



# Instalación de GeoExpress Portal Entre Ríos



# Índice

<b>Pre-requisitos Base de Datos</b>	<b>3</b>
<b>Instalación Base de Datos</b>	<b>4</b>
<b>Pre-requisitos GeoExpress Portal</b>	<b>5</b>
<b>Preparación de los archivos GeoExpress Portal</b>	<b>6</b>
<b>Preparación de la base de datos</b>	<b>7</b>
<b>Instalación GeoExpress Portal</b>	<b>8</b>

# Pre-requisitos Base de Datos

1. Entramos al servidor por SSH con nuestro cliente favorito (recomendación personal: [Termius](#))
2. Cambiamos a usuario administrador con `sudo su`
3. Actualizamos los paquetes del equipo con `apt-get update && apt-get upgrade`
4. Instalamos docker (se recomienda usar el script get docker con `curl -fsSL https://get.docker.com/ | sh`)
5. Cambiar entorno de ejecución de docker (opcional) (el entorno de ejecución de docker por defecto es `/var/lib`, en este caso lo cambiaremos a `/datos`)
  1. Creamos la carpeta docker con `mkdir /datos/docker`
  2. Abrimos el archivo de configuración `vim /lib/systemd/system/docker.service`
  3. Añadimos el parametro `--data-root=/datos/docker` en la variable `ExecStart`
  4. Recargamos systemctl `systemctl daemon-reload`
  5. Reiniciamos docker `service docker restart`

# Instalación Base de Datos

2. Creamos un directorio para nuestra configuración `mkdir /datos/postgres`
3. Entramos al directorio creado `cd /datos/postgres`
4. Creamos un archivo de configuración `vim compose.yml` con el contenido descrito abajo
5. Levantamos el servicio con `docker compose up -d`
6. Listo

```
services:
  postgres:
    container_name: db
    image: postgis/postgis:15-master
    environment:
      POSTGRES_USER: postgres
      POSTGRES_PASSWORD: *****
      PGDATA: /data/postgres
    volumes:
      - postgres:/data/postgres
    ports:
      - "5432:5432"
    restart: unless-stopped

volumes:
  postgres:
```

# Pre-requisitos GeoExpress Portal

1. Instalar docker
  1. Entramos al servidor por SSH con nuestro cliente favorito (recomendación personal: [Termius](#))
  2. Cambiamos a usuario administrador con `sudo su`
  3. Actualizamos los paquetes del equipo con `apt-get update && apt-get upgrade`
  4. Instalamos docker (se recomienda usar el script get docker con `curl -fsSL https://get.docker.com/ | sh`)
2. Cambiar entorno de ejecución de docker (opcional) (el entorno de ejecución de docker por defecto es `/var/lib`, en este caso lo cambiaremos a `/datos`)
  1. Creamos la carpeta docker con `mkdir /datos/docker`
  2. Abrimos el archivo de configuración `vim /lib/systemd/system/docker.service`
  3. Añadimos el parámetro `--data-root=/datos/docker` en la variable `ExecStart`
  4. Recargamos systemctl `systemctl daemon-reload`
  5. Reiniciamos docker `service docker restart`

# Preparación de los archivos GeoExpress Portal

2. Nos movemos al directorio de trabajo `cd /datos`
3. Extraemos el ZIP entregado por kan en este directorio
4. Nos movemos al directorio clonado `cd geonode`
5. Comentamos todas las líneas del archivo `wait-for-databases.sh` (debido a usar una database externa)
6. Comentamos todas las menciones de DB en el archivo `docker-compose.yml` (debido a usar una database externa)
7. Generamos un archivo de variables de entorno `python3 create-envfile.py -hn geoservicios.entrerios.gov.ar`
8. Editaremos el archivo `.env` generado cambiando `db` en las variables `DATABASE_HOST`, `DATABASE_URL` y `GEODATABASE_URL` por la ip de el servidor de nuestra base de datos (debido a usar una database externa)
9. Editaremos el archivo `.env` generado cambiando la variable `POSTGRES_PASSWORD` por la contraseña del usuario postgres de nuestra base de datos (debido a usar una database externa)
10. Tomaremos nota de las variables `GEONODE_DATABASE_PASSWORD` y `GEONODE_GEODATABASE_PASSWORD` para usarlas en el siguiente apartado (debido a usar una database externa)
11. Cambiaremos el protocolo de `http://` a `https://` en las variables `SITEURL`, `NGINX_BASE_URL`, `GEOSERVER_WEB_UI_LOCATION` y `GEOSERVER_PUBLIC_LOCATION` (debido a usar un proxy reverso)
12. Tomaremos nota de las variables `GEOSERVER_ADMIN_PASSWORD` y `ADMIN_PASSWORD`, estas serán nuestras credenciales de administrador de geoserver y geonode respectivamente

# Preparación de la base de datos

1. Entramos al servidor por SSH con nuestro cliente favorito (recomendación personal: [Termius](#))
2. Cambiamos a usuario administrador con `sudo su`
3. Entramos al postgres que esta en el contenedor `docker exec -it db psql -U postgres`
4. Creamos usuario `geonode` con la contraseña extraída de la variable `GEONODE_DATABASE_PASSWORD`.  
`CREATE ROLE geonode WITH PASSWORD '*****';`
5. Le permitimos iniciar sesión `ALTER ROLE geonode WITH login;`
6. Creamos una base de datos con su nombre y bajo su control `CREATE DATABASE geonode WITH OWNER geonode;`
7. Nos conectamos a la base de datos `\c geonode`
8. Creamos la extensión postgis `CREATE EXTENSION postgis;`
9. Repetimos pasos 4 a 8 pero con el usuario `geonode_data` y la contraseña extraída de la variable `GEONODE_GEODATABASE_PASSWORD`
10. Salimos de la base de datos `\q`

# Instalación GeoExpress Portal

1. Buildeos y levantamos los contenedores con `docker compose up -d --build` (el proceso lleva un buen rato, se recomienda aprovechar para prepararse un café, es posible que recursos deficientes del sistema causen que después del buildeo el contenedor "django4geonode" intente iniciar antes de que el resto de contenedores estén listos y aparentemente el proceso haya fallado, si esto ocurre se recomienda esperar unos minutos a que el resto de contenedores inicien correctamente y luego iniciar manualmente el contenedor con `docker compose up -d django`)
2. Modificamos nginx para rutear mapstore, abrimos el archivo `vim /datos/docker/volumes/geonode-nginxconfd/_data/sites-enabled/geonode.conf.envsubst` y le añadimos el contenido de abajo antes de `# GeoNode`
3. Reiniciamos nginx `docker restart nginx4geonode`
4. Listo

```
# Mapstore
location /static/mapstore {
    set $upstream mapstore:8081;
    proxy_set_header Host 0.0.0.0;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto $scheme;
    proxy_pass http://$upstream;
}
```