NACARSOFT · EXPLORADOR DE ADABAS

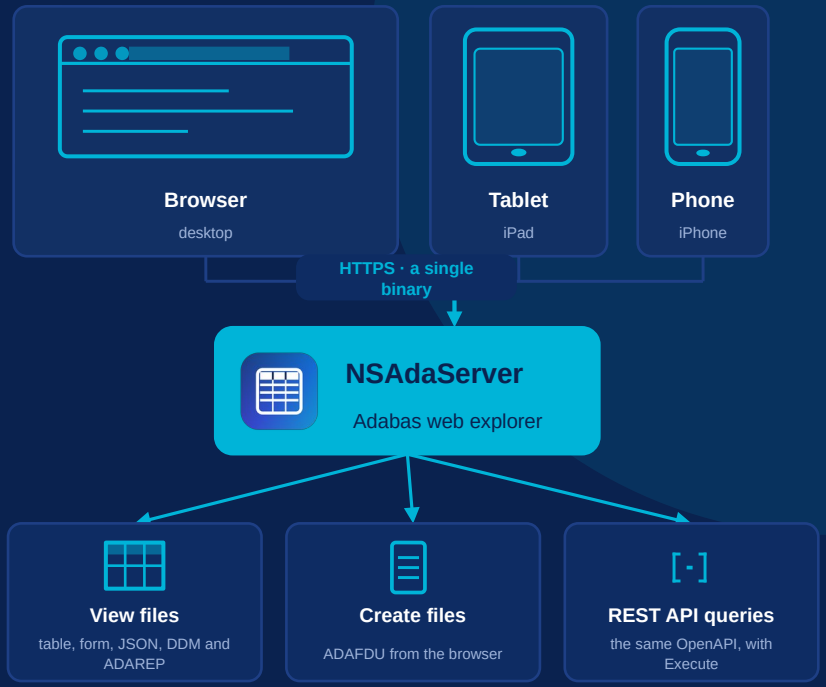
Your Adabas files, in the browser. No client.

NSAdaServer is the web explorer for Adabas files: see the records, create new files and run REST API queries, all from the browser and with the company's own access control.

SEE FEATURES →

View files · Create files · REST API queries

NacarSoft



Client connections to view, create and query through the API

browse · DDM · ADAREP · ADAFDU · OpenAPI

EL PROBLEMA

Looking at Adabas needs a desktop client

Every query is an input file, an ADABAS session and an export: nobody looks at the data when they need it, and the access stays outside the company's control.

01 Local installation

Every operator needs the ADABAS client installed, with its DDM and its credentials.

02 No access control

The file key travels with the person. There is no user, no role, no per-file permission.

03 No traceability

What was queried, with which criteria and by whom is not recorded anywhere.

04 No API

The data lives in a silo: the web, the ERP and the reports share nothing.

- **Per-seat installs**
- **Keys per person**
- **Siloed data**

QUÉ ES

The Adabas explorer, as a website

NSAdaServer is an Adabas file explorer in a single binary: the administrator declares the databases and their permissions, and users browse the files, see the records and work with them from any browser.

- ✓ Databases by name, with a connection pool and reconnection
- ✓ Files by DDM, with the field definition and its indexes
- ✓ Six views of the same file: table, form, JSON, DDM, ADAREP and API
- ✓ Creation of physical files with ADAFDU, without leaving the browser

6 views per file	**0** installations per desk
1 binary, no dependencies	**6** languages

ARQUITECTURA

How a record reaches the screen

1 — Browser

The request travels over HTTPS with the user session.

2 — Session and RBAC

Cookie + JWT: user, group, roles and per-file permissions.

3 — Adabas pool

Connections with idle timeout and automatic reconnection.

4 — Record

The ISN, the fields and the descriptor, exactly as the DDM defines them.

5 — Response

Table, form or JSON, with pagination and search.

⚠ The local DDM leads: the field definition is cached on disk and does not depend on a live connection.

