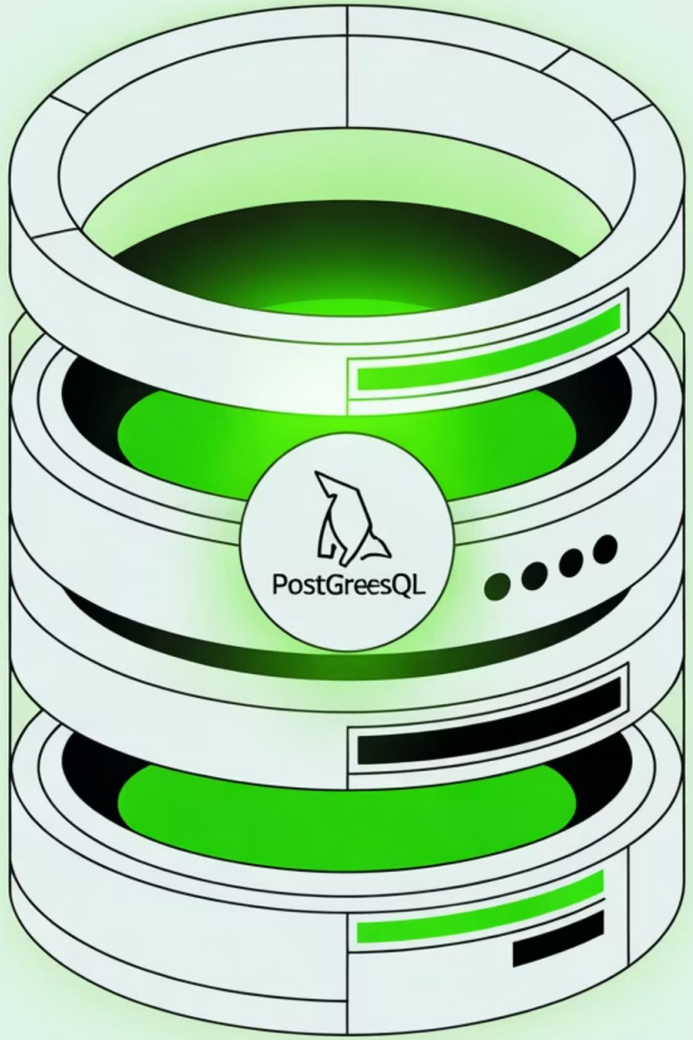
Instalación y Configuración de PostgreSQL

Una guía técnica completa para instalar PostgreSQL, crear bases de datos y configurar el entorno de desarrollo. Aprenderás desde la instalación básica hasta la creación de esquemas y tablas, estableciendo las bases para desarrollar aplicaciones robustas con PostgreSQL.



Descarga e Instalación de PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos relacional de código abierto, conocido por su fiabilidad y rendimiento. La instalación es compatible con Windows, macOS y Linux, proporcionando flexibilidad para cualquier entorno de desarrollo.

Pasos de instalación

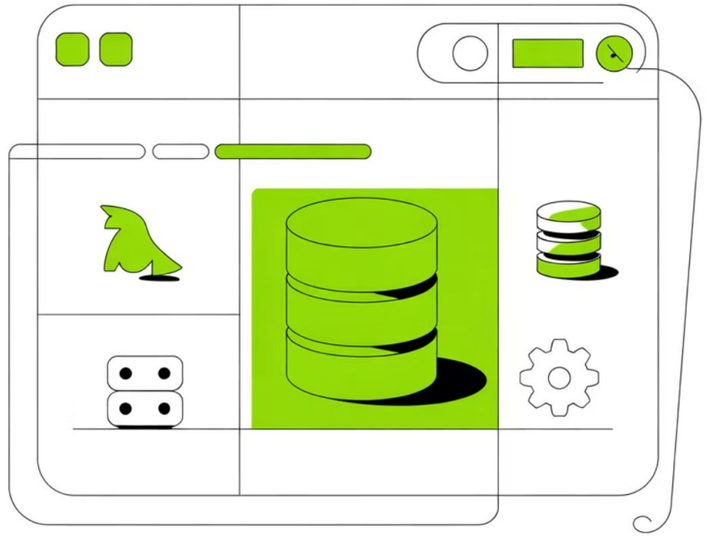
Accede a postgresql.org y descarga el instalador correspondiente a tu sistema operativo

