor previamente configurado, para eso tenemos que utilizar un SSH File Transfer, ejemplo WINSCP.
2. Antes de pasar todos los archivos del backend hacia la instancia se tiene que compilar el código, navegaremos hacia la siguiente ruta con una consola.

**Ruta:** Municipio\Proyectos\nashor\nashor-backend\

Una vez en la ruta, tenemos que correr los siguientes comandos.

**npm i** (instalar dependencias)

**npm run tsc** (transpilar el código ts a js)

**npm run build** (compilar el código js)

1. Una vez realizado todo el proceso de compilación, procedemos a pasar los archivos al servidor.

Ya teniendo WINSCP o cualquier otro Software similar se tiene que hacer la conexión al servidor remoto, y pasar la carpeta del proyecto dentro de la instancia, no hay una ruta por defecto, usted la puede poner en cualquier ubicación, ya que el entorno de NodeJS es global.

Nota: no es necesario copiar todos los archivos, a continuación, se lista los archivos obligatorios para que el servidor se levante:

A picture containing text

Description automatically generated

Tanto el node\_modules como el package-lock.json, se generan automáticamente usando el comando npm i

1. Después de pasar los archivos dentro de la instancia, se tiene que realizar una conexión SSH al servidor para poder instalar las dependencias y levantarlo.

Para instalar las dependencias hay que entrar donde está ubicado el package.json y utilizar el comando **npm i** para que npm instale todas las dependencias necesarias.

Una vez instalado las dependencias se tiene que compilar el frontend y ubicarlo dentro de la carpeta public del backend.

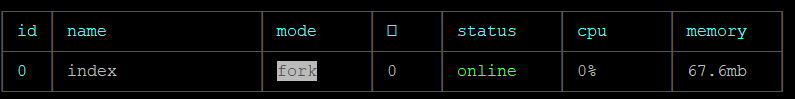
1. Para compilar el frontend se tiene que navegar al package.json del frontend y utilizar el comando **npm run build** una vez utilizado este comando se generara la compilación dentro de la carpeta **dist/wsfe-frontend.**

Hay que copiar todos los archivos de **./dist/nashor-frontend** a **./public/** (Dentro de la instancia).

1. Una vez realizado esto se puedo levantar el servicio.

Para levantar el servicio utilizaremos el gestor de procesos de **PM2** el cual ya lo instalamos previamente.

Para levantar el servicio utilizaremos el comando **pm2 start ./dist/index.js** después de presionar el siguiente comando nos mostrara un panel similar al siguiente.



Y listo ya está levantado el servicio.

El servicio estará corriendo bajo **HTTP**:**IP\_INSTANCIA**:**80**

**Notas:**

* Con respecto al dominio se puede utilizar bind9 para generar los registros de dominio o se puede administrar directamente desde la configuración del proxy.

En este caso en particular Beto lo administra directamente desde el proxy.

* Otro tema importante es el HTTPS, el cual también se lo está manejando desde el proxy, la otra forma es levantando directamente el servicio con el puerto 443, y utilizando una clave pública y privada, pero para este caso práctico no aplica.