SERVIDOR PROPIO · PARA TI Y SEGURIDAD

Nota de seguridad

Cada afirmación se comprobó el 15 de noviembre de 2026 en una máquina limpia con Ubuntu 22.04 (2 núcleos, 4 GB) y Layers 3.0.1 instalado con el comando del correo de la licencia. Cada una lleva un comando para que su equipo la compruebe en su servidor. Lo que no comprobamos, lo decimos.

La contraseña del panel de administración

Desde el 15.11.2026 el instalador pone una contraseña aleatoria al panel de administración (God Mode, /godmode) en /opt/layers/.env (ADMIN_PASSWORD, permisos 600). Los servidores instalados antes aceptan admin / admin: ejecute otra vez el comando de instalación del correo o el comando de abajo. Comprobado en una máquina limpia: después, admin / admin recibe 401, también tras reiniciar; repetirlo no modifica nada.

```
cd /opt/layers && P=$(openssl rand -hex 16) && T=$(sudo docker compose exec -T layers wget -qO- --post-data "username=admin&password=admin&bearer=true" http://localhost:8080/data/auth/admin/login | sed -n 's/.*"token":"\([^"]*\)".*/\1/p') && [ -n "$T" ] && sudo docker compose exec -T layers wget -qO- --header "Authorization: Bearer $T" --header "Content-Type: application/json" --post-data "{\"current\":\"admin\",\"next\":\"$P\"}" http://localhost:8080/data/auth/admin/password >/dev/null && echo "ADMIN_PASSWORD=$P" | sudo tee -a .env >/dev/null && echo "Done: God Mode password is in /opt/layers/.env (ADMIN_PASSWORD)"
```

Comprobación

```
cd /opt/layers && sudo docker compose exec -T layers wget -qO- --post-data "username=admin&password=admin" http://localhost:8080/data/auth/admin/login >/dev/null 2>&1 && echo "NOT CLOSED: admin/admin still works" || echo "Closed: admin/admin is refused"
```

Dónde están los datos

- Todo está en /opt/layers, en su máquina: la base de datos PostgreSQL en data/db, los archivos de las personas en data/layers/files, los certificados HTTPS en data/caddy, la clave de licencia en license/, los secretos del servidor en .env.
- .env se crea con permisos 600 (solo root); la carpeta de la base de datos, 700.
- Los archivos de las personas se escriben con permisos 644. Desde el 15.11.2026 el instalador cierra su carpeta (750) a otras cuentas del servidor; en servidores instalados antes ejecute el comando de abajo: el servidor sigue funcionando (comprobado, sano tras reiniciar).

Comprobación

```
sudo ls -la /opt/layers /opt/layers/data
sudo docker inspect layers-layers-1 -f '{{range .Mounts}}{{.Source}} → {{.Destination}}{{println}}{{end}}'
```

Cerrar los archivos a otras cuentas del servidor

```
sudo chmod 750 /opt/layers/data/layers && sudo docker compose -f /opt/layers/docker-compose.yml restart layer
s
```

