



---

## MANUAL TÉCNICO

---

### GEOPORTAL MAPA DE RUIDO DE MEDELLÍN



ABRIL 2024

UNIVERSIDAD SAN BUENAVENTURA / AREA METROPOLITANA DEL VALLE DE ABURRA  
Medellín, Antioquia

# MANUAL TÉCNICO

## Contenido

1. INTRODUCCIÓN .....	1
2. COMPONENTES TÉCNICOS.....	2
2.1. Cliente.....	2
2.2. Servidor.....	3
3. HERRAMIENTA DE SOPORTE .....	3
4. INGRESO AL SERVIDOR.....	4
5. CONFIGURACIÓN.....	5
5.1. Creación del sistema de gestión .....	5
5.2. Creación del entorno virtual .....	5
5.3. Activación del entorno virtual .....	6
ANEXO.....	7

## Tabla de figuras

Figura 1. Arquitectura servidor AMVA.....	2
Figura 2. Estado del servidor.....	5

## Tabla de tablas

Tabla 1. Capacidad de componentes del servidor.....	2
Tabla 2. Lenguajes del proyecto [cliente] .....	2
Tabla 3. Lenguajes del proyecto [servidor] .....	3
Tabla 4. Herramientas de soporte .....	3

## Tabla de anexos

Anexo 1Credenciales de acceso .....	7
Anexo 3. Código de administración.....	7
Anexo 2. Código de configuración del despliegue.....	7
Anexo 4. Versiones de tecnologías instaladas .....	7

## 1. INTRODUCCIÓN

El documento tiene el objetivo de mostrar los elementos y procedimientos respectivos para realizar mantenimiento y actualizaciones de la infraestructura y el proyecto del Geoportal del Mapa de ruido del Área Metropolitana del Valle de Aburrá.

Esta muestra las herramientas y servicios empleados, como también los microservicios y estructuras que permiten su funcionamiento, los lenguajes aplicados, esto con el objetivo de proporcionar una guía que permita mantener el proyecto habilitado y su mantenimiento no presente mayor problema y resistencia en cuanto actualizaciones y mantenibilidad.

## 2. COMPONENTES TÉCNICOS

El proyecto se encuentra alojada en los servidores proporcionados por el AMVA. El proyecto no cuenta con un dominio, se trabaja con la dirección privada y se soporta en una arquitectura Linux Ubuntu 22.04 presentada en la Figura 1.

```
maparuido@ammaparuido01:/$ lscpu
Architecture:          x86_64
  CPU op-mode(s):      32-bit, 64-bit
  Address sizes:        44 bits physical, 48 bits virtual
  Byte Order:           Little Endian
CPU(s):                8
  On-line CPU(s) list: 0-7
Vendor ID:             GenuineIntel
  Model name:           Intel(R) Xeon(R) Gold 6130 CPU @ 2.10GHz
    CPU family:         6
    Model:              85
    Thread(s) per core: 1
    Core(s) per socket: 8
    Socket(s):          1
    Stepping:           4
    Bogomips:           4200.00
```

Figura 1. Arquitectura servidor AMVA

Tabla 1. Capacidad de componentes del servidor

Componente	Capacidad
CPU	73GB
RAM	9GB
CORES	8

### 2.1. Cliente

Tabla 2. Lenguajes del proyecto [cliente]

Lenguaje	Propósito
PYTHON	Lenguaje base del proyecto. Este permite tener comunicación entre el cliente y el servidor, adicional actual como el motor de procesamiento de información.

HTML	Lenguaje de maquetado. Este permite crear la estructura del cliente.
CSS	Lenguaje de estilos. Este permite brindarle un aspecto adecuado y un diseño deseado al proyecto.
JAVASCRIPT	Lenguaje de interacción y dinamismo. Este lenguaje permite brindarle al proyecto un entorno dinámico, trabajar con microservicios y procesamiento de información.

