

Documentación

Invoice Ninja

Invoice Ninja

Anonymous · 25/09/2026

Tabla de contenidos

Invoice Ninja 3

Configuración inicial 4

Instalación con Docker 5

Documentación » Orígenes de datos » Invoice Ninja

Invoice Ninja

Invoice Ninja

Anonymous · 25/09/2026

Invoice Ninja

Última actualización el 25/09/2026

Documentación » Orígenes de datos » Invoice Ninja » Configuración inicial

Configuración inicial

Configuración de Invoice Ninja

Anonymous · 25/09/2026

Configuración de Invoice Ninja

Documentación en progreso.

Si tienes alguna sugerencia o necesitas ayuda, por favor contáctanos a través de nuestro [soporte](#).

Última actualización el 25/09/2026

Documentación » Orígenes de datos » Invoice Ninja » Instalación con Docker

Instalación con Docker

Instalar Invoice Ninja con Docker

Anonymous · 25/09/2026

Instalar Invoice Ninja con Docker

Sigue la [guía de los stacks de Docker](#) para instalar Invoice Ninja con nuestro [stack](#) y ten en cuenta lo propio de esta plataforma. El detalle completo está en el README del repositorio.

Recursos

Este stack necesita unos 2 GB de disco para las imágenes y 1 GB de RAM. En reposo usa alrededor de 910 MiB en total, la mayor parte en MySQL (unos 495 MiB). El primer arranque tarda menos de un minuto.

Variables obligatorias

Variable	Qué debe contener
NINJA_URL	URL pública de la plataforma, por ejemplo <code>https://billing.example.com</code> . En ella quedan la aplicación, el portal de clientes (<code>/client</code>) y la API.
SITE_ADDRESS	El dominio de la plataforma, por ejemplo <code>billing.example.com</code> .
APP_KEY	Clave de cifrado de Laravel: <code>base64:</code> seguido del resultado de <code>openssl rand -base64 32</code> .
DB_PASSWORD y DB_ROOT_PASSWORD	Contraseñas de la base de datos.
NINJA_ADMIN_EMAIL y NINJA_ADMIN_PASSWORD	Correo y contraseña del administrador. Solo se usan al crearlo en el primer arranque: cambiarlos después en el <code>.env</code> no modifica la cuenta.
SMTP_HOST	Recomendada: el stack arranca sin ella, pero la plataforma no podrá enviar correos. Servidor SMTP. Completa también <code>SMTP_PORT</code> , <code>SMTP_SECURE</code> , <code>SMTP_USER</code> , <code>SMTP_PASSWORD</code> , <code>SMTP_FROM</code> y <code>SMTP_FROM_NAME</code> según tu proveedor (por defecto, puerto <code>587</code> con <code>SMTP_SECURE=tls</code>).

Guarda `APP\_KEY` fuera del servidor

`APP_KEY` cifra datos como las credenciales de las pasarelas de pago. Está en el `.env`, no en los respaldos, y una base de datos restaurada la necesita.

Configuración inicial de la empresa

La empresa se crea configurada para Chile (moneda CLP, país CL, idioma es_ES, zona horaria America/Santiago, fechas d/m/Y y un impuesto IVA de 19% por defecto en las facturas nuevas, sumado al precio de cada línea). Si operas en otro país, cambia NINJA_COMPANY_NAME, NINJA_CURRENCY, NINJA_COUNTRY, NINJA_LANGUAGE, TIMEZONE, NINJA_DATE_FORMAT, NINJA_TAX_RATE, NINJA_TAX_NAME y NINJA_PRICES_INCLUDE_TAX **antes** del primer arranque: estos valores solo se aplican en la instalación inicial y después se modifican desde la aplicación. TIMEZONE también se aplica en cada arranque como zona horaria de los contenedores, pero con poco efecto: Laravel trabaja por defecto en UTC.

Se usa es_ES porque la traducción es de Invoice Ninja tiene textos con errores, entre ellos los asuntos de los correos de facturas y cotizaciones.

Acceso

Cuando setup termina, la aplicación queda disponible en tu dominio con el correo de NINJA_ADMIN_EMAIL. El portal de clientes está en /client y la API REST en /api/v1, con un token creado en la configuración de la aplicación (cabecera X-API-TOKEN).

En cada arranque, setup guarda en la empresa la dirección del portal de clientes que usan los enlaces de los correos: NINJA_PORTAL_URL si la defines, o NINJA_URL.

Actualizaciones

Invoice Ninja se actualiza cambiando NINJA_VERSION en el .env y ejecutando docker compose up -d. Respalda antes. Se descarga la nueva imagen y setup copia sus archivos públicos y ejecuta las migraciones antes de que arranque la aplicación. La actualización automática desde la aplicación no aplica, porque el código está en la imagen.

Restaurar un respaldo

Los respaldos incluyen la base de datos y la carpeta storage (logos y documentos). Para restaurar uno:

```
docker compose stop app worker scheduler
docker compose run --rm --no-deps backup restore <fecha>
docker compose exec redis redis-cli FLUSHALL
docker compose up -d
```

Notas

- Las facturas de Invoice Ninja no son documentos tributarios electrónicos válidos ante el SII: para emitirlos se conecta con BillMySales.
- La aplicación solo responde a los dominios de `NINJA_URL` y `NINJA_PORTAL_URL`. Si usas otro dominio más (por ejemplo `www`), agrégalo en `NINJA_EXTRA_HOSTS`.
- Invoice Ninja rechaza webhooks hacia direcciones privadas o locales: la URL que registres debe ser pública.
- Invoice Ninja valida los correos por DNS: una dirección en un dominio sin registros de correo no puede recuperar su contraseña.
- Invoice Ninja se distribuye bajo la Elastic License 2.0: puedes instalarlo en tu servidor, pero no ofrecerlo como servicio alojado. Quitar el texto “Created by Invoice Ninja” de correos y PDF requiere una licencia white label de pago.

Siguiente paso

Con Invoice Ninja funcionando, configúralo como origen de datos en BillMySales siguiendo la [configuración inicial](#).

Última actualización el 25/09/2026