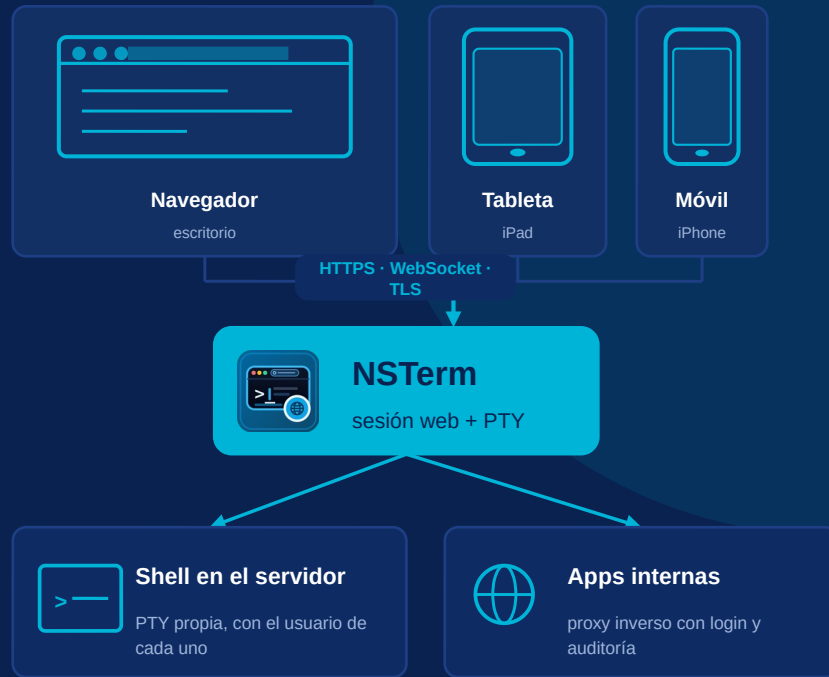


El terminal de su servidor, en el navegador. Sin SSH.

NSTerm publica el shell como una aplicación web: se abre en cualquier navegador, desde el portátil, la tableta o el móvil, por HTTPS. Y las consolas internas se publican con proxy inverso, con su login y su auditoría.

[VER FUNCIONALIDADES →](#)

Solo HTTPS · Nada de claves SSH · Un solo binario



conexiones por navegador, tableta y móvil

sessions · apps · groups · audit · proxy

Entrar al servidor por SSH no escala

Cada operador lleva su cliente SSH, sus claves y sus túneles: la sesión no se puede auditar y la consola interna queda fuera del control de acceso.

01 Claves por persona

Un keyring que nadie mantiene. La rotación es un proyecto de fin de quarter.

02 Sin identidad central

Cada uno entra como puede. No hay usuario, ni rol, ni caducidad.

03 Sin auditoría

El login SSH no deja rastro útil: no sabe qué se ejecutó ni desde dónde.

04 Consolas sueltas

La intranet se publica por su cuenta, saltándose la política de acceso.

• Portapapeles y una caja de texto

• Sin control de acceso

• Consolas sin traza

QUÉ ES

El shell del servidor, como una aplicación web

NSTerm es un terminal web multi-sesión en un solo binario: el administrador publica aplicaciones (comando, argumentos, directorio, entorno y tamaño), los usuarios las abren en terminales con pestañas sobre WebSocket y cada proceso queda registrado como sesión.

- ✓ Terminales xterm en el navegador, con pestañas y reconexión
- ✓ Sesiones con PID, usuario, IP, país, navegador y procesos hijos
- ✓ El comando corre en el servidor, con el usuario que el administrador elija
- ✓ Consolas internas publicadas por proxy inverso, con login y RBAC

0

claves SSH que
gestionar

1

binario, sin
dependencias

HTTPS

el único requisito de
red

3

tipos de dispositivo

Cómo viaja una pulsación, paso a paso



⚠ La sesión vive en el servidor: si el navegador se cierra, el proceso sigue y se puede recuperar al volver.

