

Guía de configuración de la base de datos en Ubuntu

Introducción

Esta guía describe cómo instalar y configurar PostgreSQL en Ubuntu para el proyecto MVP. PostgreSQL es un sistema gestor de bases de datos relacional de código abierto, muy utilizado y con conectores para prácticamente cualquier lenguaje de programación. La base de datos propuesta en el archivo `database_schema.sql` está normalizada en tablas independientes y emplea claves foráneas para garantizar la integridad referencial. Este enfoque reduce la redundancia y mejora la integridad de los datos en.wikipedia.org.

Requisitos

- Un sistema Ubuntu 22.04, 24.04 o superior con usuario con privilegios **sudo**.
- Conexión a internet para descargar paquetes.
- Opcional: tener a mano el script SQL `database_schema.sql` que contiene la estructura de la base de datos.

Paso 1: Instalar PostgreSQL

Ubuntu incluye PostgreSQL en sus repositorios y se puede instalar usando [apt](https://apt.postgresql.org) [postgresql.org](https://apt.postgresql.org). Para la mayoría de proyectos es suficiente instalar la versión predeterminada:

```
sudo apt update
sudo apt install postgresql
```

Este comando instala el servidor y el cliente y configura el servicio con valores seguros por defecto documentation.ubuntu.com. Si necesita una versión más reciente (por ejemplo, PostgreSQL 18) use el repositorio oficial PGDG que proporciona paquetes actualizados [postgresql.org](https://apt.postgresql.org):

```
sudo apt install -y postgresql-common
sudo /usr/share/postgresql-common/pgdg/apt.postgresql.org.sh
sudo apt update
```

```
sudo apt install postgresql-18    # reemplace 18 con la versión deseada
```

Paso 2: Configurar PostgreSQL

PostgreSQL admite distintos métodos de autenticación; en Ubuntu, **peer** se usa para conexiones locales y **scram-sha-256** es el valor por defecto para conexiones de red documentation.ubuntu.com. Si su aplicación se ejecutará en el mismo servidor puede dejar la configuración predeterminada. Para aceptar conexiones desde otras máquinas edite `/etc/postgresql/<versión>/main/postgresql.conf` y cambie:

```
#listen_addresses = 'localhost'
```

por

```
listen_addresses = '*'
```

Esto permite escuchar en todas las interfaces de red documentation.ubuntu.com. Después reinicie el servicio para aplicar los cambios:

```
sudo systemctl restart postgresql.service
```

Paso 3: Configurar la contraseña del superusuario

Después de instalar PostgreSQL, cree una contraseña segura para el usuario **postgres**. Conéctese a la base de datos **template1** y ejecute un comando **ALTER USER** documentation.ubuntu.com:

```
sudo -u postgres psql template1
ALTER USER postgres WITH ENCRYPTED PASSWORD 'contraseña_segura';
\q
```

Paso 4: Crear un rol y una base de datos para el proyecto

Por buenas prácticas conviene crear un rol dedicado para la aplicación y una base de datos que sea propiedad de ese rol:

```
sudo -u postgres createuser --pwprompt mvp_user
sudo -u postgres createdb --owner=mvp_user sistema_mvp
```

Alternativamente, desde `psql` puede ejecutar:

```
CREATE ROLE mvp_user WITH LOGIN ENCRYPTED PASSWORD 'contraseña_rol';  
CREATE DATABASE sistema_mvp OWNER mvp_user ENCODING 'UTF8';
```

El uso de **UTF-8** como codificación permite almacenar texto en varios idiomas y evita problemas de compatibilidad [geeksforgeeks.org](https://www.geeksforgeeks.org/).

Paso 5: Ejecutar el script de creación de tablas

Copie el archivo `database_schema.sql` en el servidor y ejecute el script conectado a la base de datos recién creada:

```
sudo -u postgres psql -d sistema_mvp -f /ruta/al/database_schema.sql
```

Esto crea todas las tablas, índices y relaciones necesarias para el MVP.

Paso 6: Probar la conexión desde su lenguaje de programación

PostgreSQL ofrece controladores para muchos lenguajes (por ejemplo, **psycopg2/asyncpg** para Python, **pg** para Node.js, **Npgsql** para .NET y **pgjdbc** para Java). Configure la cadena de conexión con el nombre de la base de datos, el usuario y la contraseña:

```
postgresql://mvp_user:contraseña_rol@localhost:5432/sistema_mvp
```

La aplicación podrá ejecutar consultas SQL estándar sin cambios, pues PostgreSQL implementa la especificación SQL y ofrece clientes para prácticamente cualquier lenguaje.

Buenas prácticas de diseño de base de datos

- **Normalización:** estructure la base en tablas independientes y use claves foráneas para garantizar integridad. La normalización reduce la redundancia y mejora la integridad de los datos en.wikipedia.org.
- **Índices:** añada índices a las columnas utilizadas en condiciones de búsqueda y relaciones (por ejemplo, claves foráneas). El script `database_schema.sql` incluye índices para mejorar el rendimiento.

- **Tipos adecuados:** defina campos numéricos con precisión y escala adecuadas (`DECIMAL(10,2)` para montos monetarios) y use `TIMESTAMP` para fechas y horas.
- **Codificación UTF-8:** asegure que la base de datos utiliza una codificación Unicode para admitir caracteres internacionales [geeksforgeeks.org](https://www.geeksforgeeks.org/).
- **Control de accesos:** limite los privilegios de cada rol al mínimo necesario; utilice vistas o funciones almacenadas para encapsular lógica sensible.
- **RespalDOS y mantenimiento:** programe copias de seguridad regulares y verifique la restauración. Considere configurar el parámetro `wal_level` a `replica` y usar herramientas como `pg_basebackup` si necesita replicación o alta disponibilidad.

Referencias

- Las instrucciones de instalación oficiales de PostgreSQL para Ubuntu indican que PostgreSQL se puede instalar desde los repositorios de Ubuntu mediante `apt install postgresql` [postgresql.org](https://www.postgresql.org/) y que se puede configurar un repositorio oficial para versiones más recientes [postgresql.org](https://www.postgresql.org/).
- La documentación de Ubuntu señala que el servicio se configura con valores viables por defecto y muestra cómo establecer la contraseña del usuario [postgresdocumentation.ubuntu.com](https://documentation.ubuntu.com/postgres/).
- La normalización en bases de datos se define como el proceso de estructurar una base relacional para reducir la redundancia y mejorar la integridad en.wikipedia.org.