INFORME TÉCNICO

Validación de Entorno Local vs Backend en NGROK

Proyecto: SRNI – NLQuery (NL → SQL)
Fecha: 17 de Febrero 2026

1.Objetivo

Alinear el entorno local con la estructura real de base de datos utilizada en el entorno expuesto por ngrok, con el fin de validar correctamente el módulo NLQuery y las consultas SQL manuales.

2.Acciones realizadas en entorno local

✔ Restauración del dump actualizado

Se restauró correctamente el archivo:

`contratistas_sync_2026_02_16.dump`

Dentro del contenedor `postgres_local` del entorno dockerizado.

✔ Validación de conexión activa

Consulta ejecutada:

```
SELECT current_database() AS db,
       current_user AS usr,
       inet_server_addr() AS addr,
       inet_server_port() AS port
```

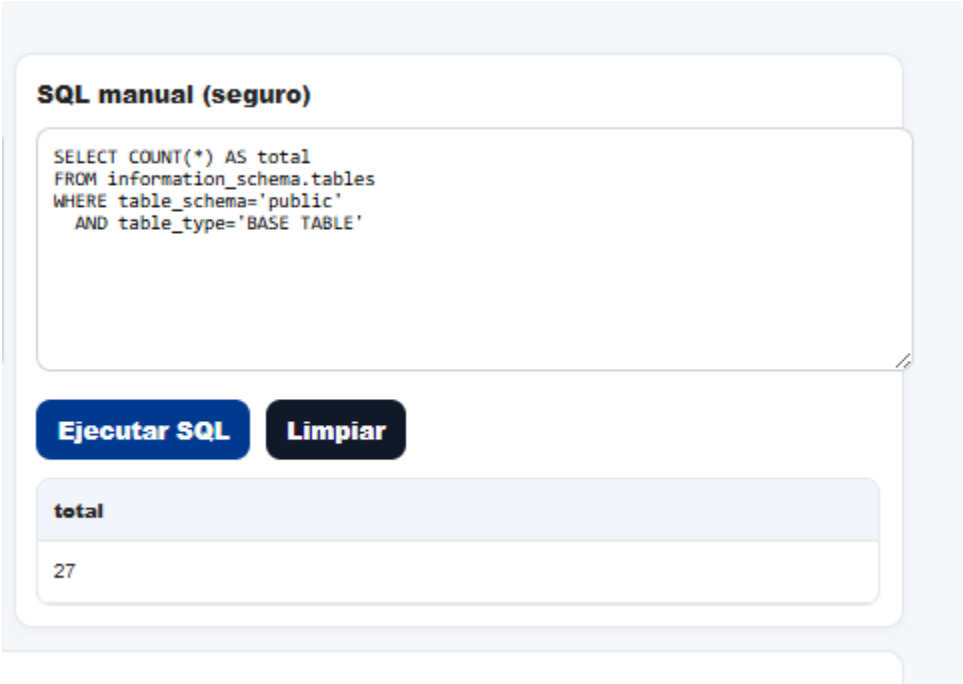
Confirma que el backend local está apuntando correctamente a la base restaurada.

3.Validación estructural de base de datos (LOCAL)

✔ Conteo total de tablas

```
SELECT COUNT(*) AS total
FROM information_schema.tables
WHERE table_schema='public'
  AND table_type='BASE TABLE'
```

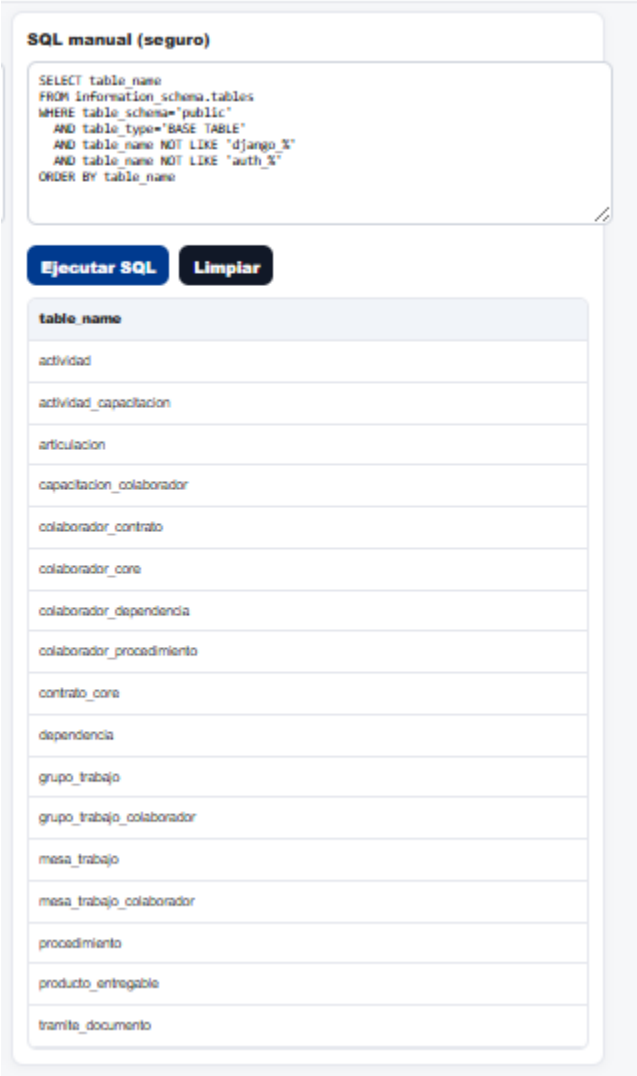
Resultado:



✔ Listado completo de tablas de negocio

```
SELECT table_name
FROM information_schema.tables
WHERE table_schema='public'
  AND table_type='BASE TABLE'
  AND table_name NOT LIKE 'django_%'
  AND table_name NOT LIKE 'auth_%'
ORDER BY table_name
```

Resultado incluye correctamente:



4.Validación de datos

Consultas ejecutadas para confirmar carga de información:

```
SELECT COUNT(*) FROM colaborador_core
```


```
SELECT COUNT(*) FROM contrato_core
```


```
SELECT COUNT(*) FROM actividad
```


```
SELECT COUNT(*) FROM articulación
```

Las consultas retornan resultados válidos, confirmando que el dump contiene estructura y datos.

