Mantenova

SaaS B2B para inspecciones técnicas y mantenimiento industrial

Idunn Freya

Junio 2026

ESTADO	EN DESARROLLO (V2)	**INICIO**	Febrero 2025
VERSIÓN	v2.0 — Fase 4 en curso	**REPO**	Privado (bajo solicitud)
URL	No publicada (V2 en desarrollo)		

RESUMEN

Mantenova es una plataforma SaaS B2B que conecta empresas propietarias de instalaciones con empresas de inspección técnica, automatiza el seguimiento de plazos normativos reglamentarios y centraliza la gestión documental con preparación para firma digital cualificada. Nace de la observación directa del sector eléctrico industrial, donde la gestión de inspecciones aún depende de hojas de cálculo, correos y llamadas. La V1, validada en 2025, demostró la viabilidad del concepto; la V2 es una reescritura profesional planificada en nueve fases, con contenido editable desde administración para personalización por empresa operadora.

SaaS B2B · Multi-tenant configurable

Plataforma Web · API · PWA

Django 6 · Python 3.14 · PostgreSQL · Docker

STACK TECNOLÓGICO

BACKEND	Django 6.0 · Python 3.14 · Django REST Framework · drf-spectacular
FRONTEND	HTML · CSS (design system propio) · JavaScript · PWA · django-tinymce
BASE DE DATOS	PostgreSQL 16 · Redis 7
INFRAESTRUCTURA	Nginx · Gunicorn · GitHub Actions (CI/CD con pip-audit + Dependabot)

CONTENERIZACIÓN	Docker Compose multi-entorno (dev · staging · prod)
TESTING Y CALIDAD	pytest 9.1 · factory_boy · ruff · 129 tests · 71% cobertura (suelo fail_under=70)

OBJETIVOS

- V1 funcional entregada y validada en 2025
- Análisis de dominio y planificación V2 en nueve fases con criterios de cierre objetivos
- Fase 0 — Infraestructura Docker multi-entorno, CI/CD, Celery + Redis, suelo de cobertura
- Fase 1 — 60 modelos en 14 apps, catálogos sembrados contra normativa vigente, i18n ES/EN
- Fase 2 — Lógica de negocio en services, permisos por rol con overrides, anti-IDOR
- Fase 3 — API REST documentada con OpenAPI/Swagger
- Fase 4 — Vistas, paneles por rol, formularios CRUD y tests de vistas (páginas públicas ya cerradas)
- Fase 5 — Componentes UI especializados, calendario inspector, mensajería en tiempo real
- Fase 6 — Tareas asíncronas (alertas, scraping de normativa y noticias, emails, backups, PDFs)
- Fase 7 — Despliegue a producción endurecido (SSL, monitorización, hardening, RGPD)
- Fase 8 — Integración con proveedor de firma digital cualificada (eIDAS)

ARQUITECTURA

Arquitectura modular organizada por dominio funcional sobre Django. El modelo de datos resuelve la complejidad del sector industrial mediante **catálogos dinámicos** que permiten gestionar múltiples tipos de instalación e inspección sin modificar código. El **contenido visual de las páginas públicas** (textos, secciones, configuración del sitio) vive en base de datos a través de `SiteConfig` (singleton global) y `ContentBlock` (bloques editables por página con `django-tinymce`), de modo que personalizar la plataforma para una empresa operadora no requiere desarrollo. La lógica de negocio vive en `services.py` por aplicación, vistas y endpoints de API solo orquestan. El sistema de permisos opera a nivel de rol con la posibilidad de conceder accesos adicionales sobre recursos concretos mediante `PermissionOverride` (los overrides amplían, nunca restringen). La API REST está documentada con OpenAPI/Swagger. Todo el entorno está orquestado con Docker Compose en tres configuraciones (dev, staging, prod) y se valida en cada push con GitHub Actions (lint, tests, cobertura mínima 70%, auditoría de dependencias).

MÓDULO / APP	DESCRIPCIÓN	ESTADO
`accounts`	Usuarios, autenticación, 5 roles + `PermissionOverride` por recurso	✓ Completo
`empresas`	Empresas propietarias e inspección bajo un modelo único con campo `tipo`	✓ Completo
`zonas`	Códigos postales y zonas geográficas (compartido entre inspectores e instalaciones)	✓ Completo
`instalaciones`	Instalación base + 25 modelos específicos por tipo · catálogo `TipoInstalacion`	✓ Completo
`inspecciones`	Modelo único + 11 modelos de detalle · 34 tipos · 120 combinaciones · 8 estados	✓ Completo
`inspectores`	Inspectores, cualificaciones, zonas de trabajo, slots de disponibilidad	✓ Completo
`documentos`	Gestión documental versionada · metadatos de firma digital nullable	✓ Completo
`mensajeria`	Conversaciones y mensajes vinculados a empresa, instalación o inspección	✓ Completo
`notificaciones`	15 tipos de notificación in-app y email (motor async pendiente Fase 6)	◑ Modelo OK · async pendiente
`normativa`	Catálogo verificado contra fuentes oficiales · alertas de cambios (scraper pendiente)	◑ Modelo OK · scraper pendiente
`noticias`	Noticias del sector (carga periódica pendiente Fase 6)	◑ Modelo OK · scraper pendiente

