railDB

Model railway collection database
Installation and User Manual

railDB v1.01 · Document revision 1.0 · 2026-11-24 · English
public edition

> **Before you start:** railDB holds personal data about your collection (prices, locations, photos). Use a long password, make backups and never expose it to the Internet without HTTPS.

Project website: tren.enmicasa.net — RailPIC projects and model railway electronics.

Contents

1. What railDB is
2. Installation
3. Getting started
4. Collection records
5. Decoders, functions and CVs
6. Workshop and spare parts
7. Photos, videos and documents
8. RailESP synchronization
9. Backups and updates
10. Troubleshooting
11. Tests, licence and references

1. What railDB is

railDB is a web application to manage a model railway collection: locomotives, wagons and scenery items, each with its own record. For every piece it keeps purchase and model data, its decoders and digital addresses, its functions with name and icon, CV lists, workshop jobs, the spare parts it uses, photos, videos and documents.

It is part of RailPIC and synchronizes with the **RailESP** command station: locomotives marked for the command station travel to RailESP with their address, functions, icons and avatar, and RailESP sends back running time and CV reads and writes.

What it is

A PHP web application with a MySQL database, used from the browser of a computer, tablet or phone.

What it is not

It is not train control software and does not replace the command station. railDB never writes to RailESP on its own: RailESP synchronizes when it starts, when asked to, or when a locomotive is saved.

Version

This manual describes railDB v1.01. The installed version is shown at the foot of every page once signed in.

> **Language:** the railDB interface is in Spanish. On-screen labels are quoted here exactly as they appear, with an English translation the first time, for example «Marcar como revisada» (mark as reviewed).

2. Installation

There are two ways to install railDB. To get started, or to run it at home, Docker is the simplest. To have it on the Internet, use a web hosting account with PHP and MySQL.

	Docker	**Web hosting**
Where	A Windows, macOS or Linux computer, a NAS or a Raspberry Pi with Docker.	A hosting account with PHP 8.5, MySQL 8 and console access (SSH with PHP).
What to prepare	Only Docker with Compose. PHP, Apache and MySQL come in the containers.	Database, folders, `config.php` and an HTTPS certificate.
Access	From the same computer, or from the home network if the port is opened.	From anywhere, always over HTTPS.

With Docker

1. Install Docker (Docker Desktop on Windows and macOS) and download the public railDB repository.
2. Optional: next to `compose.yaml`, create a `.env` file with your own database password: `RAILDB_DB_PASSWORD=...`. Do it before the first start: the database is created with it.
3. Start railDB, create the tables and create your user. The password (at least 10 characters) goes in the `RAILDB_NEW_PASSWORD` variable of your terminal (PowerShell: `$env:RAILDB_NEW_PASSWORD = Read-Host`; Linux or macOS: `read -s RAILDB_NEW_PASSWORD; export RAILDB_NEW_PASSWORD`), so it is never written in the command:

   ```
   docker compose up -d
   docker compose exec web php /var/www/html/src/bin/migrate.php
   docker compose exec -e RAILDB_NEW_PASSWORD web php /var/www/html/src/bin/create-user.php admin
   ```

4. Open http://127.0.0.1:8088 and sign in with that user.

To stop railDB: `docker compose stop`; to start it again: `docker compose up -d`. Data lives in two Docker volumes, `raildb-mysql` (the database) and `raildb-storage` (files, thumbnails, inbox and backups), and survives stopping and updating. `docker compose down -v` **deletes** them.

> **Access from other devices:** `compose.yaml` publishes railDB on `127.0.0.1` only, the computer itself. To use it from your home network, change `127.0.0.1:8088:80` to `8088:80`. Do not expose it to the Internet that way: without HTTPS the password would travel unencrypted.

