

Syntropy

Infraestructura web profesional construida como sistema integral

Idunn Freya

octubre 2024

ESTADO **TERMINADO** **INICIO** octubre 2024 **VERSIÓN** v1.0.0
REPO Privado · disponible bajo solicitud **URL** idunnfreya.com

RESUMEN

Syntropy es una infraestructura web profesional diseñada y desplegada de forma integral – backend, frontend, DevOps, seguridad y SEO – como un sistema cohesivo donde cada decisión técnica responde a un criterio real de producción. Construida desde cero porque ninguna plantilla resuelve el problema de presentar trabajo que cruza disciplinas sin aplanar su complejidad.

Portfolio profesional

Web App

Django · PostgreSQL · Docker

STACK TECNOLÓGICO

BACKEND	Django 4.2 LTS · Python 3.x
FRONTEND	Tailwind CSS · JavaScript vanilla
BASE DE DATOS	PostgreSQL 15
INFRAESTRUCTURA	Hetzner VPS · GitHub Actions CI/CD
CONTENERIZACIÓN	Docker · Docker Compose multi-entorno
TESTING	pytest · 104+ tests

OBJETIVOS

- Arquitectura modular Django con siete aplicaciones independientes
- Sistema bilingüe completo (ES/EN) con django-modeltranslation

- Pipeline CI/CD automatizado con deploy a producción
- SEO bilingüe: hreflang, Open Graph, Twitter Card, sitemap
- Seguridad en producción: iptables, rate limiting, honeypot, anti-spam
- Backups automáticos diarios con retención de 14 días
- Analítica propia sin dependencias de terceros
- Sistema de visibilidad por proyecto con lead generation
- Poblado completo de contenido en todas las secciones
- Pipeline de procesamiento de imagen con coherencia cromática

ARQUITECTURA

Un portfolio multidisciplinar tiene un problema estructural: el contenido no es homogéneo. Código, planos CAD, piezas audiovisuales y documentos técnicos no conviven bien en plantillas diseñadas para galerías de imágenes o blogs. Syntropy se construye desde cero porque el problema lo requiere — no por capricho técnico.

El nombre no es casual. Sintropía es el principio opuesto a la entropía: orden que emerge de la complejidad. Esa es la tesis del proyecto y del perfil que presenta.

MÓDULO / APP	DESCRIPCIÓN	ESTADO
core	Funcionalidad compartida, templates base, middleware, context processors	✓ Completo
home	Página principal con sistema orbital interactivo	✓ Completo
sections	Cinco categorías profesionales con slugs bilingües	✓ Completo
projects	Gestión de proyectos con bloques de contenido y galería multimedia	✓ Completo
bio	Información profesional con secciones expandibles	✓ Completo
contact	Formulario de contacto y solicitud de acceso a contenido privado	✓ Completo
legal	Páginas legales con renderizado Markdown	✓ Completo

HITOS

Oct 2024

Diseño y arquitectura

Definición de stack, estructura de apps Django, modelos de datos y sistema de contenido heterogéneo. Configuración Docker multi-entorno.

Dic 2024

Core funcional

Backend completo con siete apps, sistema bilingüe, panel de administración personalizado y 88+ tests. Entornos de desarrollo y staging operativos.

Ene 2025

Deploy a producción

Despliegue en Hetzner VPS. Pipeline CI/CD con GitHub Actions, hardening de seguridad, backups automáticos, SSL y email transaccional vía Brevo.

Feb-Mar 2025

Refinamiento y SEO

Responsive completo, SEO bilingüe (hreflang, OG, sitemap), analítica propia con dashboard de admin, favicon, pipeline de imagen y 104+ tests.

2025-actualidad

Poblado de contenido

Introducción de proyectos, biografía, textos legales y recursos multimedia en las cinco secciones del portfolio.

DECISIONES TÉCNICAS Y NOTAS

DECISIÓN

Django sobre frameworks SPA. El contenido es fundamentalmente estático y multilingüe. Un framework como React o Vue añadiría complejidad sin beneficio real: peor SEO, mayor tiempo de carga y una dependencia de build innecesaria para un sitio que no requiere interactividad en tiempo real.

DECISIÓN

Tailwind CSS + JavaScript vanilla en frontend. Utility-first permite iterar rápido sin acumular CSS muerto. JavaScript vanilla porque las interacciones son puntuales y no justifican el peso de un framework.

DECISIÓN

Análítica propia en lugar de Google Analytics. Control total sobre los datos de visitas, sin cookies de terceros ni dependencia de servicios externos. Implementada como middleware Django con dashboard en el admin.

NOTA

Docker Compose con overlays por entorno (dev, staging, producción) permite replicar exactamente el entorno de producción en local. La paridad de entornos elimina la categoría entera de bugs de "funciona en mi máquina".

ATENCIÓN

UFW y Docker tienen conflictos conocidos con iptables. La solución es gestionar reglas directamente en la cadena DOCKER-USER, no a través de UFW.

INSTALACIÓN Y DESPLIEGUE

Repositorio privado. Para acceder al código fuente y documentación técnica detallada, utilice el formulario de solicitud de información en idunnfreya.com.

```
# Entorno de desarrollo (requiere Docker)
docker compose -f docker-compose.yml -f docker-
compose.dev.yml --env-file .env.dev up -d --build

# Deploy a producción (desde local)
./scripts/deploy.sh

# Tests
docker compose exec web pytest --cov
```