## 2.2. Servidor

Tabla 3. Lenguajes del proyecto [servidor]

Lenguaje	Propósito
SQL	Lenguaje de conexión e interacción con la base de datos. Este permite realizar las peticiones y gestionar la información entre el cliente y el servidor.
BASH	Lenguaje interno de servidor. Este permite establecer la conexión y despliegue del proyecto con el cliente.

## 3. HERRAMIENTA DE SOPORTE

Tabla 4. Herramientas de soporte

GIT	Lenguaje y microservicio para el registro y gestión del sistema de versiones. Este permite establecer conexión entre el sistema de versiones y el servidor, al igual que actuara como el motor de administración del proyecto.
SUPERVISOR	Es un sistema que se encarga de administrar procesos. Este actúa como un microservicio para el monitoreo del sistema y automatización de puesta en marcha.
PIP	Es un sistema de gestión de paquetes de Python. Este motor provee los archivos y paquetes de distintas librerías de Python.

#### 4. INGRESO AL SERVIDOR

Todo el proceso se realiza en el servidor, por lo que se debe tener a mano las credenciales para su ingreso, el proceso a continuación explicado se basa en el ingreso por medio de ssh. No obstante, el sistema se halla configurado para entrar con otros tipos de tecnologías como FTP, webDAV entre otros.

El primer proceso que se debe realizar en caso de realizar la instalación del proyecto consiste en clonar el proyecto desde el repositorio indicado por el ingeniero. Para llevar a cabo este proceso, aplique el siguiente protocolo para clonar el proyecto en el servidor:

1. Active el sistema Forticlient, verifique que el sistema si se conectó, en caso contrario verifique el usuario y contraseña.
2. Abra la consola Windows [cmd] o Linux [terminal]
3. Ingrese por medio del ssh con la siguiente instrucción → `ssh user@ip`
4. Ingrese la contraseña
5. Identifique el directorio en que se encuentra con el comando → `pwd`
6. Diríjase a la raíz con el comando → `cd /`
7. Cree un nuevo directorio con el siguiente comando → `mkdir /nuevo_directorio`
8. Clone el repositorio con el siguiente comando → `git clone url_del_proyecto`

**Nota:** Sin la activación de la VPN, no es posible el ingreso al servidor, debe tener a mano las credenciales de activación del servicio de VPN.

Para actualizar el proyecto, aplique el siguiente protocolo, este procedimiento se debe realizar al interior del directorio de trabajo. A continuación, se indica como se realiza el ingreso y ubicación del proyecto, como también su actualización:

9. Después de activar la VPN, abra la consola Windows [cmd] o Linux [terminal]
10. Ingrese por medio del ssh con la siguiente instrucción → `ssh user@ip`
11. Ingrese la contraseña
12. Identifique el directorio en que se encuentra con el comando → `pwd`
13. Diríjase a la raíz con el comando → `cd /`
14. Diríjase al directorio del proyecto con el siguiente comando → `cd /nombre_del_directorio_del_proyecto/`
15. Actualice el proyecto con el siguiente comando → `git pull origin rama_donde_se_encuentra`
16. Ingrese el usuario y contraseña

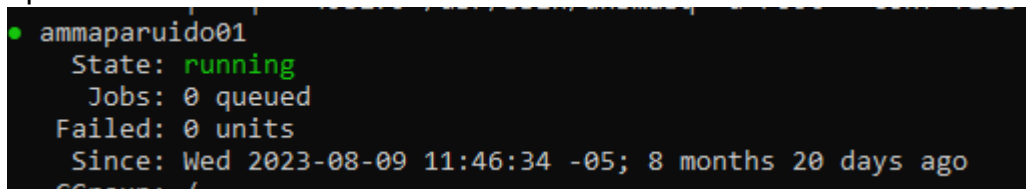
**Nota:** Recuerde tener a mano el usuario y contraseña del repositorio, tener presente que la contraseña corresponde a la llave de seguridad del proyecto en el repositorio.