No local client

No DDM on the desk

The same rules for everyone

VER

Viewing files, in six views

Search by field or by ISN, with operators, sorting by descriptor and pagination.

Table

The file as a grid, with the descriptor and the ISN of every record.

Form

One record per form, with the DDM fields and their group elements.

JSON

The same information as JSON, ready to copy or integrate.

DDM

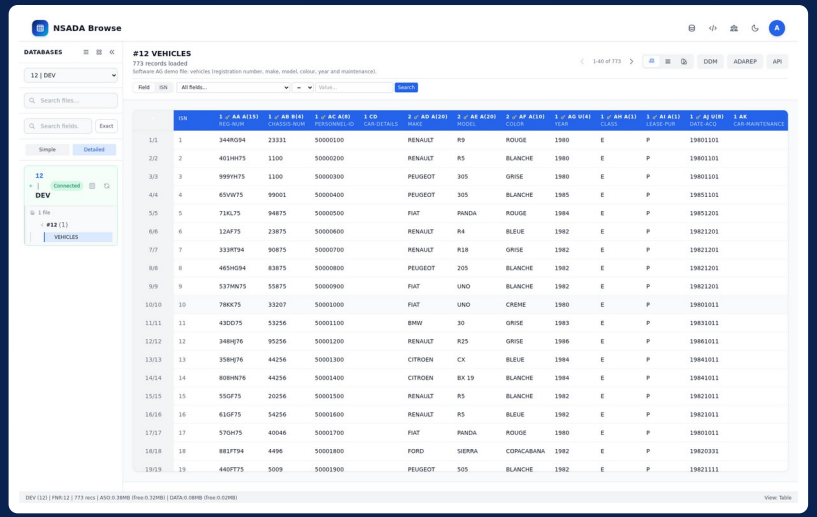
The full definition: level, type, length, description and indexes.

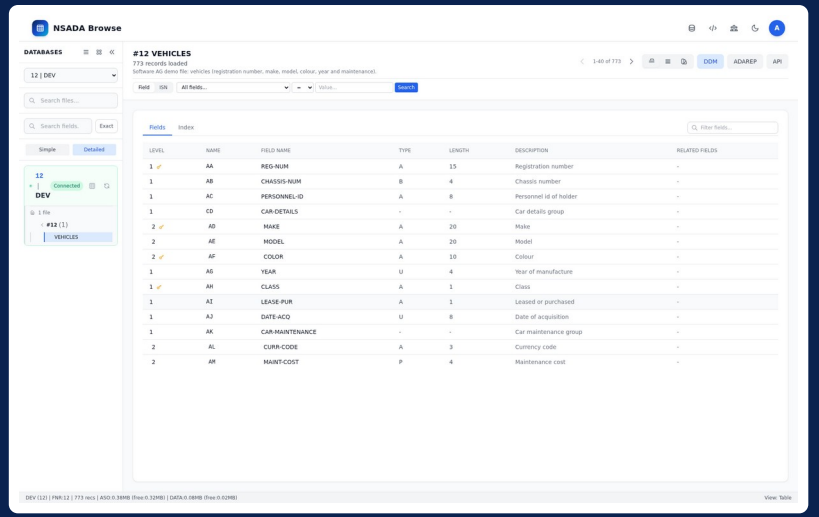
ADAREP

The real output of the server's ADAREP command.

API

The equivalent REST call, with its response and its cURL.