Solo HTTPS y WSS

Sin shell en el cliente

Sesión auditable

Sin SSH: solo HTTPS

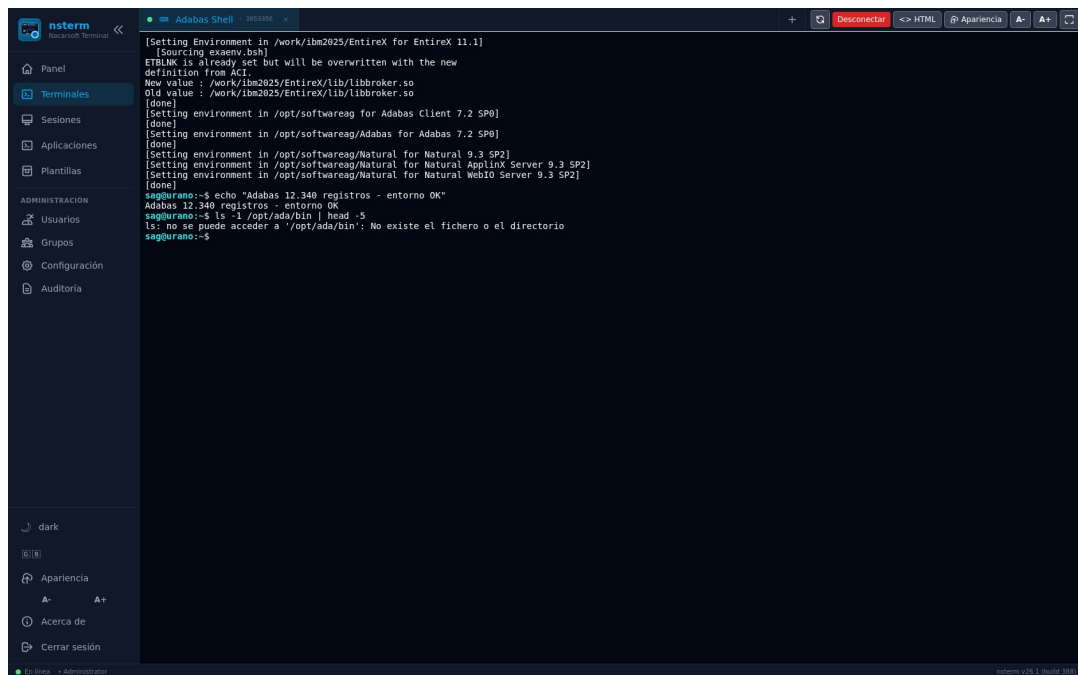
No hace falta sshd, ni claves, ni bastion, ni túneles. NTerm speaks WebSocket sobre TLS y el comando se ejecuta en el propio servidor.

Acceso	Cliente SSH + clave privada	Navegador con cookie de sesión
Identidad	Clave pública del usuario	Usuario, grupo y rol del panel
Auditoría	En el mejor de los casos, el auth.log	Registro por sesión, usuario e IP
Cupos	No existen	Máximo de sesiones por usuario y grupo
Consolas internas	Túneles y port forwarding	Proxy inverso con RBAC
Para el usuario	Instalar y configurar un cliente	Abrir una URL

El único requisito de red es un proxy inverso con HTTPS en el puerto 11030.

CONEXIONES

Desde el navegador de escritorio



- xterm.js con el juego de caracteres y los colores del servidor
- Pestañas: varias terminales a la vez, con nombre y PID
- Reconexión: al recargar la página, la sesión se recupera
- Pantalla completa, ajuste de fuente y espejo HTML
- Atajos y teclas de función de Natural (PF) cuando la app lo pide