MÓDULO / APP	DESCRIPCIÓN	ESTADO
`pages`	`ContentBlock` editable desde admin · páginas públicas completas	✓ Completo
`contacto`	Formulario público de contacto con adjunto	✓ Completo
`auditoria`	Log de acciones y de impersonaciones (root)	✓ Completo

HITOS

Feb – Jun 2025

V1 entregada y validada

Primera versión funcional con registro multi-rol, paneles por tipo de usuario, gestión de instalaciones e inspecciones, mensajería interna y catálogo de normativa. Sirvió como validación del concepto antes de plantear la reescritura profesional.

Mar 2026

Análisis y planificación V2

Revisión crítica de la V1, modelado del dominio industrial y plan de nueve fases con criterios objetivos de cierre. Decisiones clave: catálogos dinámicos, modelo único de inspección, permisos con overrides, contenido editable desde admin, firma digital preparada desde la capa de datos. Backlog de calidad explícito (Q1–Q13).

Abr – May 2026

Fases 0 – 3 completadas

Infraestructura Docker multi-entorno, CI/CD con suelo de cobertura (fail_under=70) y auditoría de dependencias (pip-audit + Dependabot), 60 modelos en 14 apps, lógica de negocio en services, mixins de permisos con validación anti-IDOR, API REST documentada con OpenAPI/Swagger. TDD aplicado fase a fase.

Mayo 2026 →

Fase 4 — Vistas, paneles y frontend

Design system implementado en CSS, plantillas base, autenticación y páginas públicas completas (con contenido editable desde admin). Stack actualizado a Python 3.14 + Django 6.0. En curso: dashboards por rol, CRUD por tipo de usuario, búsqueda marketplace, formularios completos y tests de vistas.

2026 – 2027

Fases 5, 6, 7 y 8 — Producción y cumplimiento legal

Componentes UI especializados (calendario inspector, mensajería en tiempo real), tareas asíncronas (alertas, scraping, emails, backups, generación de PDFs), despliegue a producción endurecido (SSL, monitorización, hardening, RGPD) e integración con proveedor de firma digital cualificada (eIDAS).

DECISIONES TÉCNICAS Y NOTAS

DECISIÓN

Catálogos dinámicos en base de datos para los tipos de instalación (25) y de inspección (34), en lugar de enums hardcoded. Añadir un tipo nuevo se hace desde el panel de administración, sin migraciones de código ni despliegues. La tabla `InstalacionTipoInspeccion` mantiene las 120 combinaciones que definen qué inspección aplica a qué instalación y con qué periodicidad. Catálogos verificados contra normativa vigente (RD 164/2025 RSCIEI, RD 355/2024 ascensores, RD 178/2021 RITE, RD 552/2019 frigorífico, RD 487/2022 + RD 614/2024 legionella, RD 390/2021 + RD 659/2025 eficiencia energética).

DECISIÓN

Modelo único de inspección + 11 modelos de detalle solo para los tipos que requieren campos extra (OCA, tierras, RITE, RSCIEI, presión, frigorífico, gases, APQ, grúa, legionella, estanqueidad), en lugar de 34 modelos paralelos. Reducción drástica de complejidad sin pérdida de funcionalidad.

DECISIÓN

Contenido de páginas públicas editable desde administración mediante `SiteConfig` (singleton global) y `ContentBlock` (bloques por página, bilingüe ES/EN, con `django-tinymce` para edición rica). Esto prepara la plataforma para personalización por empresa operadora (modo marca blanca) sin desarrollo adicional.

DECISIÓN

Tests independientes del idioma: las aserciones se hacen sobre códigos de error estables (`code` de `ValidationError`), no sobre texto traducible. Traducir un mensaje nunca rompe un test. Suelo de cobertura `fail_under=70` enforced en CI, validación anti-IDOR en escrituras de API (tests de seguridad propios), pip-audit + Dependabot operativos.

NOTA

Preparación para firma digital cualificada (eIDAS) integrada desde la Fase 1: los modelos de documentos llevan metadatos de firma nullables desde el origen. Esto permite contratar e integrar el proveedor en la Fase 8 sin reestructurar la capa de datos. Las decisiones arquitectónicas se registran como ADRs (Architecture Decision Records) en `docs/adr/` .

ATENCIÓN

La integración real con proveedor de firma digital cualificada (Fase 8) requiere contratación externa con coste asociado y verificación de cumplimiento eIDAS. La conformidad RGPD (Fase 7) implica política de privacidad, gestión de consentimiento y procedimiento de derecho al olvido. La arquitectura está preparada, pero la puesta en producción de estas funcionalidades depende de esas decisiones de negocio.

INSTALACIÓN Y DESPLIEGUE

Repositorio privado. Despliegue en contenedores Docker con configuración multi-entorno (dev, staging, prod) y orquestación mediante Makefile estandarizado replicable entre proyectos.

```
# Desarrollo local (cuando V2 sea accesible)
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
```