Instalar nodcodb

**Instala Node.js para Windows**  
Descarga el instalador desde:  
https://nodejs.org/en/download/  
y sigue los pasos para instalarlo.

node-v22.18.0-x64.msi con esta versión lo probe.

Powershell

Invoke-WebRequest -Uri "https://github.com/nocodb/nocodb/releases/latest/download/Noco-win-x64.exe" -OutFile "Noco-win-x64.exe"

Start-Process -FilePath ".\Noco-win-x64.exe"

**Cómo clonar el repositorio de NocoDB con GitHub Desktop**

**1. Abre GitHub Desktop**

Si no lo tienes abierto, búscalo en el menú inicio o desde el acceso directo.

**2. En el menú, haz clic en:**

File → Clone repository...

**3. Selecciona la pestaña URL (arriba)**

Aquí es donde pegarás el enlace del repositorio de NocoDB.

**4. Pega la URL del repositorio de NocoDB**

En el campo **Repository URL**, escribe o copia:

https://github.com/nocodb/nocodb.git

En **Local Path**, elige la carpeta donde se guardará el proyecto en tu PC, por ejemplo:

C:\Users\TuUsuario\Documents\Proyectos\nocodb

**5. Haz clic en Clone**

GitHub Desktop descargará todo el código fuente del repositorio a tu máquina.

**6. Abre el proyecto en Visual Studio Code**

Cuando termine de clonar, puedes hacer clic en:

Repository → Open in Visual Studio Code

O bien, abre VS Code y desde allí abre la carpeta clonada.

**✅ A partir de aquí puedes seguir los pasos previos:**

Ejecuta en la terminal de VS Code:

npm install

Luego:

npm run dev

Y ¡listo! Ya estarás ejecutando y modificando tu propia copia del código fuente de NocoDB.

Ahorita manda este error npm error code ECONNRESET

npm error errno ECONNRESET

npm error network request to https://172.16.19.1/UserCheck/PortalMain?IID=\*\*\*&origUrl=aHR0cHM6Ly9jZG4uc2hlZXRqcy5jb20veGxzeC0wLjIwLjMveGxzeC0wLjIwLjMudGd6 failed, reason: socket hang up

npm error network This is a problem related to network connectivity.

npm error network In most cases you are behind a proxy or have bad network settings.

npm error network

npm error network If you are behind a proxy, please make sure that the

npm error network 'proxy' config is set properly. See: 'npm help config'

npm error A complete log of this run can be found in: C:\Users\mgomez\AppData\Local\npm-cache\\_logs\2025-08-14T20\_03\_51\_102Z-debug-0.log

Sí! Puedes usar **NocoDB en tu red local** y permitir que varios usuarios accedan desde sus computadoras, **como si fuera una app web interna**. Aquí te explico cómo hacerlo paso a paso.

**✅ ¿Qué necesitas?**

Que tu PC esté conectada a una red local (Wi-Fi o LAN).

Que NocoDB esté ejecutándose en esa PC.

Saber la **IP local de tu máquina** (donde corre NocoDB).

Permitir el acceso en el **firewall de Windows** (si aplica).

**🛠️ Paso 1: Obtener tu IP local**

En tu PC (donde corre NocoDB), abre PowerShell o cmd y ejecuta:

ipconfig

Busca una línea que diga algo como:

Dirección IPv4. . . . . . . . . . . . : 192.168.1.10

Ese número (ej: 192.168.1.10) es tu IP local.

**🛠️ Paso 2: Ejecutar NocoDB escuchando en todas las IPs**

Por defecto, NocoDB solo escucha en localhost, lo que significa que **solo tú puedes acceder desde tu PC**.

Para que otros puedan entrar, debes ejecutar NocoDB así:

.\Noco-win-x64.exe --host 0.0.0.0 --port 8080

Esto le dice a NocoDB:  
➡️ “escucha conexiones desde cualquier equipo de la red”.

