

De Excel también se sale

Cómo aprendí a no preocuparme y amar a NetBox

Enrique Matías
<quique@unizar.es>



Universidad
Zaragoza

3 de julio de 2025

Si sólo sabes usar un martillo
todo te parece un clavo.

“AKA martillo de Maslow o martillo de oro.”

! Desventajas de usar Excel en redes

Problema con Excel	Consecuencias
Falta de estructura relacional	Dificultad para vincular dispositivos, IPs, racks, ubicaciones u otros elementos de red relacionados
Errores humanos frecuentes	Celdas mal editadas, fórmulas rotas, datos duplicados o inconsistentes
Colaboración limitada	Sobrescritura accidental entre múltiples usuarios o pérdida de cambios
Escalabilidad deficiente	Archivos pesados, lentitud, y mayor riesgo de corrupción con volúmenes altos de datos
Sin historial de cambios	Falta de trazabilidad: no se sabe quién cambió qué ni cuándo
Automatización limitada	Difícil integración con herramientas externas o flujos de trabajo automatizados
No es una fuente de verdad confiable	Posibilidad de múltiples versiones del mismo archivo, datos desactualizados o contradictorios

? ¿Qué es NetBox?

- NetBox es una aplicación web de código abierto para documentar y gestionar infraestructura de red (DCIM (gestión física) + IPAM (gestión de direcciones IP)).
- Fue creada originalmente por Jeremy Stretch en DigitalOcean en 2016 para resolver la falta de una “fuente de verdad” fiable y flexible en entornos de red complejos.
- Desde entonces, ha evolucionado con el impulso de una comunidad activa y organizada bajo la iniciativa `netbox-community`.
- En 2023 se anunció la creación de **NetBox Labs**, una empresa dedicada al desarrollo y soporte comercial de NetBox, manteniendo su núcleo open source.
- Está desarrollada con Django (framework web en Python) y utiliza PostgreSQL como base de datos.
- Su diseño modular permite ampliaciones mediante plugins, automatización vía API REST, y visualizaciones de topología.



Un proyecto abierto que crece con la comunidad y evoluciona hacia la



Excel vs NetBox (1 de 2)

Problema con Excel	Consecuencias	Cómo lo soluciona NetBox
Falta de estructura relacional	Difícil vincular dispositivos, IPs, racks	Modelo de datos interrelacionado y estructurado
Errores humanos frecuentes	Celdas mal editadas, datos inconsistentes	Validaciones automáticas y seguimiento de cambios
Colaboración limitada	Sobrescritura de datos entre usuarios	Interfaz multiusuario con permisos granulares

Excel vs NetBox (2 de 2)

Problema con Excel	Consecuencias	Cómo lo soluciona NetBox
Escalabilidad deficiente	Archivos lentos o corruptos con muchos datos	Backend con PostgreSQL y Redis para alto rendimiento
Sin historial de cambios	No se sabe quién modificó qué ni cuándo	Registro de auditoría completo por usuario y fecha
Automatización limitada	Difícil de integrar con otras herramientas	API RESTful para integraciones con Ansible, Terraform, etc.
No es una fuente de verdad confiable	Datos duplicados o desactualizados	Fuente centralizada y validada de información de red

- Python 3.7+
- PostgreSQL 10+
- Redis 4.0+
- Linux (Ubuntu, CentOS) o Docker

Componentes Principales

- **DCIM:** gestión de racks, ubicaciones, dispositivos, conexiones físicas.
- **IPAM:** gestión de IPs, subredes, VLANs y VRFs.
- **Virtualización:** clústeres, máquinas virtuales, controladores.
- **Proveedores y circuitos:** enlaces físicos/lógicos y contratos.
- **Cableado y energía:** conexiones eléctricas y de red.
- **Secretos:** credenciales y datos sensibles de forma segura.

✓ Beneficios clave de usar NetBox

- **Interfaz web moderna e intuitiva.**
- **Fuente de verdad confiable:** centraliza y valida toda la documentación de red.
- **Prevención de errores:** validaciones automáticas y seguimiento de cambios.
- **Colaboración eficiente:** entorno multiusuario con control de acceso granular.
- **Escalabilidad real:** rendimiento óptimo para grandes entornos gracias a PostgreSQL y Redis.
- **Automatización e integración:** API RESTful lista para integrarse con tus herramientas DevOps favoritas.
- **Flexibilidad y personalización:** plugins, campos personalizados y compatibilidad con Docker.



Transforma tu documentación en una herramienta estratégica.