Puertos y a dónde se conecta el servidor

- Escucha: TCP 80 (redirección a HTTPS y emisión del certificado), TCP 443 (sitio y TURN sobre TLS para llamadas), UDP 443, TCP 7881, UDP 7882, UDP 3478 (llamadas). Nada más.
- Los editores de documentos, a4 (.docx) y f1 (.xlsx), son contenedores aparte sin puertos propios: con ellos la lista de arriba es la misma (comprobado el 15.11.2026), y el navegador llega a ellos solo a través de Caddy en /a4/ y /f1/. Según su configuración en docker-compose.yml, preguntan a Layers quién es la persona y toman y guardan el archivo a través de Layers, así que el Disco decide quién lo abre; las copias de trabajo de los documentos abiertos están en data/editors.
- Desde el 15.11.2026 el panel de administración (/godmode y su API) solo responde a redes locales y a la propia máquina; el resto recibe 403 (comprobado en una instalación limpia). Otras direcciones se añaden con la línea GODMODE_ALLOW en .env o con --godmode-allow del instalador.
- Al instalar, la máquina descarga de layers.md (install.sh, docker-compose.yml), de Docker Hub (imágenes), del archivo de Ubuntu (Docker, si falta) y pregunta a api.ipify.org su dirección externa para avisar sobre el DNS.
- En funcionamiento (10 minutos observados con la salida a internet cortada, tras un reinicio completo): el servicio de llamadas preguntó a servidores STUN públicos (stun.l.google.com, global.stun.twilio.com) su dirección externa, y Layers intentó llegar a api.layers.md: la marca horaria de la licencia (sección siguiente). Nada más; el navegador no hace peticiones a direcciones externas.
- Red cerrada: defina NODE_IP (desde el 15.11.2026 el instalador lo escribe solo) y desactive la marca de la licencia. Comprobado: tras un reinicio completo sin internet, 6 minutos, cero intentos de salida, servicio de llamadas activo, licencia activa.
- Con los editores de documentos (comprobado el 15.11.2026: salida a internet cortada, reinicio completo, 7 minutos de observación): el servidor arrancó en 104 s, licencia activa, servicio de llamadas en marcha con NODE_IP, ambos editores healthy. El editor de documentos (a4) intentó 7 veces llegar a support.sumoffice.com, el servicio del fabricante del editor, y siguió funcionando sin respuesta; el editor de hojas de cálculo (f1) no intentó salir.

- Sin NODE_IP y sin internet el servicio de llamadas no arranca (se reinicia con «could not resolve external IP»); el resto de Layers funciona.
- Con un dominio real, Caddy obtiene un certificado de Let’s Encrypt por internet; no lo comprobamos en la máquina de prueba (usaba layers.localhost y su propio certificado).

Puertos abiertos

```
sudo ss -ltnup | grep docker-proxy
```

Cómo contamos los intentos de salida

```
SUB=$(sudo docker network inspect layers_default -f '{{(index .IPAM.Config 0).Subnet}}')
sudo iptables -I DOCKER-USER 1 -s $SUB -j DROP
sudo iptables -I DOCKER-USER 1 -s $SUB -j LOG --log-prefix "LAYERS-OUT "
for n in 10.0.0.0/8 172.16.0.0/12 192.168.0.0/16; do sudo iptables -I DOCKER-USER 1 -s $SUB -d $n -j RETURN;
done
sudo dmesg | grep LAYERS-OUT          # what the server tried to send out
```

Red cerrada

```
cd /opt/layers
echo "NODE_IP=10.0.0.10" | sudo tee -a .env >/dev/null       # the server's address in your network
sudo tee docker-compose.override.yml >/dev/null <<'EOF'
services:
  layers:
    environment:
      LAYERS_LICENSE_SEEN_ENABLED: "false"
EOF
sudo docker compose up -d
```

Qué sale del servidor; ¿la licencia se comprueba sin conexión?

- La clave se comprueba en el propio servidor. Comprobado: salida a internet cortada, reinicio completo de todos los servicios — licencia LICENSED, modo full, nadie bloqueado.
- Si el servidor tiene internet, una vez por hora envía a api.layers.md el id de la clave, el id de la instalación, el id del nodo y la versión — sin personas, archivos ni contenido. La respuesta puede traer una clave renovada para esta instalación, que el servidor comprueba (firma, instalación) e instala solo. Sin respuesta, nada cambia.
- La marca se desactiva con LAYERS_LICENSE_SEEN_ENABLED=false (vea «Red cerrada» arriba); entonces la clave renovada se copia como archivo en /opt/layers/license.