**🛡️ Paso 3: Permitir NocoDB en el firewall de Windows**

Cuando ejecutas NocoDB por primera vez con red habilitada, Windows te puede preguntar:

❓ ¿Permitir que esta app se comunique en redes privadas/públicas?

Haz clic en **“Permitir”** para redes **privadas** (como la red de tu casa o empresa).

Si no aparece ese mensaje, hazlo manualmente:

Abre el menú de inicio y escribe:  
**Firewall de Windows con seguridad avanzada**

En "Reglas de entrada", crea una nueva regla:

Tipo: **Puerto**

Puerto local: 8080

Permitir la conexión

Aplicar en redes privadas

Nombre: NocoDB

**👨‍👩‍👧‍👦 Paso 4: Conectarse desde otras PCs**

Desde otro equipo en la misma red:

Abre un navegador.

Escribe la IP de tu servidor + el puerto. Ejemplo:

http://192.168.1.10:8080

Debería cargar la interfaz de NocoDB.

**🔐 Opcional: Crear usuarios o controlar acceso**

Por defecto, el primer usuario que creaste es administrador. Luego puedes:

Crear más usuarios desde el panel de NocoDB.

Definir permisos por proyecto o tabla.

Habilitar login con email y contraseña para otros usuarios.

**✅ Resumen**

| **Tarea** | **Acción** |
| --- | --- |
| Obtener IP local | ipconfig |
| Ejecutar NocoDB en red | .\Noco-win-x64.exe --host 0.0.0.0 |
| Permitir en firewall | Abrir puerto 8080 |
| Acceder desde otra PC | http://[IP-de-tu-PC]:8080 |

Excelente pregunta. Si vas a usar **NocoDB solo como interfaz para conectarte a bases de datos PostgreSQL externas**, y **no para guardar datos internamente** (es decir, no usar la nocodb.db más que para configuración mínima), entonces **el rendimiento depende casi totalmente de PostgreSQL**, no de NocoDB.

Vamos por partes:

## ✅ ¿Qué hace NocoDB cuando te conectas a una base externa?

Cuando conectas NocoDB a una base PostgreSQL (por ejemplo, para visualizar o editar tablas externas), **NocoDB actúa como un puente**, no como un sistema que duplica los datos.

* No guarda los datos de tus tablas en nocodb.db
* Solo guarda configuración de:
  + Usuarios
  + Permisos
  + Dashboards creados
  + Vistas personalizadas
  + Metadatos del workspace

## 📌 ¿Afecta esto el rendimiento?

### ❌ ****NO afecta el rendimiento de tu base externa si:****

* Tu servidor PostgreSQL está bien dimensionado (CPU/RAM decentes)
* No haces operaciones masivas o consultas muy pesadas
* No hay 100+ usuarios concurrentes en NocoDB

### ✅ ****SÍ puede afectar si:****

* Muchas consultas complejas o filtros pesados son lanzados desde NocoDB
* El servidor donde corre PostgreSQL está sobrecargado
* Usas muchas vistas con uniones (JOIN) complejas

## 🔧 ¿Y si nocodb.db está en uso?

Aunque nocodb.db (SQLite) siga usándose para la parte interna de NocoDB, **eso no interfiere con la base externa**.

SQLite se usa para:

| **Uso** | **¿Impacta rendimiento externo?** |
| --- | --- |
| Configuración de usuarios | ❌ No |
| Vistas creadas en la app | ❌ No |
| Control de acceso | ❌ No |
| Datos reales externos | ✅ Están en tu PostgreSQL |

## 🛠 Recomendación si te preocupa el rendimiento

1. **Sigue usando SQLite (nocodb.db) para la configuración** si todo está estable
2. Asegúrate de que tu **PostgreSQL esté optimizado**:
   * Indexes bien definidos
   * Buen uso de recursos (memoria/cache)
   * Logs de rendimiento activados para consultas lentas
3. **Evita usar NocoDB como cliente de reporting pesado** (usa Metabase o Grafana si necesitas dashboards grandes)

## 🎯 Conclusión

✅ **No hay impacto significativo en rendimiento al usar NocoDB con bases PostgreSQL externas**, incluso si usas SQLite para la config interna.  
🔄 Si en algún momento necesitas alto rendimiento o seguridad avanzada, ahí sí valdría migrar la config interna a PostgreSQL también (totalmente posible más adelante).