2. Ejecuta el instalador y sigue el asistente de configuración

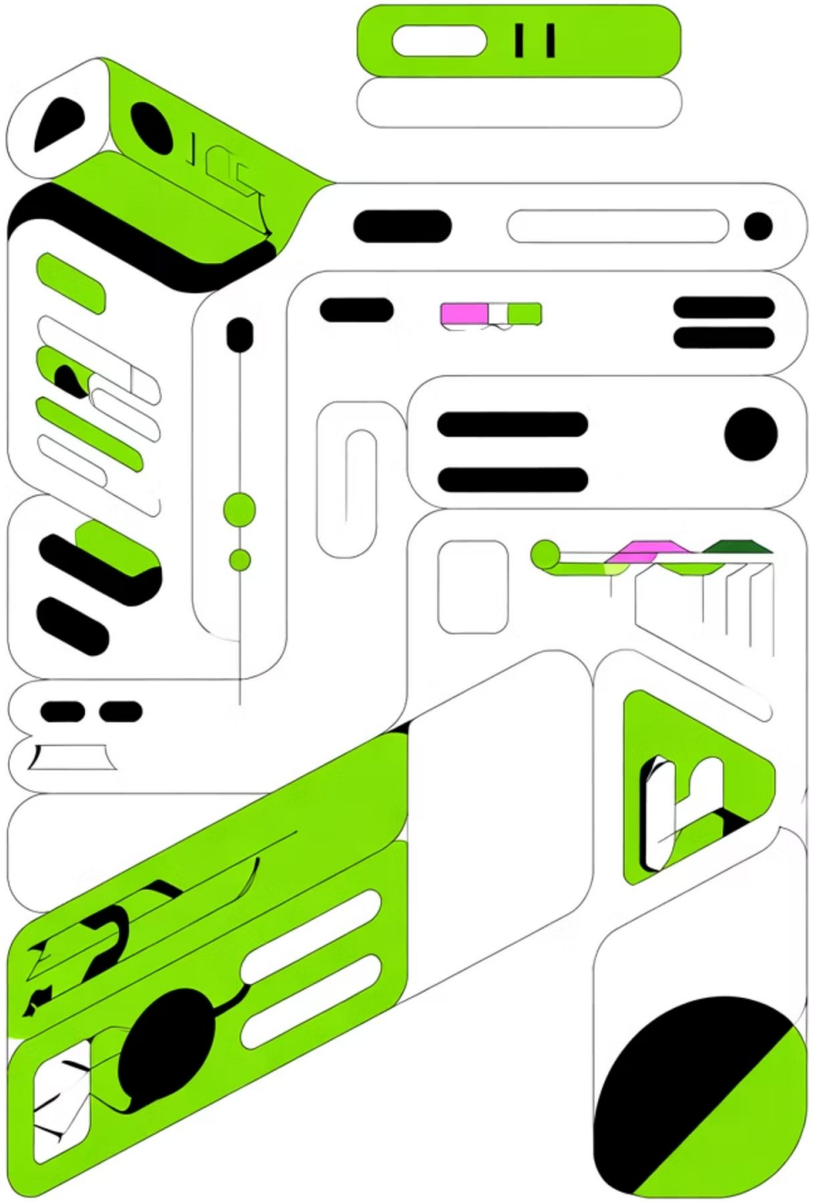
Define una contraseña robusta para el superusuario **postgres**

Configura el puerto de conexión (por defecto: **5432**)

Opcionalmente, instala **pgAdmin** como interfaz gráfica de administración



Recomendación: Anota la contraseña del superusuario postgres en un lugar seguro. La necesitarás para tareas administrativas posteriores.