On a web hosting account

Requirements: PHP 8.5 with `pdo_mysql`, `intl`, `zip`, `gd` and `exif`; MySQL 8; HTTPS; an SSH console with PHP for migrations. Recommended PHP limits: 512 MB memory and 128 MB uploads (as in `docker/php/php.ini`).

```
raildb/                outside the public web folder
  app/ bin/ migrations/ contents of src/app, src/bin and src/migrations
  config.php           from deploy/config.example.php (permissions 600)
  datos/               RAILDB_STORAGE_DIR: files/, thumbs/, entrada/, copias/
  logs/                PHP errors (created automatically)
www/raildb/            contents of src/public: the only part reachable from the web
```

1. Create the MySQL database in the hosting panel.
2. Upload the folders as shown. The public folder of the domain or subdomain must point to `www/raildb`; `index.php` finds the code in `../../raildb/app`.

3. Copy `deploy/config.example.php` to `raildb/config.php` and fill in the database and data folder. With this file railDB runs in production mode: it only answers over HTTPS and errors go to `raildb/logs/`, never to the screen.
4. Over SSH: `php raildb/bin/migrate.php` and `RAILDB_NEW_PASSWORD=... php raildb/bin/create-user.php user`.
5. Create a daily scheduled task running `raildb/bin/backup-cron.php` for backups (chapter “Backups and updates”).

3. Getting started

After signing in you land on «Inicio» (home): a summary of the collection (locomotives by traction, decoders, spare parts to restock, manuals) and the latest workshop job. At the top is the search over the whole collection and, on the right, «Contraseña» (password) and «Cerrar sesión» (sign out).

Menu section	Purpose
«Locomotoras», «Vagones», «Decoración» (locomotives, wagons, scenery)	The records of the collection, with filters and sorting.
«Repuestos» (spare parts)	Parts with stock levels and the pieces that use them.
«Taller» (workshop)	Every job in the collection, pending and done.
«Decoders»	Catalogue of decoder types with photo and manuals.
«Entrada» (inbox)	Drop folder to upload files in bulk and assign them later.
«Documentación» (documentation)	General documents that do not belong to one piece.
«RailESP»	Command stations, tokens, locomotives in the command station, open questions, icons and log.

- **Search:** finds pieces by name, article number, brand, series, running number, railway, epoch, address, protocol, decoder or motor, and also spare parts and decoders.
- **Password:** change it from «Contraseña» by typing the current one and the new one twice. After eight failed sign-ins from the same address, railDB waits 15 minutes.
- **Users:** all users see and edit the same data. They are created with `create-user.php`; the same command with an existing user sets a new password.

4. Collection records

Adding a piece

In locomotives, wagons or scenery, «+ Añadir» (add). Only the name is required. A record holds brand, article number, traction (locomotives), railway, series and running number, epoch, purchase year and price, current value, box, location, origin, category, production years, length, description, motor, real-prototype text, pending items and notes. The same form takes the spare parts it uses (wheel tyres, pickup shoe, brushes, magnet, smoke unit, pantograph — several references separated by commas) and images, which can also be pasted with Ctrl+V. The first image becomes the cover.

Data is entered by hand: railDB does not query external catalogues.

Lists, filters and sorting

Each list can be shown as **cards** («Fichas», with cover and tags: traction, address, RailCom, reviewed, decoder status) or as a **table** («Tabla»), sorted by clicking the headings. «Filtros y orden» (filters and sorting) filters by text, traction, railway, epoch, protocol, decoder, decoder status, RailCom and review status, and sorts by any of those columns or by date.

Review

Each record shows when it was last updated. «Marcar como revisada» (mark as reviewed) records that the record has been checked; if any data changes afterwards, it shows «Con cambios desde la revisión» (changed since review) again. The «Sin revisar» (not reviewed) and «Revisadas» (reviewed) filters help to go through the collection.

Record tabs

Tab	Contents
«Resumen» (summary)	Main data, avatars, decoders and addresses, pending items and notes.
«Modelo real» (real prototype)	Text about the real locomotive or wagon.
«Digital»	Decoders, addresses, functions and the RailESP command station card (not for scenery).
«Taller» (workshop)	Jobs for this piece and running time.
«Repuestos» (spare parts)	Motor and spare parts used.
«Archivos» (files)	Photos, videos and documents of the piece.

«Editar ficha» (edit record) changes the data; at the end of the editor, «Eliminar esta pieza» (delete this piece) removes the piece and its data but keeps its files in the store.

