

Documentación » Actualización

Actualización

Actualización de LibreDTE Slim

Anonymous · 02/10/2026

Actualización de LibreDTE Slim

Antes de actualizar, haz un [respaldo](#):

```
docker compose run --rm --no-deps backup now
```

Esto requiere el perfil `backup` activo y la base de datos corriendo.

Actualizar LibreDTE Slim

La versión se elige con `SLIM_VERSION` en `.env`, con el nombre del tag de la versión. El pie de la aplicación muestra la que está corriendo.

1. Cambia `SLIM_VERSION` en `.env` (por ejemplo, a `v0.2.0`).
2. Descarga la imagen y recrea los servicios:

```
docker compose pull
docker compose up -d
```

Al arrancar, el servicio `setup` aplica las migraciones de la base de datos y actualiza los archivos estáticos. Sigue su avance con `docker compose logs -f setup` hasta ver `==> Listo`; mientras corre, la aplicación no está disponible.

Si la imagen de ese tag no se puede descargar, Docker Compose lo avisa y la construye desde el código del repositorio (tarda varios minutos). Un tag que no existe termina en error.

Volver a una versión anterior

Cambiar `SLIM_VERSION` a la versión anterior no deshace las migraciones: una versión nueva puede haber modificado la base de datos. Para volver atrás, restaura el respaldo hecho antes de actualizar (ver [Respaldos](#)) y fija `SLIM_VERSION` a la versión anterior.

Construir desde el último código

Con `SLIM_VERSION` vacío, la imagen se construye en tu servidor con el último commit del repositorio.

Sirve para probar cambios, no para producción (no es una versión fija). Para traer lo último:

```
docker compose build --pull --no-cache
docker compose up -d
```

Actualizar LibreDTE Lib Core API

CORE_API_VERSION es `latest` por defecto, que es siempre la última versión publicada:

```
docker compose pull libredte-lib-core-api
docker compose up -d
```

Para fijar una versión, usa como valor el hash completo del commit (40 caracteres) de la versión deseada, que es el tag de sus imágenes. En ese caso, el pie de Slim muestra además la versión exacta de la Core API.

Actualizar PostgreSQL, Valkey y Caddy

Sus versiones se fijan con `POSTGRES_VERSION`, `VALKEY_VERSION` y `CADDY_VERSION`; los valores por defecto son los probados con cada versión de Slim. Los cambios dentro de la misma versión mayor (por ejemplo 18.1 a 18.2) se aplican con `docker compose pull` y `docker compose up -d`.

No cambies `POSTGRES_VERSION` a otra versión mayor (de 18 a 19, por ejemplo) sobre una base existente: PostgreSQL no puede abrir los datos de otra versión mayor. Para migrar, haz un respaldo, parte con volúmenes nuevos en la versión nueva y restaura el respaldo.

Parches de seguridad

Las imágenes de Slim y de la Core API se publican con las versiones de Python y PHP de ese momento. Para recibir parches de los componentes base, actualiza a la última versión publicada con los pasos anteriores.

Última actualización el 02/10/2026