Comprobación

```
curl -s https://layers.example.com/layers/license/status | grep -o '"mode":"[A-Z]*"\|"regime":"[a-z-]*"'
# "mode":"LICENSED"
# "regime":"full"
```

Copias de seguridad y restauración

- Todo el estado es la base de datos, los archivos de las personas, .env y la carpeta de la licencia. La copia son dos comandos, sin detener el servidor.
- Restauración comprobada: base de datos borrada y archivos eliminados, luego restaurados desde la copia — las mismas 5 personas, 8 páginas, el mismo archivo con la misma suma de control, licencia activa, el propietario ve sus archivos. El servidor volvió en 1 min 40 s aproximadamente.

Copia

```
cd /opt/layers
sudo docker compose exec -T db pg_dump -U layers -d layersdb -Fc -Z6 > ~/layersdb.dump
sudo tar czf ~/layers-files.tar.gz -C /opt/layers data/layers .env license
```

Restauración

```
cd /opt/layers
sudo docker compose stop layers
sudo docker compose exec -T db psql -U layers -d postgres -c "drop database layersdb with (force)"
sudo docker compose exec -T db psql -U layers -d postgres -c "create database layersdb owner layers"
sudo docker compose exec -T db pg_restore -U layers -d layersdb --no-owner < ~/layersdb.dump
sudo tar xzf ~/layers-files.tar.gz -C /opt/layers data/layers
sudo docker compose up -d
```

Actualización y vuelta atrás

- La versión es una línea en .env (LAYERS_IMAGE). Actualizar: copia de la base, nueva línea, docker compose up -d; los cambios de la base se aplican solos. La interrupción fue de unos 80 segundos.
- Vuelta atrás comprobada: 3.0.1 → 3.0.0 sobre la misma base arrancó y conservó la licencia (la herramienta de la base avisa de que el esquema es más nuevo), y después otra vez a 3.0.1. La vuelta atrás segura es la versión anterior más la copia hecha antes de actualizar (restauración arriba).

Actualizar

```
cd /opt/layers
sudo docker compose exec -T db pg_dump -U layers -d layersdb -Fc -Z6 > ~/before-update.dump
sudo sed -i '/^LAYERS_IMAGE=/d' .env && echo "LAYERS_IMAGE=hissih/layers:3.0.1" | sudo tee -a .env >/dev/null
sudo docker compose up -d
sudo docker inspect layers-layers-1 -f '{{.Config.Image}} {{.State.Health.Status}}'
```

Volver atrás

```
cd /opt/layers
sudo sed -i '/^LAYERS_IMAGE=/d' .env && echo "LAYERS_IMAGE=hissih/layers:3.0.0" | sudo tee -a .env >/dev/null
sudo docker compose up -d
```

Registros y auditoría

- Cada petición es una línea en el registro del servidor: método, ruta, respuesta, tiempo, id de usuario y navegador.
- El panel de administración guarda los intentos de acceso (hora, dirección, éxito); conserva los recientes desde el último reinicio.
- Los accesos de las personas se guardan en la base de datos (auth_journal: hora, método, resultado, dirección, navegador).
- Los registros de los contenedores rotan: docker-compose.yml conserva como máximo 5 archivos de 20 MB por contenedor (controlador json-file, max-size y max-file); cambie los límites allí o envíe los registros a su sistema. Un nodo instalado antes obtiene la rotación con el nuevo docker-compose.yml y docker compose up -d.

Registro del servidor y controlador

```
cd /opt/layers
sudo docker compose logs --since 1h layers | grep "\[access\]"
sudo docker inspect layers-layers-1 -f '{{.HostConfig.LogConfig.Type}} {{json .HostConfig.LogConfig.Config}}'
# json-file {"max-file":"5","max-size":"20m"}
```

Accesos al panel de administración

```
J=$(mktemp)
curl -s -c $J -o /dev/null https://layers.example.com/data/auth/admin/login --data "username=admin&password=
$(sudo sed -n 's/^ADMIN_PASSWORD=//p' /opt/layers/.env)"
curl -s -b $J https://layers.example.com/data/auth/admin/logins; rm -f $J
# [{"at":"2026-11-15T00:31:13Z","ip":"…","ok":true}, …]
```