## 5. CONFIGURACIÓN

Para su correcto funcionamiento es importante que verifique si el servidor cuenta con todos los lenguajes descritos en el capítulo 2 de lo contrario realice la debida instalación de los lenguajes y tecnologías requeridas. Las versiones de las tecnologías actualmente instaladas se pueden evidencia en el Anexo 4.

Para verificar, realice el siguiente protocolo de comprobación:

1. Abra la consola Windows [cmd] o Linux [terminal]
2. Reinicie el servidor con el siguiente comando → `sudo systemctl restart apache2.service`
3. Verifique el estado del servidor con el comando → `sudo systemctl status apache2`



```
● ammaparuido01
   State: running
     Jobs: 0 queued
  Failed: 0 units
   Since: Wed 2023-08-09 11:46:34 -05; 8 months 20 days ago
  CGroup: /
```

Figura 2. Estado del servidor

4. Diríjase a la raíz con el comando → `cd /`

**No se recomienda realizar actualización de versiones de los lenguajes o librerías**, sin antes revisar la versión actual del sistema y verificar los microservicios asociados. Las actualizaciones de sistemas pueden incurrir en problemas de funcionalidad, por lo que es recomendable realizar pruebas antes de realizar el despliegue.

### 5.1. Creación del sistema de gestión

Para la configuración del sistema de monitoreo y gestión, siga las siguientes instrucciones:

1. Ubíquese en el directorio principal usando el comando → `cd /`
2. Diríjase al directorio → `cd etc/supervisor/conf.d/`
3. Cree un archivo `.conf`
4. Pegue el código del Anexo 2

### 5.2. Creación del entorno virtual

Para poder realizar la instalación de paquetes y librerías necesarias para el funcionamiento del proyecto, se requiere crear un entorno de trabajo. A continuación, se expone como se debe crear el entorno virtual y su activación.

1. Ubíquese en el directorio del proyecto
2. Instale el paquete ingresando el siguiente comando → `pip install virtualenv`
3. Ingrese la siguiente instrucción → `python -m venv nombre_del_entorno`
4. Verifique que cuenta con el archivo → `requirements.txt`

5. Instale las librerías necesarias en el proyecto ingresando la siguiente instrucción → `pip install -r requirements.txt`

Nota: Si no se cuenta con el archivo requirements lo puede generar ingresando el comando → `pip freeze > requirements.txt`, o generando uno de forma manual.

### 5.3.Activación del entorno virtual

La activación se realizará haciendo uso del supervisor, para cumplir con este objetivo siga las siguientes instrucciones.

1. Cree un archivo de configuración .sh
2. Pegue el código del Anexo 3
3. Seleccione el puerto donde desea desplegar
4. Guarde y verifique

Nota: Si no se activa el entorno virtual, el proyecto puede presentar varios fallos problemas, debido a la falta de componentes, que no son detectados.

## ANEXO

### Anexo 1 Credenciales de acceso

Parámetro	Valor
Usuario	maparuido
Contraseña	Amva.2022

### Anexo 2. Código de administración

```
[program:amva_site]
command=/amva_site/server.sh
autostart=true
autorestart=true
stderr_logfile=/amva_site/logs/err.log
stdout_logfile=/amva_site/logs/out.log
environment=LANG=en_US.UTF-8,LC_ALL=en_US.UTF-8
```

### Anexo 3. Código de configuración del despliegue

```
#!/bin/bash
DJANGODIR=$(dirname "$0")
DJANGO_SETTINGS_MODULE=webgisApp.settings
cd $DJANGODIR
source venv/bin/activate
export DJANGO_SETTINGS_MODULE=$DJANGO_SETTINGS_MODULE
exec python3 geotrendstest/manage.py runserver 0:port
```

### Anexo 4. Versiones de tecnologías instaladas

Lenguaje	Versión
APACHE	2.4.52
PYTHON	3.10.12
MySQL	8.0.36
BASH	5.1.16