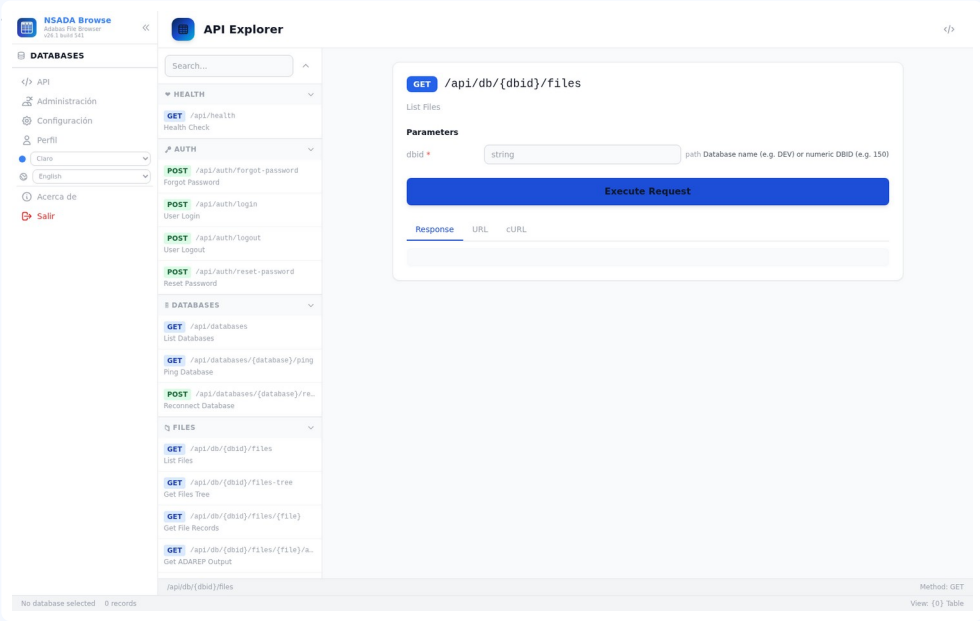


API-REST

REST API queries from the browser itself

The API explorer is a full REST console: the same endpoints the tool.

- OpenAPI 3.0 served and explorable from the application itself
- Endpoints grouped by domain: health, auth, databases, files, records and admin
- Parameters, JSON body and Execute, with the response, the URL and the cURL
- Authentication with the session token or with a scoped API key



API Explorer: the endpoint, its parameters, Execute Request and the response.

The very same calls as the web, in a script: cURL ready to copy.

CREAR

Creating physical files from the browser

ADAFDU

FNR	`250`
Nombre	`MI-FICHERO`
Descripción	`Clientes 2026`
Nivel/Tipo/Long	`1 · A · 20`
Campos	`AA, AB, AC …`

1. **FNR and name**
   The file number and its logical name are chosen in the header.
2. **Default values**
   Level, type, length and the NU, DE, UQ, PE, GR and MU flags.
3. **Field table**
   Each field with its level, name, type, length and description.
4. **ADAFDU**
   NSAdaServer generates the physical file and its DDM on the server.



Fichero físico creado

con su DDM, sus campos y sus permisos

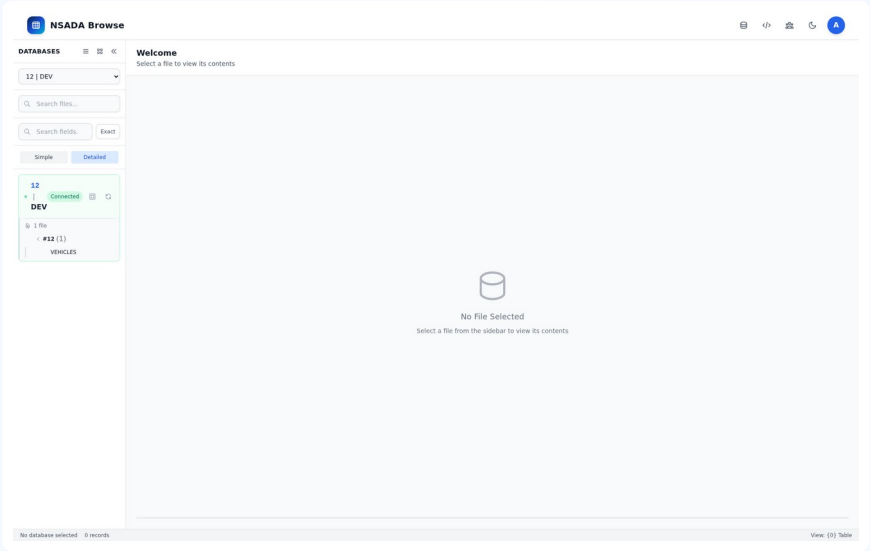
- ✓ **Fichero 250 en la base**
- ✓ **DDM MI-FICHERO generado**
- ✓ **Visible para quien tenga permiso**
- ✓ **Listo para escribir y leer**