- `netbox-topology-views`: genera mapas de red interactivos
- `nextbox-ui-plugin`: visualización avanzada con auto-layout e íconos
- `netbox-bgp-view`: visualización de sesiones BGP

 Muchos más en <https://netboxlabs.com/plugins/>.

Migración desde Excel a NetBox (1 de 2)

- **1. Limpieza de datos:** Revisa tu hoja de Excel y elimina duplicados, errores y columnas innecesarias.
- **2. Estructura compatible:** Asegúrate de que los campos coincidan con los modelos de NetBox (por ejemplo: nombre del dispositivo, dirección IP, ubicación).
- **3. Exportación a CSV:** Guarda tu hoja como archivo .csv con codificación UTF-8.
- **4. Validación previa:** Verifica que los valores coincidan con los datos existentes en NetBox (por ejemplo: nombres de sitios, roles, fabricantes).

 Puedes usar la función de “Exportar CSV” en NetBox para ver el formato esperado.

● **Opción 1: Interfaz Web**

- Ve al modelo deseado (por ejemplo, Dispositivos, IPs).
- Haz clic en `Import` y sube tu archivo CSV.
- Revisa los datos antes de confirmar la importación.

● **Opción 2: Scripts personalizados**

- Usa la API REST de NetBox para importar datos desde scripts en Python.
- Ideal para migraciones masivas o automatizadas.

● **Opción 3: Herramientas externas**

- Herramientas como Ansible, Terraform o NetBox Loader pueden facilitar la carga masiva.

✔ **Consejo: comienza con un conjunto pequeño de datos para validar el proceso.**

Scripts personalizados en NetBox (1 de 2)

- Permiten ejecutar lógica personalizada directamente desde la interfaz web de NetBox.
- Se escriben en Python y heredan de la clase `extras.scripts.Script`.
- Acceden directamente al modelo de datos de NetBox.
- Útiles para:
 - Crear dispositivos, interfaces o IPs en lote.
 - Validar consistencia de datos.
 - Automatizar tareas repetitivas.
- Se ejecutan desde el menú: Personalización > Scripts.

 Los scripts se almacenan en `/opt/netbox/netbox/scripts/`.

Ejemplo básico de script en Python

```
from extras.scripts import Script, StringVar
from dcim.models import Site

class CrearSitio(Script):
    nombre = StringVar(description="Nombre del sitio")

    def run(self, data, commit):
        Site.objects.create(
            name=data['nombre'],
            slug=data['nombre'].lower()
        )
        self.log_success(f"Sitio '{data['nombre']}' creado.")
```

- StringVar, IntegerVar, ObjectVar, etc. permiten entrada de datos desde la UI.
- El método run() contiene la lógica principal.
- El parámetro commit permite ejecutar en modo prueba.

- Un plugin es una aplicación Django empaquetada que se instala junto a NetBox.
 - Permite extender NetBox sin modificar su núcleo.
 - Puede incluir:
 - Modelos personalizados
 - Vistas y plantillas propias
 - API REST y GraphQL extendidas
 - Middleware, señales, navegación, etc.
 - Se activan desde el archivo `configuration.py` y se instalan con `pip`.
- i** Requiere NetBox v2.8 o superior.

Ejemplo básico en Python

```
mi_plugin/  
├── __init__.py % contiene PluginConfig  
├── models.py  
├── views.py  
├── urls.py  
├── templates/  
│   └── mi_plugin/  
│       └── *.html  
├── api/  
│   └── serializers.py  
├── navigation.py  
├── pyproject.toml  
└── README.md
```

- `__init__.py` define la clase `PluginConfig` con metadatos y configuración.
- `pyproject.toml` o `setup.py` permite empaquetar e instalar el plugin.
- Se recomienda seguir las buenas prácticas de Django.

- **Ansible:**

- Colección oficial: `netbox.netbox`
- Inventario dinámico desde NetBox
- Uso como fuente de verdad para playbooks

- **Terraform:**

- Proveedor oficial: `e-breuninger/netbox`
- Gestión declarativa de objetos (sitios, dispositivos, IPs)
- Ideal para flujos de trabajo IaC

- **Otras integraciones:**

- Grafana (vía API)
- GitLab CI/CD (vía Webhooks)
- Prometheus (exportadores personalizados)

- **API RESTful:**

- Acceso completo a todos los modelos
- Ideal para scripts, pipelines y herramientas externas

- **Webhooks:**

- Disparadores automáticos ante cambios (crear, editar, borrar)
- Envío de eventos a sistemas externos (ej. GitLab, Slack)
- Soporte para plantillas Jinja2 y firmas HMAC