¿Quieres una guía rápida para conectar NocoDB a tu base PostgreSQL externa (y tips para optimizar esa conexión)?

##### Tú dijiste:

la verdad el proposito es darle a un usuario el privelegio de cargar datos a las tablas de postgres ya las consultas o vistas de informacion la usaran los usuarios conectados desde excel y usaran tablas dinamicas

###### ChatGPT Plus

Perfecto, ahora sí todo encaja.  
Tu caso es **muy común y muy bien planteado**, y la solución que estás armando es **práctica, escalable y simple de mantener**. Vamos a validarla punto por punto.

## ✅ Tu objetivo real:

👉 Usar NocoDB como una interfaz sencilla para que ***uno o pocos usuarios carguen o modifiquen datos*** en una base PostgreSQL.  
👉 Luego, otros usuarios (por ejemplo, desde Excel) harán consultas con tablas dinámicas directamente desde PostgreSQL.

## 🎯 ¿Esta arquitectura es buena?

### ✔️ ****Sí, es una excelente arquitectura para:****

* Simplificar la entrada de datos (sin programar formularios)
* Restringir el acceso de escritura solo a ciertos usuarios
* Dejar la lectura libre vía Excel/ODBC para otros
* Mantener tu base PostgreSQL como el único “source of truth”

## 🧠 Ventajas de hacerlo así

| **Característica** | **Resultado ✔️** |
| --- | --- |
| NocoDB como frontend de carga | Usuario no necesita SQL |
| PostgreSQL como backend | Seguridad + rendimiento |
| SQLite (nocodb.db) solo para config | Simple y sin sobrecarga |
| Usuarios de Excel se conectan directo | Lectura sin pasar por NocoDB |
| No necesitas programar formularios | NocoDB lo genera automáticamente |

## 🔒 Seguridad recomendada

1. En PostgreSQL:
   * Crea un **usuario dedicado para NocoDB** con permisos de INSERT, UPDATE, DELETE solo en las tablas necesarias.
   * A los usuarios de Excel, dales acceso **solo de lectura (SELECT)**.
2. En NocoDB:
   * Asigna roles al usuario (Admin o Editor)
   * Limita qué tablas puede ver o modificar
3. Desde Excel:
   * Usa conexiones ODBC o PostgreSQL con credenciales de solo lectura.

## 💡 Tips técnicos

### En PostgreSQL

-- Usuario para NocoDB

CREATE USER nocodb\_user WITH PASSWORD 'superseguro';

GRANT CONNECT ON DATABASE tu\_base TO nocodb\_user;

GRANT USAGE ON SCHEMA public TO nocodb\_user;

GRANT SELECT, INSERT, UPDATE, DELETE ON ALL TABLES IN SCHEMA public TO nocodb\_user;

ALTER DEFAULT PRIVILEGES IN SCHEMA public

GRANT SELECT, INSERT, UPDATE, DELETE ON TABLES TO nocodb\_user;

-- Usuario solo lectura para Excel

CREATE USER excel\_user WITH PASSWORD 'sololectura';

GRANT CONNECT ON DATABASE tu\_base TO excel\_user;

GRANT USAGE ON SCHEMA public TO excel\_user;

GRANT SELECT ON ALL TABLES IN SCHEMA public TO excel\_user;

## 📦 En resumen

| **Elemento** | **Función** |
| --- | --- |
| 🧑‍💼 NocoDB | Carga y edición de datos |
| 🗃 PostgreSQL | Base central de datos |
| 📊 Excel | Consultas y reportes |
| 🔐 Seguridad | Control por roles |

👉 Es un flujo limpio, no depende de muchas herramientas, y puedes escalarlo cuando quieras (más usuarios, más tablas, migrar a NocoDB+PostgreSQL completo, etc.).