A new file, its DDM and its permissions, without hand-writing a DDM or requesting an input file.

CONEXIONES

Databases declared in the configuration

- Each database with name, DBID, connection string and configuration file
- Several databases in the same instance, each with its own pool
- Live state: connected, stopped for inactivity or in error
- Reconnect, stop and ping from the panel itself



The database tree with its state and the files of each one.

Each file's DDM is stored on disk: the list does not depend on the connection.

PERMISOS

Permissions per file, not per person

- Users → groups → roles → permissions per file
- Read, write and delete per file, with a wildcard
- File listings and records are filtered by permissions
- API keys with scopes per integration, with revocation and audit

Access control

- RBAC by roles and groups
- Permission per file (read/write/delete)
- Listings filtered by permissions
- Access by token or API key

API keys

- Scope per integration
- Configurable expiry
- Revoke and reactivate
- Usage audit

Session

- HS256 JWT in a cookie
- Passwords with bcrypt
- Recovery by email with a code
- Lockout after failed attempts

config.yaml 0600 · bcrypt · JWT HS256 · per-operation audit · 30-day trial

OPERACIÓN

What you can do with a file

Read

Records by pagination, with search by field, by ISN and by range.

Update

Update a record by ISN, with its audit trail.

Delete

Delete by ISN, one or many, with a record.

Empty

Empty a whole file, with the matching permission.

Create

A new physical file with ADAFDU, with its fields and its DDM.

Drop

Remove the physical file and its definition from the DDM.

Todo lo que se hace con un dato queda registrado: acción, usuario y resultado.

GETTING STARTED

From zero to queryable files, in three steps

1 Create the environment

nsadaserver init /opt/nsadaserver: it creates config.yaml, the SQLite user database, the logs and the admin user.

2 Declare the databases

In config.yaml: name, DBID, connection string and the DDM file of each database.

3 Register the users

Administration → Users and groups: each user only sees the files their role allows.

REQUIREMENTS

- Linux (RHEL 8 / RHEL 9)
- Native Adabas client (CGO)
- A single binary, no dependencies
- Reverse proxy with HTTPS in production
- Web panel on port 11010

```
nsadaserver init /opt/nsadaserver   →   nsadaserver start /opt/nsadaserver   →   http://localhost:11010
```

FREQUENTLY ASKED

The questions we hear most

Is the Adabas client needed?

Only on the server, and optionally: the DDM is stored on disk, so the file list and its definition are shown without a live connection.

Can the data be queried through an API?

Yes: the REST API with OpenAPI 3.0 is published, explorable from the panel itself and consumable with a token or scoped API keys.

Can new files be created?

Yes: from the browser, with the ADAFDU form: FNR, name, default values and the field table. The DDM is generated too.

Who sees which file?

The permission lives on the role, per file, with read, write and delete. Listings and records are filtered with those permissions.

Is what they do recorded?

Yes: every operation is logged with its user and its result, and the API keys have their own usage log.

Does the user need the DDM on their desk?

No: the DDM travels with the file definition and the browser shows it. The operator installs nothing.

Your company Adabas, open in the browser

Try NSAdaServer with your databases: we set it up with your files and watch it run.

Guided demo | Try it in your environment | Let's talk about your case

nacarsoft.com · support@nacarsoft.com

NacarSoft