Quién tiene acceso

- root en el servidor tiene todo: .env, base de datos, archivos. Mantenga la máquina solo en manos de sus administradores.
- El panel de administración (God Mode): una cuenta de servicio, admin, con una contraseña aleatoria en /opt/layers/.env (ADMIN_PASSWORD); solo se abre desde redes locales, salvo que indique otras direcciones en GODMODE_ALLOW. En servidores instalados antes del 15.11.2026 la contraseña es admin y el panel se abre desde cualquier sitio: cambie la contraseña (comando arriba) y use el archivo de instalación del 15.11.
- Dentro de Layers, el propietario del espacio y los administradores que designe gestionan personas y accesos.
- El equipo de Layers no tiene acceso a su servidor: no se instala nada para acceso remoto, y la única llamada de Layers hacia fuera es la marca de la licencia, que puede desactivar.

Cambiar la contraseña de God Mode

```
cd /opt/layers && P=$(openssl rand -hex 16) && T=$(sudo docker compose exec -T layers wget -qO- --post-data
"username=admin&password=admin&bearer=true" http://localhost:8080/data/auth/admin/login | sed -n 's/.*"toke
n":"\([^"]*\)".*/\1/p') && [ -n "$T" ] && sudo docker compose exec -T layers wget -qO- --header "Authorizatio
n: Bearer $T" --header "Content-Type: application/json" --post-data "{\"current\":\"admin\",\"next\":\"$P\"}"
http://localhost:8080/data/auth/admin/password >/dev/null && echo "ADMIN_PASSWORD=$P" | sudo tee -a .env >/de
v/null && echo "Done: God Mode password is in /opt/layers/.env (ADMIN_PASSWORD)"
```

Comprobación

```
cd /opt/layers && sudo docker compose exec -T layers wget -qO- --post-data "username=admin&password=admin" ht
tp://localhost:8080/data/auth/admin/login >/dev/null 2>&1 && echo "NOT CLOSED: admin/admin still works" || ec
ho "Closed: admin/admin is refused"
```

Desde dónde se abre God Mode

```
curl -s -o /dev/null -w "%{http_code}\n" https://layers.example.com/godmode      # from a machine not in GODM
ODE_ALLOW
# 403
cd /opt/layers && sudo sed -i '/^GODMODE_ALLOW=/d' .env && echo "GODMODE_ALLOW=private_ranges 203.0.113.7" |
sudo tee -a .env >/dev/null && sudo docker compose up -d
```

Cifrado

- En tránsito: HTTPS con TLS 1.3 (Caddy); el puerto 80 responde 308 a HTTPS. Las llamadas desde redes con solo el 443 abierto van por TURN sobre TLS en el 443.
- En disco: la base de datos y los archivos de las personas se guardan tal cual: un .docx en disco es el mismo archivo. Si su política exige cifrado en reposo, ponga /opt/layers en un disco cifrado (por ejemplo, LUKS). No comprobamos ese montaje.
- Las contraseñas de las personas se guardan como hashes bcrypt; la de God Mode, tras el cambio, también; las contraseñas de la base y de Redis y los secretos del servidor son aleatorios y están en .env (600).

En tránsito

```
echo | openssl s_client -connect layers.example.com:443 -servername layers.example.com 2>/dev/null | grep -E
"Protocol|Cipher"
curl -s -o /dev/null -w "%{http_code} %{redirect_url}\n" http://layers.example.com/
# 308 https://layers.example.com/
```

En disco

```
sudo file /opt/layers/data/layers/files/* | head
sudo docker compose -f /opt/layers/docker-compose.yml exec -T db psql -U layers -d layersdb -Atc "select left
(password,4) from users limit 3"
# $2a$  (bcrypt)
```

Comprobado el 15.11.2026, Layers 3.0.1

Esta página se preparó con ayuda de IA a partir de los registros de nuestras pruebas.