- **GitOps:**

- NetBox como fuente de verdad sincronizada con Git
- Integración con pipelines CI/CD para despliegues automáticos
- Ejemplo: añadir un dispositivo en NetBox → despliegue automático vía GitLab CI



Pasos y recomendaciones para su implantación

- **1. Audita tu documentación actual:** identifica carencias, duplicidades y estructura desorganizada.
- **2. Define una taxonomía clara:** roles, etiquetas, convenciones de nombres y flujos estándar.
- **3. Planifica tu estructura en NetBox:** sitios, racks, dispositivos, interfaces, etc.
- **4. Comienza pequeño:** inicia con un sitio o segmento antes de escalar al resto de la red.
- **5. Importa datos existentes:** desde Excel, scripts o herramientas externas; valida el proceso.
- **6. Activa seguridad y control de acceso:** SSO, LDAP o autenticación externa si es posible.
- **7. Explora automatización e integración:** Ansible, Terraform, CI/CD y API RESTful.
- **8. Capacita a tu equipo:** formación técnica y operativa según los perfiles.
- **9. Itera y mantén:** revisa datos, actualiza procesos y amplía funcionalidades según crezca la red.

🔗 **NetBox no es solo una herramienta, es una práctica continua de gestión de red.** 🔍 🔗 🔍 🔗

¿Qué es Nautobot?

- **Nautobot** es un fork de NetBox lanzado por la empresa *Network to Code* en 2021.
- Nació con el objetivo de explorar nuevas funcionalidades y enfoques que no encajaban en la visión original de NetBox, y buscar mayor extensibilidad.
- Incluye funcionalidades avanzadas como:
 - Validación personalizada de datos.
 - Integración nativa con Git para GitOps.
 - Base experimental con control de versiones (DoltDB).
 - Aplicaciones como "jobsz" data sources integradas.
- Al igual que NetBox, está desarrollado en **Python** con Django y PostgreSQL.
- Su comunidad es más pequeña pero con fuerte respaldo empresarial.

 Un enfoque alternativo centrado en la extensibilidad y GitOps.

NetBox vs Nautobot

Característica	NetBox	Nautobot
Origen	DigitalOcean (2016)	Network to Code (2021)
Licencia	Apache 2.0	Apache 2.0
Lenguaje / Framework	Python / Django	Python / Django
Automatización	API REST + Webhooks	API REST + GraphQL + GitOps
Extensibilidad	Plugins	Plugins + Apps internas
Validaciones	Básicas por modelo	Reglas definibles por el usuario
Base de datos	PostgreSQL	PostgreSQL / Dolt (experimental)
Comunidad	Muy activa, oficial y OSS	Más reducida, orientada por NTC
Soporte comercial	NetBox Labs	Network to Code

Ambos proyectos son software libre
y válidos según tus necesidades y filosofía DevOps.

? ¿Deberíamos migrar a Nautobot?

- ¿Necesitas validaciones de datos más flexibles y lógicas condicionales?
- ¿Tu organización ya trabaja con flujos GitOps y pipelines basados en Git?
- ¿Estás dispuesto a asumir una curva de aprendizaje mayor a cambio de más extensibilidad?

- **NetBox**: madurez, comunidad activa y soporte extendido.
- **Nautobot**: innovación orientada a automatización avanzada.

 No hay una única respuesta correcta. Depende de tu cultura DevOps y tus objetivos.

-  **NetBox Tutorial - Zero To Hero Training:**
<https://netboxlabs.com/netbox-zero-to-hero-training/>
-  **Migración de datos a NetBox:**
<https://netboxlabs.com/docs/netbox/getting-started/planning/>
-  **Scripts personalizados:**
<https://netboxlabs.com/docs/netbox/customization/custom-scripts/>
-  **Desarrollo de plugins:**
<https://netboxlabs.com/docs/netbox/plugins/development/>
-  **Plantilla Cookiecutter para plugins:**
<https://github.com/netbox-community/cookiecutter-netbox-plugin>
-  **Documentación general de NetBox:**
<https://netbox.readthedocs.io/en/stable/>
-  **Canal de Youtube sobre Netbox:**
<https://www.youtube.com/playlist?list=PL3axWAYLEDVQmaCIQw9okadaNtWzrUlq>

 Todos los enlaces son oficiales y mantenidos por la comunidad o NetBox Labs.

¿Preguntas, ideas o experiencias que quieras compartir?

“Documentar es el primer paso para automatizar.”