5.Validación funcional NLQuery (LOCAL)

El endpoint `/api/nlquery/schema/` devuelve correctamente un JSON con las tablas permitidas según `SchemaLoader.ALLOWED_TABLES`.

Ejemplo de esquema devuelto:

```
{
  "actividad": {...},
  "articulacion": {...},
  "colaborador_contrato": {...},
  "procedimiento": {...}
}
```

Conclusión:

El módulo NLQuery está funcionando correctamente en entorno local con la base restaurada.

6.Diagnóstico del entorno expuesto por NGROK

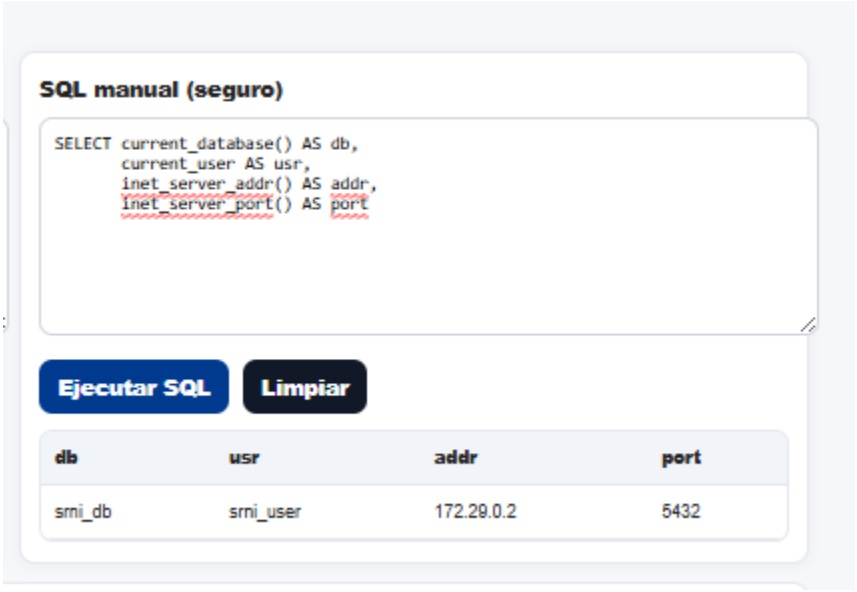
Validación ejecutada en:

```
https://srni-backend.ngrok.io
```

✔ Validación de conexión activa

Consulta ejecutada:

```
SELECT current_database() AS db,
       current_user AS usr,
       inet_server_addr() AS addr,
       inet_server_port() AS port;
```



✕ Conteo de tablas

```
SELECT COUNT(*)
FROM information_schema.tables
WHERE table_schema='public'
  AND table_type='BASE TABLE'
```

Resultado:

Las tablas corresponden únicamente a:

- auth_group

- auth_permission
- auth_user
- django_admin_log
- django_migrations
- etc.

No existen tablas de dominio como `colaborador_core`, `contrato_core`, `actividad`, etc.

❌ Resultado del endpoint Schema
`/api/nlquery/schema/`

Devuelve:

```
{ "schema": {} }
```

Esto es correcto desde el punto de vista lógico, ya que el `SchemaLoader` solo devuelve tablas incluidas en la whitelist, y actualmente esa base no contiene dichas tablas.

7.Conclusión Técnica

El código del módulo NLQuery y el SchemaLoader funcionan correctamente.

La diferencia entre local y ngrok no es de código ni de rama Git.

La diferencia es de conexión a base de datos.

El backend en ngrok está apuntando a una base distinta (o sin el dump restaurado).

8.Restricción por Seguridad (.env / gitignore)

Las variables de conexión están aisladas en `.env` (gitignore).

Desde mi rama no es posible:

- Verificar las credenciales activas del entorno ngrok

- Modificar el target de la base
- Confirmar el contenedor Postgres real al que está apuntando

El problema debe validarse desde el entorno donde corre ngrok.

9.Lo que debe revisarse en el entorno de NGROK

1. Confirmar variables:
    - DB_NAME
    - DB_USER
    - DB_PASSWORD
    - DB_HOST
    - DB_PORT
2. Confirmar que el backend esté apuntando al Postgres o donde se restauró el dump.
3. Si no está restaurado, ejecutar:

```
pg_restore -U <usuario> -d srni_db --clean --if-exists --no-owner --no-
privileges contratistas_sync_2026_02_16.dump
```

4. Reiniciar backend y validar nuevamente:

```
SELECT COUNT(*) FROM information_schema.tables WHERE
table_schema='public';
```

Debe pasar de 10 a ~27–28 tablas.

10.Estado Actual

✔ Entorno local: Correcto y validado
✔ Dump actualizado restaurado
✔ NLQuery funcionando
❗ Entorno ngrok: apuntando a base incorrecta o sin dump

11.Cierre

El desarrollo en mi rama está alineado con la estructura real del dump actualizado.
El módulo funciona correctamente.

Lo único pendiente es que el backend en ngrok apunte a la base correcta que contiene las tablas de dominio.

Una vez eso esté alineado, el schema dejará de aparecer vacío y las consultas funcionarán igual que en local.