Sin plugin

Sin Java de putty

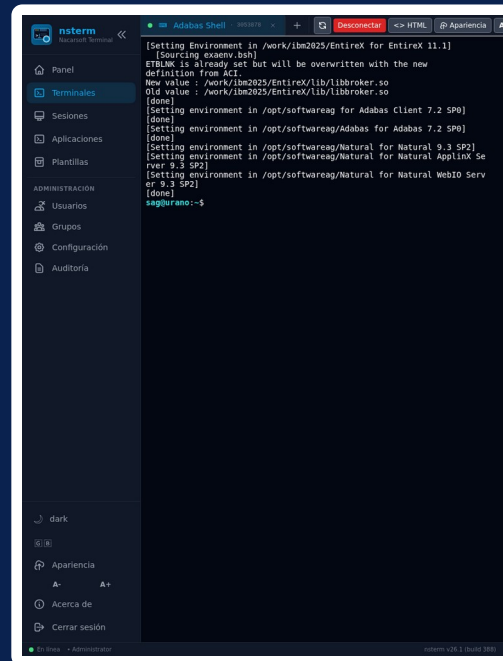
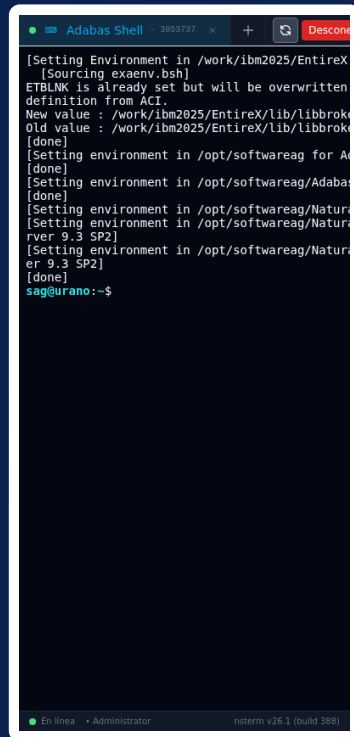
Bookmarks y favourites

Terminal real en el navegador, con el PID de la sesión en la pestaña.

También desde la tableta y el móvil

- La misma sesión, con el layout adaptado al ancho de la pantalla
- Teclado en pantalla con las teclas que la app necesita
- Pulsación larga y selección de texto sobre la salida del comando
- Ideal para consultas puntuales y guardia fuera del escritorio

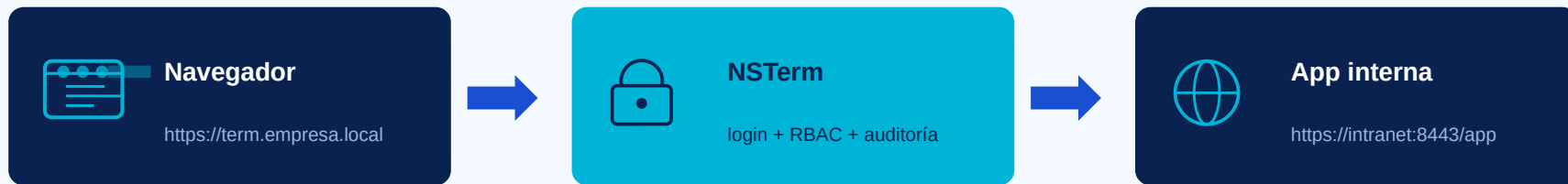
Un mismo enlace vale para el PC, la tableta y el teléfono.



Tableta: 834 px, ancho de trabajo cómodo.

Proxy inverso: publica tus consolas internas

Además de terminales, NTerm publica aplicaciones web internas: expone una URL http/https a través del propio NTerm, heredando el login, el RBAC por grupos y la auditoría.



- Solo URLs definidas por el administrador: el usuario nunca escribe una
- Allowlist de hosts, bloqueo de IP privadas y de metadatos cloud
- Reescritura de enlaces y soporte de WebSocket en el backend
- Límite de cuerpo y timeout de respuesta por aplicación

La consola interna deja de estar en la red y pasa a entrar por el mismo login que el terminal.

Aplicaciones: el administrador decide qué se publica

- Comando, argumentos, directorio de trabajo y entorno
- Tamaño de terminal (cols/rows) o ajuste al contenedor
- Esquema de color, icono, logo y color de la aplicación
- Usuario del sistema con el que corre, fijo o el del usuario conectado
- Timeout de sesión, diseño HTML y teclas de Natural

Un usuario solo ve las aplicaciones que su grupo tiene asignadas.

Editar Aplicación

General Ejecución Terminal y estilo

ID de App: adashell Nombre: Adabas Shell

Clave única. Renombrar reasigna las sesiones abiertas; el historial cerrado conserva el anterior.

Descripción: Shell del servidor Adabas

Tipo: Terminal (comando SSH)

Terminal lanza un comando del servidor. URL proxy incrusta una app web en una pestaña a través de nsterm.

