

Mantenova

B2B SaaS for technical inspections and industrial maintenance

Idunn Freya

June 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, validated in 2025, proved the viability of the concept; V2 is a professional rewrite planned across nine phases, with admin-editable content for customization per operating company.

B2B SaaS · Configurable multi-tenant

Web platform · API · PWA

Django 6 · Python 3.14 · PostgreSQL · Docker

TECH STACK

BACKEND	Django 6.0 · Python 3.14 · Django REST Framework · drf-spectacular
FRONTEND	HTML · CSS (custom design system) · JavaScript · PWA · django-tinymce
DATABASE	PostgreSQL 16 · Redis 7
INFRASTRUCTURE	Nginx · Gunicorn · GitHub Actions (CI/CD with pip-audit + Dependabot)

CONTAINERIZATION	Multi-environment Docker Compose (dev · staging · prod)
TESTING AND QUALITY	pytest 9.1 · factory_boy · ruff · 129 tests · 71% coverage (fail_under=70 floor)

GOALS

- V1 functional delivered and validated in 2025
- Domain analysis and V2 planning across nine phases with objective completion criteria
- Phase 0 – Multi-environment Docker infrastructure, CI/CD, Celery + Redis, coverage floor
- Phase 1 – 60 models across 14 apps, seeded catalogs against current regulations, ES/EN i18n
- Phase 2 – Business logic in services, role-based permissions with overrides, anti-IDOR
- Phase 3 – REST API documented with OpenAPI/Swagger
- Phase 4 – Views, role-based dashboards, CRUD forms, and view tests (public pages already closed)
- Phase 5 – Specialized UI components, inspector calendar, real-time messaging
- Phase 6 – Asynchronous tasks (alerts, regulations and news scraping, emails, backups, PDFs)
- Phase 7 – Hardened production deployment (SSL, monitoring, security hardening, GDPR)
- Phase 8 – 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. The **visual content of public pages** (texts, sections, site configuration) lives in the database via `SiteConfig` (global singleton) and `ContentBlock` (per-page editable blocks with `django-tinymce`), so customizing the platform for an operating company requires no development. Business logic lives in `services.py` per application; 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. The whole environment is orchestrated with Docker Compose in three configurations (dev, staging, prod) and validated on every push by GitHub Actions (lint, tests, minimum 70% coverage, dependency audit).

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 6)	● Model OK · async pending

MODULE / APP	DESCRIPTION	STATUS
normativa	Catalog verified against official sources · change alerts (scraper pending)	● Model OK · scraper pending
noticias	Industry news (periodic loading pending Phase 6)	● Model OK · scraper pending
pages	Admin-editable ContentBlock · public pages complete	✓ Complete
contacto	Public contact form with attachment	✓ Complete
auditoria	Action and impersonation log (root)	✓ Complete

MILESTONES

Feb - Jun 2025

V1 delivered and validated

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

Mar 2026

V2 analysis and planning

Critical review of V1, industrial domain modeling, and nine-phase plan with objective completion criteria. Key decisions: dynamic catalogs, single inspection model, permissions with overrides, admin-editable content, digital signature prepared from the data layer. Explicit quality backlog (Q1-Q13).

Apr - May 2026

Phases 0 - 3 completed

Multi-environment Docker infrastructure, CI/CD with coverage floor (fail_under=70) and dependency auditing (pip-audit + Dependabot), 60 models across 14 apps, business logic in services, permission mixins with anti-IDOR validation, 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 public pages complete (with admin-editable content). Stack upgraded to Python 3.14 + Django 6.0. In progress: role-based dashboards, CRUD per user type, marketplace search, full forms, and view tests.

2026 - 2027

Phases 5, 6, 7, and 8 - Production and legal compliance

Specialized UI components (inspector calendar, real-time messaging), asynchronous tasks (alerts, scraping, emails, backups, PDF generation), hardened production deployment (SSL, monitoring, hardening, GDPR), 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. Catalogs verified against current regulations (RD 164/2025 RSCIEI, RD 355/2024 lifts, RD 178/2021 RITE, RD 552/2019 refrigeration, RD 487/2022 + RD 614/2024 legionella, RD 390/2021 + RD 659/2025 energy efficiency).

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.

DECISION

Public page content editable from the admin via `SiteConfig` (global singleton) and `ContentBlock` (per-page blocks, bilingual ES/EN, with `django-tinymce` for rich editing). This prepares the platform for per-company customization (white-label mode) without additional development.

DECISION

Language-independent tests: assertions are made on stable error codes (`code` of `ValidationError`), not on translatable text. Translating a message never breaks a test. Coverage floor `fail_under=70` enforced in CI, anti-IDOR validation on API writes (dedicated security tests), pip-audit + Dependabot operational.

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. Architectural decisions are recorded as ADRs (Architecture Decision Records) in `docs/adr/`.

ATTENTION

Real integration with a qualified digital signature provider (Phase 8) requires external contracting with associated costs and eIDAS compliance verification. GDPR conformance (Phase 7) requires a privacy policy, consent management, and a right-to-erasure procedure. The architecture is prepared, but the production rollout of these features depends on those business decisions.

INSTALLATION AND DEPLOYMENT

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

```
# 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
make migrate
make seed-all
make test
```