

KRONETIC V1

Manual Técnico e Instalación

Sistema avanzado de control horario inteligente

Versión: V1

Fecha: 2026

© 2026 · Ciclotic S.L. · Sistema avanzado AI

<https://kronetic.com>

Índice

1. Introducción técnica
 2. Arquitectura general
 3. Requisitos del servidor
 4. Instalación inicial
 5. Estructura multiempresa
 6. Creación de empresas
 7. Sistema de migraciones
 8. Sistema PUSH
 9. Sistema de alertas
 10. Geolocalización
 11. Auditoría
 12. Seguridad
 13. Copias de seguridad
 14. Actualizaciones
 15. Resolución de incidencias
 16. Checklist de puesta en producción
 17. Roadmap Kronetic
-

1. Introducción técnica

Objetivo

Este documento describe la arquitectura técnica, instalación, configuración, mantenimiento y operación de Kronetic V1.

Está dirigido a:

- Técnicos de implantación
 - Administradores de sistemas
 - Equipo de desarrollo
 - Personal de soporte
 - Responsables de mantenimiento
-

2. Arquitectura general

Descripción

Kronetic es una aplicación web desarrollada en PHP 8.2 orientada a control horario avanzado, gestión de incidencias, ausencias, auditoría y notificaciones.

La solución utiliza arquitectura multiempresa mediante aislamiento de bases de datos por empresa.

Tecnologías principales

- PHP 8.2
- MariaDB
- HTML5
- JavaScript
- Tailwind CSS
- Service Worker
- PWA
- Web Push Notifications

Componentes

Frontend

Interfaz web responsive optimizada para:

- Escritorio
- Tablet
- Smartphone
- PWA

Backend

Aplicación PHP responsable de:

- Autenticación

- Control horario
- Gestión de incidencias
- Gestión de ausencias
- Auditoría
- Alertas
- Informes

Base principal

Almacena:

- Empresas
- Usuarios globales
- Configuración general
- Información de tenants

Bases tenant

Cada empresa dispone de una base de datos independiente.

Ventajas:

- Seguridad
 - Escalabilidad
 - Aislamiento de información
 - Simplificación de copias de seguridad
-

3. Requisitos del servidor

Sistema operativo recomendado

- Linux
- Debian
- Ubuntu

Servidor web

- Apache 2.4+
- HTTPS obligatorio

PHP

Versión mínima:

PHP 8.2

MariaDB

Versión recomendada:

MariaDB 10.6 o superior

Extensiones PHP necesarias

- pdo_mysql
 - openssl
 - mbstring
 - json
 - curl
 - sodium
 - fileinfo
-

4. Instalación inicial

Subida de archivos

Subir el código fuente al servidor mediante:

- Git
- FTP
- SFTP
- Panel de hosting

Configuración .env

Configurar:

- APP_KEY
- Configuración base principal
- VAPID_PUBLIC_KEY
- VAPID_PRIVATE_KEY
- ALERT_CRON_KEY

HTTPS

Es obligatorio utilizar certificados SSL válidos.

5. Estructura multiempresa

Concepto

Kronetic utiliza arquitectura multiempresa basada en:

Una base principal + una base de datos por empresa.

Base principal

Gestiona:

- empresas
- empresas_bbdd
- usuarios globales
- configuración

Bases tenant

Cada empresa almacena:

- fichajes
 - pausas
 - incidencias
 - ausencias
 - auditoría
 - alertas
-

6. Creación de empresas

Flujo de creación

Alta empresa

↓

Creación base de datos

↓

Ejecución migraciones

↓

Creación administrador

↓

Sistema operativo

Proceso técnico

1. Registrar empresa.
 2. Crear base de datos tenant.
 3. Ejecutar SchemaManager.
 4. Aplicar migraciones.
 5. Crear usuario administrador.
 6. Validar acceso.
-

7. Sistema de migraciones

Ubicación

database/migrations

Control de versiones

Tabla:

schema_migrations

Componente principal

SchemaManager

Responsable de:

- Detectar migraciones pendientes.
- Ejecutar scripts SQL.
- Registrar versiones aplicadas.