☒ Activo

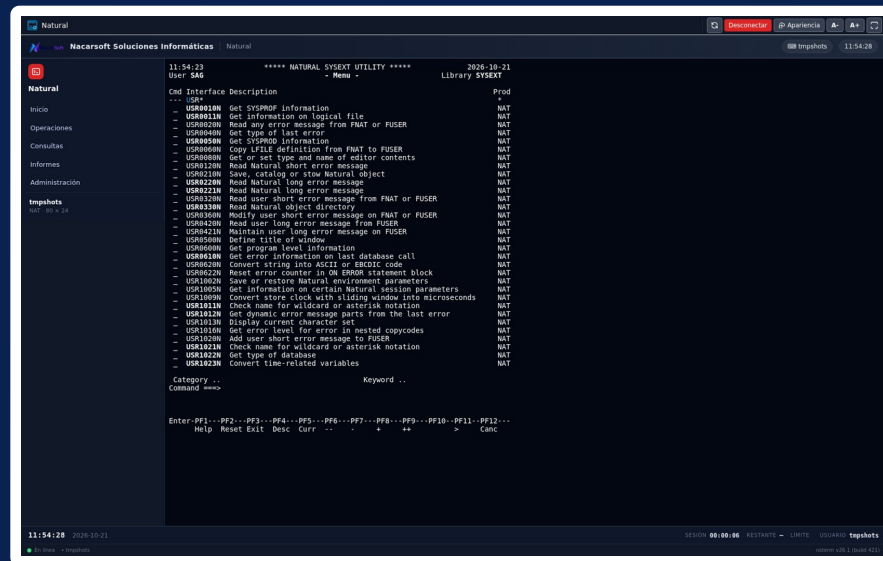
Cancelar Guardar

El editor de una aplicación: id, nombre, descripción, tipo y estado.

Plantillas: el terminal con aspecto de aplicación web

- Cabecera, contenedores laterales y barra de estado con HTML y CSS
- Logo y nombre corporativo desde la configuración del servidor
- Botones PF1..PF12 que envían la tecla a la sesión
- Marcadores vivos: reloj, fecha, tiempo de sesión y restante
- El CSS va aislado: no puede romper la interfaz de NSterm

Once plantillas de ejemplo se crean al instalar: asigna la que quieras a cada aplicación.



La misma aplicación, con cabecera corporativa, menú lateral y barra de estado con el reloj.

HORARIOS

Horarios: la aplicación solo está disponible en su franja

- Plantillas de horario: mañana, tarde, noche, fines de semana
- Fuera de la franja la app aparece como Cerrada en el panel
- El servidor la bloquea: no se puede abrir saltándose el panel
- Franja por día, varias ventanas y horario de noche
- Los administradores también quedan sujetos

Nombre	Días	Ventanas	Estado	Acciones
Fines de semana Todo el día, sábado y domingo	S-D	00:00-24:00	Cerrada	 
Horario laboral De 08:00 a 12:00 y de 16:00 a 18:00, de lunes a viernes	L-V	08:00-12:00, 16:00-18:00	Abierta	 
Mañana De 08:00 a 14:00, de lunes a viernes	L-V	08:00-14:00	Abierta	 
Noche De 22:00 a 06:00, de lunes a viernes (cruza medianoche)	L-V	22:00-06:00	Cerrada	 
Tarde De 16:00 a 20:00, de lunes a viernes	L-V	16:00-20:00	Cerrada	 

Las plantillas de horario disponibles y su estado actual.

**El acceso a la consola pasa a ser horario de oficina,
no siempre abierto.**

SESIONES

Cada sesión es un proceso, y queda registrada

- Aplicación, usuario, PID, inicio y duración de cada sesión
- IP, país y navegador de quien la abrió
- Procesos hijos: lo que ha lanzado el comando
- CPU, memoria y timeout de sesión configurado
- Terminar la sesión pidiendo un motivo, que va al registro

App	Usuario	PID	Iniciado ♀	Duración	Límite	CPU	MEM	IP	País	Navegador	Acciones
 adashell	admin	3954039	22:35:13	6s	⌚ 4h · 3h 59m 51s	–	6 M	127.0.0.1	Local	Chrome 144 · Linux	
 adashell	admin	3953878	22:35:08	11s	⌚ 4h · 3h 59m 46s	–	6 M	127.0.0.1	Local	Chrome 144 · Linux	
 adashell	admin	3953737	22:35:03	16s	⌚ 4h · 3h 59m 41s	–	6 M	127.0.0.1	Local	Chrome 144 · Linux	
 naturalsh	admin	3953545	22:34:59	20s	⌚ 2h · 1h 59m 37s	–	6 M	127.0.0.1	Local	Chrome 144 · Linux	
 adashell	admin	3953356	22:34:51	28s	⌚ 4h · 3h 59m 29s	–	6 M	127.0.0.1	Local	Chrome 144 · Linux	