5. Decoders, functions and CVs

Decoders of a piece

In the «Digital» tab, «+ Añadir decoder». Each decoder has a type (from the decoder catalogue) or a free name, a role (main, sound, lights…), sound project, status («Instalado», «Previsto» or «Por confirmar»: installed, planned, to be confirmed), whether it provides RailCom, and one or more addresses, each with its protocol. Mark as **active** the address you use: it is shown in the lists and sent to RailESP. The last address row is always free to add another one.

Functions

Each function has a number (F0 to F68), a name and an icon. The icons are RailESP's and are all shown in the RailESP section. The last row is always free to add another one.

CV list

«Lista de CV» of each decoder keeps the known values. «Anotar una CV» (note a CV) adds or changes a value by hand; reads and writes made by RailESP arrive automatically.

Decoder catalogue

In «Decoders» each decoder type stores its photo and manuals (with language and edition) once, and every piece using it shares them. A type cannot be deleted while a piece still uses it.

6. Workshop and spare parts

Jobs are recorded in the «Taller» tab of each record or in the «Taller» section, which gathers the whole collection. A job has a type, date, details and comment; without a date it stays pending. «Hecho hoy» (done today) closes it with today's date. Filters separate pending and done jobs.

In «Repuestos» each part has a name, reference, category, stock (and whether it is confirmed) and notes, and lists the pieces that use it. Names of the usual part families are always written the same way («Patín T-998», «Aros 7153»…). In a record, the «Repuestos» tab also shows the motor and its reference. «Horas de uso» (running time) adds up the time reported by RailESP.

7. Photos, videos and documents

Files of a piece

The «Archivos» tab uploads photos, videos and documents (up to 128 MB per file with the included configuration). They are grouped by type (photos, videos, sound and decoder, boards, manuals, notes, other). Each photo can carry tags: «Taller», «Real», «Catálogo», «Avatar» and «Portada» (workshop, real, catalogue, avatar, cover). railDB stores each file once even if several pieces use it, and «Quitar de esta pieza» (remove from this piece) does not delete it from the store.

RailESP avatar

On a photo's page, «Generar avatar para RailESP» crops the photo to 480 × 160. With «Fondo transparente» (transparent background), click on the image to choose the colour to remove. The preview shows how it will look on the RailESP web background. «Guardar en la ficha» (save to record) stores it as the avatar; railDB sends RailESP a 192 × 64 thumbnail and the large version.

Videos

railDB stores videos exactly as uploaded and plays them in the browser. Browsers play MP4 with H.264 video (and AAC sound); for other formats railDB warns that they may not play, and they can always be downloaded. To watch them in railDB, upload them as MP4 H.264.

The «Entrada» inbox

To upload many files at once, or very large ones, copy them to the `entrada/` folder of the store. If they are in a subfolder named after the article number or the piece's name, railDB proposes that piece. In «Entrada», choose the destination of each file (a piece, or Documentation with its category) and «Consolidar» (consolidate) moves them into the store. A file shows «Subiendo…» (uploading) until it has been unchanged for one minute.

With Docker the folder is inside the volume. To copy files into it:

```
docker compose cp ./photos web:/var/www/storage/entrada/
docker compose exec web chown -R www-data:www-data /var/www/storage/entrada
```

On a hosting account, upload them by SFTP to `raildb/datos/entrada/`.

Documentation

For manuals, catalogues and documents that do not belong to one piece. Organised by category; each document can be renamed, moved to another category or removed.

8. RailESP synchronization

RailESP is the RailPIC command station. railDB gives it the locomotives marked for the command station, and RailESP reports back what happens on it.

> **Status:** the railDB side is complete and tested with the RailESP simulator (`tools/railesp_sim.py`). A RailESP firmware that includes railDB synchronization is required; until you have it, the RailESP section of railDB can be prepared but no command station will connect.

Registering a command station

1. In «RailESP», under «Centrales» (command stations), type a name and click «Crear token» (create token).
2. Copy the token into the RailESP settings, together with the railDB address. **It is shown only once.**
3. If it is lost or the command station changes, «Nuevo token» (new token) invalidates the previous one; «Desactivar» (disable) stops it from synchronizing.

