Mantenova

B2B SaaS for technical inspections and industrial maintenance

Idunn Freya

May 2026

STATUS IN DEVELOPMENT (V2) **START** February 2025

VERSION v2.0 — Phase 4 in progress

REPO Private (available on request)

URL Not published (V2 in development)

SUMMARY

Mantenova is a B2B SaaS platform that connects facility-owning companies with technical inspection firms, automates the tracking of regulatory deadlines, and centralizes document management with preparation for qualified digital signatures. It originates from direct observation of the industrial electrical sector, where inspection management still relies on spreadsheets, emails, and phone calls. V1 was delivered as the final project of an Advanced Vocational Training programme; V2 is a professional rewrite planned across eight phases.

B2B SaaS | Web platform · API · PWA | Django · PostgreSQL · Docker

TECH STACK

BACKEND	Django 5.1 · Python 3.12 · Django REST Framework · drf-spectacular
FRONTEND	HTML · CSS (custom design system) · JavaScript · PWA
DATABASE	PostgreSQL 16 · Redis 7
INFRASTRUCTURE	Nginx · Gunicorn · GitHub Actions (CI/CD)
CONTAINERIZATION	Multi-environment Docker Compose (dev · staging · prod)

TESTING	pytest · pytest-django · factory_boy · ruff · 215 tests

GOALS

- V1 delivered as the final project of the Advanced Vocational Training in Web Application Development (2025)
- Domain analysis and V2 planning across eight phases with objective completion criteria
- Phase 0 — Multi-environment Docker infrastructure, CI/CD, Celery + Redis
- Phase 1 — 60 models across 14 apps, seeded catalogs, ES/EN i18n
- Phase 2 — Business logic in services, role-based permissions with overrides
- Phase 3 — REST API documented with OpenAPI/Swagger (215 cumulative tests)
- Phase 4 — Views, role-based dashboards, CRUD forms, and view tests
- Phase 5 — Asynchronous tasks (alerts, scraping, emails, backups, PDFs)
- Phase 6 — Hardened production deployment (SSL, monitoring, security)
- Phase 7 — Integration with a qualified digital signature provider (eIDAS)

ARCHITECTURE

Modular architecture organized by functional domain on Django. The data model addresses the complexity of the industrial sector through **dynamic catalogs** that allow multiple facility and inspection types to be managed without modifying code. Business logic lives in `services.py` per application, so views and API endpoints only orchestrate. The permission system operates at the role level, with the option to grant additional access on specific resources through `PermissionOverride` (overrides extend, never restrict). The REST API is documented with OpenAPI/Swagger to power the PWA and future integrations. The whole environment is orchestrated with Docker Compose in three configurations (dev, staging, prod) and validated on every push by GitHub Actions.

MODULE / APP	DESCRIPTION	STATUS
`accounts`	Users, authentication, 5 roles + `PermissionOverride` per resource	✓ Complete
`empresas`	Owner and inspection companies under a single model with a `type` field	✓ Complete
`zonas`	Postal codes and geographical zones (shared between inspectors and facilities)	✓ Complete
`instalaciones`	Base facility + 25 type-specific models · `TipoInstalacion` catalog	✓ Complete
`inspecciones`	Single model + 11 detail models · 34 types · 120 combinations · 8 states	✓ Complete
`inspectores`	Inspectors, qualifications, work zones, availability slots	✓ Complete
`documentos`	Versioned document management · nullable digital signature metadata	✓ Complete
`mensajeria`	Conversations and messages linked to company, facility, or inspection	✓ Complete
`notificaciones`	15 notification types in-app and email (async engine pending Phase 5)	◑ Model OK · async pending

MODULE / APP	DESCRIPTION	STATUS
`normativa`	Technical regulations catalog · change alerts (scraper pending)	◑ Model OK · scraper pending
`noticias`	Industry news (periodic loading pending Phase 5)	◑ Model OK · scraper pending
`pages`	Public pages with content blocks editable from the admin	◑ Model OK · views in progress
`contacto`	Public contact form with attachment	✓ Complete
`auditoria`	Action and impersonation log (root)	✓ Complete

MILESTONES

Feb – Jun 2025

V1 delivered as the final project of the Advanced Vocational Training

First functional version with multi-role registration, dashboards per user type, facility and inspection management, internal messaging, and regulations catalog. Served as concept validation.

Mar 2026

V2 analysis and planning

Critical review of V1, industrial domain modeling, and eight-phase plan with objective completion criteria. Key decisions: dynamic catalogs, single inspection model, permissions with overrides, digital signature prepared from the data layer.

Apr – May 2026

Phases 0 – 3 completed (215 tests passing)

Multi-environment Docker infrastructure, CI/CD, 60 models across 14 apps, business logic in services, permission mixins, REST API documented with OpenAPI/Swagger. TDD applied phase by phase.

May 2026 →

Phase 4 — Views, panels, and frontend

Design system implemented in CSS, base templates, authentication, and initial public pages closed. In progress: role-based dashboards, CRUD per user type, marketplace search, full forms, and view tests.

2026 – 2027

Phases 5, 6, and 7 — Production and legal compliance

Asynchronous tasks (alerts, scraping, emails, backups, PDF generation), hardened production deployment (SSL, monitoring, security), and integration with a qualified digital signature provider (eIDAS).

TECHNICAL DECISIONS AND NOTES

DECISION

Dynamic catalogs in the database for facility types (25) and inspection types (34), instead of hardcoded enums. Adding a new type is done from the admin panel, with no code migrations or deployments. The `InstalacionTipoInspeccion` table holds the 120 combinations that define which inspection applies to which facility and at what periodicity.

DECISION

Single inspection model + 11 detail models only for the types that require extra fields (OCA, earthing, RITE, RSCIEI, pressure, refrigeration, gases, APQ, cranes, legionella, tightness), instead of 34 parallel models. Drastic reduction in complexity without loss of functionality.

NOTE

Preparation for qualified digital signatures (eIDAS) integrated from Phase 1: document models carry nullable signature metadata from the start. This allows the provider to be contracted and integrated in Phase 8 without restructuring the data layer.

ATTENTION

Real integration with a qualified digital signature provider (Phase 7) requires external contracting with associated costs and eIDAS compliance verification. The architecture is prepared, but the production rollout of this feature depends on that provider selection.

INSTALLATION AND DEPLOYMENT

Private repository. Deployment in Docker containers with multi-environment configuration (dev, staging, prod) and orchestration through a standardized Makefile.

```
# Local development (when V2 is accessible)
git clone git@github.com:idunnmaria/mantenova.git
cd mantenova
cp .env.example .env.dev
make dev-build
make dev-up
make migrate
make seed
make test
```