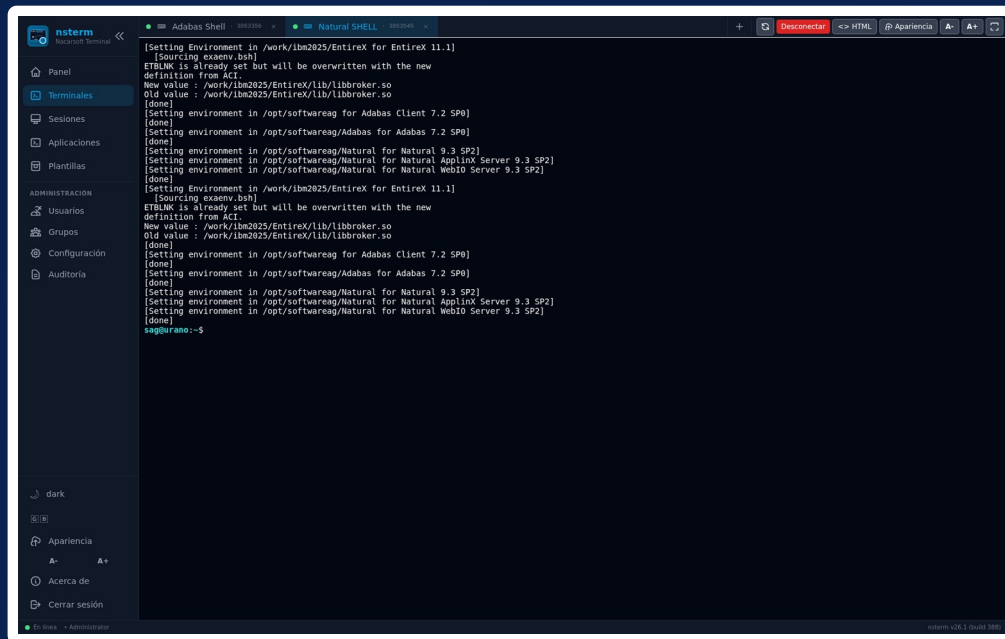
Sesiones: quién ejecuta qué, desde dónde y desde cuándo.

La tabla se actualiza en vivo por WebSocket: no hay que recargar.

MULTITERMINAL

Varias terminales, una ventana

- Nueva pestaña sin perder la sesión anterior
- Cada pestaña con su PID y su nombre de aplicación
- Reconexión automática tras un corte de red
- El estado de la sesión, visible en la barra de pestañas



Pestañas: una terminal por sesión, todas en la misma ventana.

Usuarios, grupos y cupos de sesiones

- Un usuario pertenece a grupos; el grupo define el rol
- Cada grupo tiene sus aplicaciones asignadas
- Cupo de sesiones simultáneas por usuario y aplicación
- Al superar el cupo, el terminal avisa y no abre la sesión
- Administración de usuarios, alias, email y estado

Nombre	Descripción	Rol	Aplicaciones	Creado	Acciones
admins	Administrators	Admin	—	2026-10-19	Apps Eliminar
guardia	Guardia de produccion	User	Natural Shell	2026-10-19	Apps Eliminar
operadores	Operadores de terminal	User	Adabas Shell Consola Interna	2026-10-19	Apps Eliminar
users	Default user group	User	—	2026-10-19	Apps Eliminar

Grupos: rol, aplicaciones asignadas y el máximo de sesiones por usuario.

La misma aplicación puede estar en varios grupos, con cupos distintos.