Instalación de DBeaver

DBeaver es una herramienta universal de administración de bases de datos que proporciona una interfaz gráfica intuitiva para trabajar con PostgreSQL. Es multiplataforma, gratuita y ofrece funcionalidades avanzadas como autocompletado SQL, visualización de esquemas y gestión de conexiones múltiples.

Descarga DBeaver

Visita dbeaver.io y descarga la versión Community Edition para tu sistema operativo. Es completamente gratuita y de código abierto.

Configuración inicial

Al primer arranque, DBeaver detectará automáticamente PostgreSQL instalado. Configura las conexiones mediante el asistente de nueva conexión.

Ventajas clave

Editor SQL avanzado, explorador de objetos, diseñador ER, exportación de datos y soporte para múltiples SGBD desde una única interfaz.

PostgreSQL

Creación de Roles y Bases de Datos

En PostgreSQL, los roles representan usuarios o grupos de usuarios con permisos específicos. Es fundamental crear roles dedicados para cada aplicación, evitando el uso directo del superusuario postgres por razones de seguridad y auditoría.

Crear rol de aplicación

Define un rol específico con permisos de inicio de sesión y una contraseña segura. Este rol gestionará la base de datos de tu aplicación.

Crear base de datos

Establece una nueva base de datos asignando como propietario el rol previamente creado, garantizando control total sobre sus objetos.

Asignar privilegios

Otorga todos los privilegios necesarios al rol sobre la base de datos, permitiendo operaciones de lectura, escritura y modificación de estructura.

PostgreSQL

Código SQL: Crear Rol y Base de Datos

Ejecuta estos comandos SQL en pgAdmin o DBeaver conectado como superusuario **postgres**. Ajusta la contraseña según tus políticas de seguridad organizacionales.

```
-- Crear rol (usuario de aplicación)
CREATE ROLE appuser
WITH LOGIN
PASSWORD 'app_pass_123';

-- Crear base de datos con propietario
CREATE DATABASE biblioteca_db
OWNER appuser;

-- Otorgar todos los privilegios
GRANT ALL PRIVILEGES
ON DATABASE biblioteca_db
```

Importante: Tras ejecutar estos comandos, cierra la conexión actual y conéctate nuevamente a **biblioteca_db** utilizando las credenciales del usuario **appuser**. Esto asegura que trabajas con los permisos correctos desde el inicio.

PostgreSQL

Arquitectura del Modelo de Datos

Antes de crear tablas, es fundamental diseñar un esquema lógico que organice los objetos de la base de datos. El modelo que implementaremos incluye dos entidades principales: **usuarios** y **clientes**, representando actores clave en el sistema de biblioteca.

Tabla: usuarios

id: Identificador único (clave primaria)