Procedimiento recomendado

1. Crear archivo SQL.
 2. Guardarlo en database/migrations.
 3. Ejecutar migraciones.
 4. Verificar schema_migrations.
-

8. Sistema PUSH

Arquitectura

Kronetic utiliza:

- Service Worker
- Push Subscription
- Web Push
- VAPID

Componentes

Service Worker

Archivo:

sw.js

Suscripciones

Tabla:

push_subscriptions

Claves VAPID

Generadas mediante:

tools/generate_vapid.php

Variables necesarias

- VAPID_PUBLIC_KEY
 - VAPID_PRIVATE_KEY
-

9. Sistema de alertas

Objetivo

Detectar automáticamente situaciones que requieren revisión.

Componente principal

AlertService

Tabla principal

alert_events

Tipos de alerta

- Jornadas abiertas
- Pausas excesivas
- Incidencias pendientes

Escaneo automático

Script:

tools/scan_alerts.php

Cron recomendado

Ejecución cada 15 minutos.

Ejemplo:

*/15 * * * * php tools/scan_alerts.php

10. Geolocalización

Objetivo

Registrar la ubicación de los fichajes.

Datos almacenados

- latitude
- longitude
- accuracy

Consideraciones

La geolocalización depende:

- Del navegador
 - Del dispositivo
 - De los permisos del usuario
-

11. Auditoría

Tabla principal

audit_logs

Información registrada

- Login
- Logout
- Fichajes
- Pausas
- Incidencias
- Ausencias
- Usuarios
- Alertas

Objetivo

Garantizar trazabilidad completa.

12. Seguridad

APP_KEY

Utilizada para:

- Cifrado
- Protección de datos sensibles

Protección CSRF

Todas las operaciones críticas utilizan token CSRF.

Contraseñas

Almacenamiento mediante hash seguro.

Sesiones

Gestión segura mediante sesión autenticada.

13. Copias de seguridad

Recomendación

Realizar copias:

- Base principal
- Bases tenant

Frecuencia recomendada

Diaria.

Procedimiento

Realizar exportación SQL completa.

Verificar periódicamente la restauración.

14. Actualizaciones

Procedimiento recomendado

1. Copia de seguridad.
2. Subida de código.
3. Ejecución de migraciones.
4. Verificación funcional.
5. Revisión de logs.

Verificaciones posteriores

- Login
 - Fichajes
 - Incidencias
 - Ausencias
 - Alertas
 - PUSH
-

15. Resolución de incidencias

No funcionan notificaciones PUSH

Verificar:

- Claves VAPID
- Service Worker
- Suscripción
- HTTPS

No funcionan alertas

Verificar:

- Cron
- AlertService
- alert_events

No funciona GPS

Verificar:

- Permisos navegador
- HTTPS
- Configuración dispositivo

Problemas migraciones

Verificar:

- schema_migrations
 - permisos SQL
 - archivos SQL
-

16. Checklist de puesta en producción

Antes de entregar una empresa:

- ✓ HTTPS activo
- ✓ Login operativo
- ✓ Base tenant creada
- ✓ Migraciones ejecutadas
- ✓ PUSH operativo

- ✓ GPS operativo
 - ✓ Alertas funcionando
 - ✓ Cron configurado
 - ✓ Auditoría validada
 - ✓ Copias de seguridad verificadas
-

17. Roadmap Kronetic

Evoluciones previstas

V2

- Comunicados internos
- Centro de mensajes empresa
- Dashboard ejecutivo
- KPI productividad
- API externa
- Integración ERP
- Notificaciones avanzadas

Futuras versiones

- Firma digital
 - Workflow documental
 - Integración biométrica
 - Inteligencia artificial aplicada a productividad
-

Conclusión

Kronetic V1 proporciona una plataforma robusta, escalable y preparada para entornos multiempresa, combinando control horario avanzado, auditoría, notificaciones, incidencias y trazabilidad completa mediante una arquitectura moderna basada en PHP 8.2 y MariaDB.