Quién abrió qué, y queda registrado

- Accesos, altas de aplicaciones, cambios de grupo y fin de sesión
- Motivo de terminación obligatorio al cerrar la sesión de otro
- LDAP o Active Directory opcional, junto a los usuarios locales
- Políticas de contraseña: caducidad, aviso previo e historial
- Token de sesión con caducidad configurable y cookie HttpOnly

Hora	Usuario	Acción	Detalle	IP	País
2026-10-19 22:35:13	admin	terminal_open	adashell	127.0.0.1	Local
2026-10-19 22:35:08	admin	terminal_open	adashell	127.0.0.1	Local
2026-10-19 22:35:03	admin	terminal_open	adashell	127.0.0.1	Local
2026-10-19 22:34:59	admin	terminal_open	naturalsh	127.0.0.1	Local
2026-10-19 22:34:51	admin	terminal_open	adashell	127.0.0.1	Local
2026-10-19 22:34:45	#0	login	admin	127.0.0.1	Local
2026-10-19 22:34:45	admin	theme_change	dark	127.0.0.1	Local
2026-10-19 18:46:39	admin	terminal_close	session=63 appnashell pid=2593451 reason=idle timeout (no clients)	127.0.0.1	Local
2026-10-19 18:46:34	admin	terminal_close	session=62 appnashell pid=2593296 reason=idle timeout (no clients)	127.0.0.1	Local
2026-10-19 18:46:29	admin	terminal_close	session=61 appnashell pid=2593159 reason=idle timeout (no clients)	127.0.0.1	Local
2026-10-19 18:46:24	admin	terminal_close	session=60 app=naturalsh pid=2592990 reason=idle timeout (no clients)	127.0.0.1	Local
2026-10-19 18:46:24	admin	terminal_close	session=59 appnashell pid=2592654 reason=idle timeout (no clients)	127.0.0.1	Local
2026-10-19 18:45:44	admin	terminal_close	session=58 appnashell pid=2591763 reason=idle timeout (no clients)	127.0.0.1	Local
2026-10-19 18:45:38	admin	terminal_close	session=57 appnashell pid=2591605 reason=idle timeout (no clients)	127.0.0.1	Local
2026-10-19 18:45:33	admin	terminal_close	session=56 appnashell pid=2591482 reason=idle timeout (no clients)	127.0.0.1	Local

Auditoría: accesos, aplicaciones, grupos y terminaciones de sesión.

Licencia firmada por máquina y trial de 30 días.

De cero a terminal web, en tres pasos

1

Crear el entorno

nsterm init /opt/nsterm: crea config.yaml, la base SQLite, los logs y el usuario admin.

2

Publicar el proxy inverso

Termina TLS en el proxy y reenvía a https://servidor:11030. No hace falta tocar el servidor.

3

Publicar la shell

Crea la aplicación (comando, args, directorio y entorno) y asígnala a los grupos que deban usarla.

```
nsterm init /opt/nsterm → nsterm start --config /opt/nsterm →  
https://term.empresa.local
```

REQUISITOS

- Linux (RHEL 8 / 9)
- Un solo binario, sin dependencias
- SQLite para el estado y la auditoría
- Proxy inverso con HTTPS en producción
- Puerto 11030 (configurable)

Sin SSH: no hay que abrir el 22, ni desplegar claves, ni tocar la configuración de sshd.

Lo que más se pregunta

¿De verdad no hace falta SSH?

No. El navegador habla HTTPS y WebSocket con NTerm, y el comando se ejecuta en el servidor con un PTY propia. No interviene sshd.

¿Quién ejecuta el comando?

El administrador lo decide: un usuario del sistema fijo o el usuario Linux mapeado de quien se conecta. NTerm verifica los permisos sudoers.

¿Puedo publicar una web interna?

Sí: crea una aplicación de tipo Proxy URL con su target y asígnala a grupos. Los usuarios la abren con su sesión y sin acceso directo a la URL.

¿Y si el navegador se cierra?

La sesión sigue viva en el servidor. Al volver, la pestaña se recupera y se reengancha al mismo proceso, con su PID.

¿Cómo se limita el uso?

Cada grupo tiene un máximo de sesiones simultáneas por usuario y aplicación. Al superarlo, el terminal avisa y no abre una sesión nueva.

¿Se puede integrar con el Active Directory?

Sí: LDAP/AD es opcional y convive con los usuarios locales. El usuario se mapea a su cuenta Linux para el PTY.

Su servidor, en el navegador y con su propia auditoría

Pruebe NTerm con su entorno: lo montamos con sus aplicaciones y lo vemos funcionando.

Demo guiada

Pruébalo en su entorno

Hablemos de su caso

nacarsoft.com · soporte@nacarsoft.com