nombre: Nombre completo del usuario

email: Correo electrónico único

activo: Estado de activación booleano

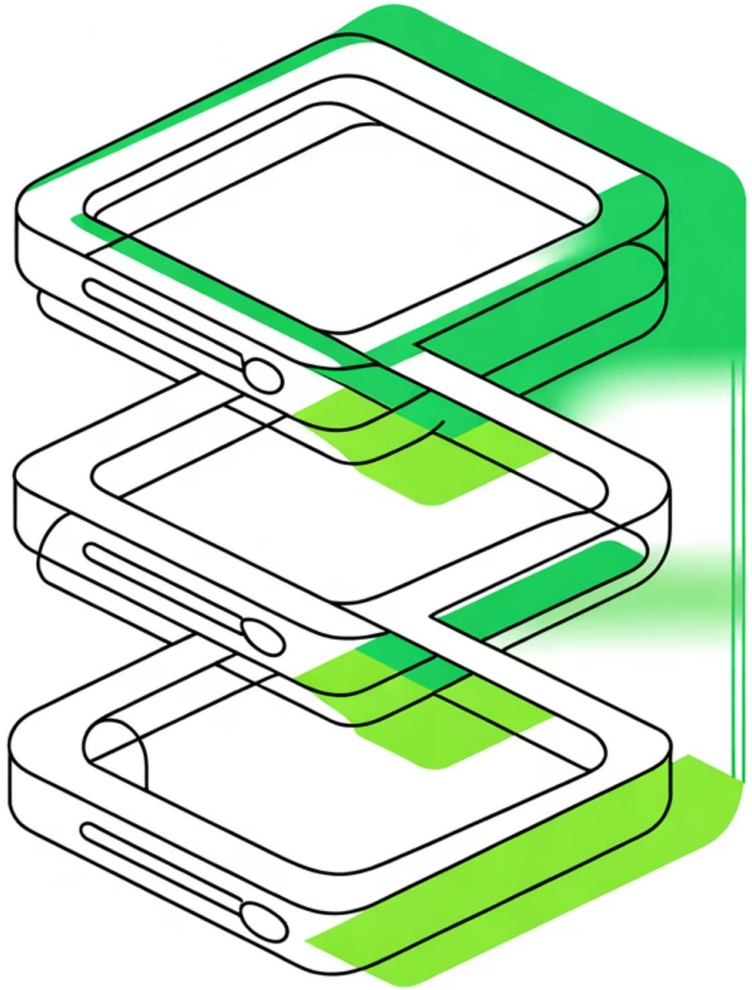
Tabla: clientes

id: Identificador único (clave primaria)

nombre: Nombre del cliente

telefono: Número de contacto

deuda: Saldo pendiente (numérico con precisión)



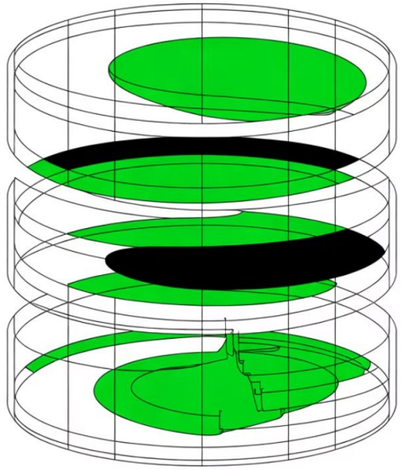
Creación del Esquema de Base de Datos

Los esquemas en PostgreSQL organizan objetos de base de datos en grupos lógicos, similar a carpetas en un sistema de archivos. Proporcionan aislamiento, seguridad y facilitan la gestión de múltiples aplicaciones en una misma base de datos.

```
 -- Crear esquema personalizado
CREATE SCHEMA IF NOT EXISTS app
AUTHORIZATION appuser;

-- Configurar ruta de búsqueda
SET search_path TO app;
```

El comando CREATE SCHEMA establece un espacio de nombres dedicado. Al configurar el search_path, PostgreSQL buscará objetos primero en el esquema **app**, simplificando las consultas al evitar prefijos constantes.



PostgreSQL

Creación de Tabla: Usuarios

La tabla **usuarios** almacena información básica de los usuarios del sistema. Implementa restricciones de integridad mediante claves primarias y restricciones de unicidad, asegurando la consistencia de los datos.

```
-- Tabla usuarios con restricciones
CREATE TABLE IF NOT EXISTS usuarios (
    id INTEGER PRIMARY KEY,
    nombre VARCHAR(120) NOT NULL,
    email VARCHAR(200) NOT NULL UNIQUE,
    activo BOOLEAN NOT NULL DEFAULT TRUE
);
```

PRIMARY KEY

La columna **id** identifica de forma única cada registro. PostgreSQL crea automáticamente un índice para optimizar búsquedas por este campo.

NOT NULL

Garantiza que campos críticos como **nombre** y **email** siempre contengan valores, previniendo inconsistencias en datos obligatorios.