Sending a locomotive to the command station

In the «Digital» tab, the «Central RailESP» card marks the piece for the command station and sets the short name for RailESP, protocol, speed steps, number of functions, visible functions and mfx UID. Its decoder must have an active address first.

What travels

Data	Direction
Name, address, protocol, speed steps, number of functions, visible functions, mfx UID	both ways
Functions with name and icon, speed curve	both ways
Decoder model and RailCom (display only), icons and avatars	railDB to RailESP
CV reads and writes, running time	RailESP to railDB

Price, box, location, notes, spare parts, workshop and photos other than the avatar never travel.

When and how

RailESP synchronizes when it starts, when «Sincronizar» (synchronize) is pressed, when a locomotive is saved and when a CV read or write finishes. railDB never writes to it on its own. If both sides changed the same data, or railDB cannot decide by itself (a new locomotive without a clear match), the question appears under «Dudas pendientes» (open questions) and is answered in the RailESP dialog. A locomotive leaves the command station only when RailESP reports that it deleted it. The «Registro» (log) records what happened in each synchronization.

> **Connection:** RailESP connects with its token over HTTPS to the railDB address. If railDB runs only on your computer (Docker on `127.0.0.1`), RailESP cannot reach it: it must be on your home network or on the Internet.

9. Backups and updates

Backups

railDB makes backups in PHP, without `mysqldump`: compressed SQL with an end marker; a backup without that marker is never restored.

```
docker compose exec web php /var/www/html/src/bin/backup.php /var/www/storage/copias/manual.sql.gz
docker compose cp web:/var/www/storage/copias/manual.sql.gz .
```

`bin/backup-cron.php` leaves a daily backup in `copias/` (`raildb-YYYY-MM-DD.sql.gz`) plus a CSV saying which piece each file belongs to, and keeps the last `RAILDB_BACKUP_KEEP` (35 if not set). On a hosting account schedule it as a daily task; with Docker, from the computer's task scheduler with `docker compose exec -T web php /var/www/html/src/bin/backup-cron.php`.

The database alone is not enough: also back up the data folder (`files/` and `thumbs/`), which holds the photos and documents.

Restoring

`php bin/restore.php backup.sql.gz --confirmar` replaces every table contained in the backup and checks at the end that the row counts match.

Updating to a new version

1. Back up the database and the files.
2. Replace the code (`src/`; with Docker, download the new version over the folder) without touching `config.php` or the data.
3. Apply the migrations: `php bin/migrate.php` (with Docker, `docker compose exec web php /var/www/html/src/bin/migrate.php`; after changes in `docker/`, `docker compose up -d --build`).
4. Check the version at the foot of the page.

10. Troubleshooting

Symptom	What to do
«Demasiados intentos» (too many attempts) when signing in	Wait 15 minutes. If the password is forgotten, set it again with `create-user.php`.
A page fails on the hosting account	Check `raildb/logs/php-errors.log`. In production errors are never shown on screen.
A video does not play	It is not MP4 H.264. Download it, convert it and upload it again.
An inbox file stays «Subiendo…»	Wait one minute without changes and reload. With Docker, check that it belongs to `www-data`.
RailESP cannot connect	Check the railDB address, HTTPS and the token. A disabled or renewed token no longer works: create a new one and copy it into RailESP.
A locomotive does not appear in the command station	It needs an active address and must be marked in «Central RailESP».
Docker: port 8088 or 3307 is in use	Change the first number under `ports` in `compose.yaml`.

11. Tests, licence and references

With the Docker environment running and a user created, the tests create their own data and remove it at the end. Set `RAILDB_TEST_PASSWORD` in the terminal the same way as `RAILDB_NEW_PASSWORD`:

```
docker compose exec -e RAILDB_TEST_USER=admin -e RAILDB_TEST_PASSWORD web php
/var/www/html/tests/core_test.php
python -m unittest tests.railesp_sync_test -v
```

The second one uses the RailESP simulator `tools/railesp_sim.py`.

railDB is distributed under the GNU General Public License v3.0 (`LICENSE` file). More RailPIC projects: tren.enmicasa.net.

RailPIC · railDB v1.01 · Public manual · revision 1.0 · 2026-11-24