UNIQUE

La restricción sobre **email** previene duplicados, asegurando que cada dirección de correo pertenezca a un único usuario.

DEFAULT

El campo **activo** se inicializa automáticamente como TRUE, simplificando inserciones y proporcionando un estado predeterminado consistente.

PostgreSQL

Creación de Tabla: Clientes

La tabla **clientes** registra información financiera y de contacto. Utiliza el tipo de dato NUMERIC para la columna deuda, garantizando precisión decimal en operaciones monetarias sin errores de redondeo típicos de tipos flotantes.

```
-- Tabla clientes con campos numéricos precisos
CREATE TABLE IF NOT EXISTS clientes (
    id INTEGER PRIMARY KEY,
    nombre VARCHAR(120) NOT NULL,
    telefono VARCHAR(40),
    deuda NUMERIC(12,2) NOT NULL DEFAULT 0
);
```

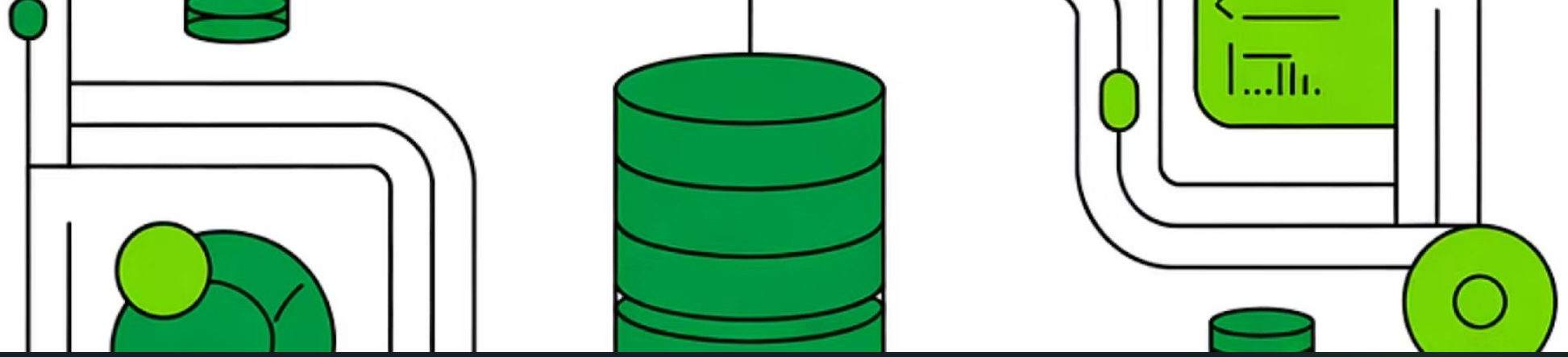
Tipos de datos VARCHAR

Los campos de texto utilizan **VARCHAR** con límites definidos: 120 caracteres para nombres (suficiente para nombres compuestos internacionales) y 40 para teléfonos (permite formatos con prefijos y extensiones).

Tipo NUMERIC para precisión

NUMERIC(12,2) permite valores hasta 9.999.999.999,99 con exactitud decimal absoluta. El valor predeterminado **0** inicializa nuevos clientes sin deuda pendiente.

PostgreSQL



Próximos Pasos

Has completado la configuración básica de PostgreSQL y el modelo de datos inicial. A partir de aquí, tu entorno está listo para desarrollo avanzado y operaciones de producción.

Inserción de datos

Utiliza comandos INSERT para poblar las tablas con datos de prueba y validar las restricciones implementadas.

Relaciones entre tablas

Define claves foráneas (FOREIGN KEY) para establecer vínculos entre usuarios, clientes y otras entidades futuras del sistema.

Optimización

Crea índices adicionales, vistas materializadas y procedimientos almacenados para mejorar el rendimiento de consultas frecuentes.

PostgreSQL