Perfecto, Chema. He analizado los dos árboles de estructura de proyecto (Arbol\_GPT.docx y Arbol\_Per.docx) y aquí tienes **una propuesta unificada, optimizada y profesional** que:

* Aúna lo mejor de ambas versiones.
* Sigue principios de arquitectura limpia, modular y escalable.
* Es compatible con despliegue en contenedores (Docker, Kubernetes).
* Está pensada para DevSecOps, observabilidad, documentación, automatización, y colaboración entre equipos.

**🌲 Árbol de Proyecto Unificado: PlatSentinel**

platsentinel/

│

├── README.md # Visión general del proyecto y guía rápida

├── .env.ejemplo # Variables de entorno ejemplo (Docker, API, DB, claves)

├── .gitignore

├── .dockerignore

│

├── .github/ # Workflows GitHub Actions

│ └── workflows/

│ ├── ci-cd.yml # Build, test, firma, push → registry

│ └── escaneo-seguridad.yml # Trivy, Semgrep, dependabot

│

├── infraestructura/ # Infraestructura como código (IaC)

│ ├── terraform/ # Configs para AWS, GCP o Azure

│ │ ├── variables.tf

│ │ ├── main.tf

│ │ └── outputs.tf

│ ├── kubernetes/ # Helm, manifiestos K8s, OPA, monitoreo

│ │ ├── k3s-valores-helm/

│ │ ├── manifests-basicos/

│ │ └── monitoreo/ # Prometheus, Grafana, Loki

│ └── scripts/

│ ├── generar-certificados.sh

│ ├── pruebas-chaosmesh.sh

│ └── backup\_restore.sh

│

├── api/ # Backend FastAPI

│ ├── app/

│ │ ├── \_\_init\_\_.py

│ │ ├── configuracion.py # Carga variables .env

│ │ ├── dependencias.py

│ │ ├── modelos.py

│ │ ├── esquemas.py

│ │ ├── rutas/

│ │ │ ├── autenticacion.py

│ │ │ ├── clientes.py

│ │ │ ├── tokens.py

│ │ │ └── resultados.py

│ │ └── servicios/

│ │ ├── orquestador.py

│ │ ├── integraciones.py

│ │ └── reportes.py

│ ├── tests/

│ │ ├── unitarios/

│ │ └── integracion/

│ ├── archivo-uvicorn.ini

│ └── requirements.txt

│

├── tokens/ # Tokens dockerizados (módulos independientes)

│ ├── base/

│ │ ├── Dockerfile

│ │ ├── manifest.yml

│ │ └── README.md

│ ├── toquen\_antesala/

│ │ ├── Dockerfile

│ │ ├── escaneo.py

│ │ └── utilidades.py

│ ├── toquen\_web/

│ │ ├── Dockerfile

│ │ └── escaneo\_web.py

│ ├── toquen\_cloud/

│ │ ├── Dockerfile

│ │ └── escaneo\_cloud.py

│ ├── virustotal/

│ │ ├── Dockerfile

│ │ └── ejecutar.sh

│ ├── firewall-aislar/

│ │ ├── Dockerfile

│ │ └── ejecutar.sh

│ └── partners/

│ └── ejemplo-token-tercero/

│ ├── Dockerfile

│ └── manifest.yml

│

├── integracion/ # Integraciones externas y APIs

│ ├── api\_whatsapp.py

│ ├── mapeo\_servicios.py

│ └── api\_telegram.py (opcional)

│

├── ui/ # Frontend Vue 3 + Tailwind

│ ├── src/

│ │ ├── main.js

│ │ ├── router/

│ │ │ └── index.js

│ │ ├── componentes/

│ │ ├── vistas/

│ │ └── utilidades/

│ ├── public/

│ │ └── index.html

│ └── tailwind.config.js

│

├── informes/ # Generación de informes y feedback analistas

│ ├── generador\_informes.py

│ └── integracion\_consultores.py

│

├── bot-whatsapp/ # WhatsApp/Twilio Bot

│ ├── main.py

│ ├── utils.py

│ └── plantillas-mensajes/

│ ├── alerta\_critica.txt

│ └── ayuda.txt

│

├── pruebas/ # Testing: funcional, integración, carga

│ ├── test\_api.py

│ ├── test\_tokens.py

│ ├── test\_dashboard.py

│ ├── postman/

│ ├── selenium/

│ └── carga-k6/

│

├── documentacion/ # Manuales, guías, decisiones técnicas

│ ├── manual\_usuario.md

│ ├── documentacion\_tecnica.md

│ ├── arquitectura\_modular.md

│ ├── guia\_despliegue.md

│ ├── registro\_cambios.md

│ └── AADR/

│ └── adr-0001-unificacion-ui.md

│

├── licencias/

│ └── LICENSE-3RD-PARTY.txt

│

└── variables\_globales.py # Centraliza claves, rutas y config común

**✅ Variables esenciales a incluir en .env.ejemplo y variables\_globales.py**

# Base de datos

BD\_HOST = "postgres"

BD\_PUERTO = 5432

BD\_NOMBRE = "sentinel"

BD\_USUARIO = "sentinel\_usr"

BD\_CLAVE = "super\_secreto"

# RabbitMQ / colas

RABBIT\_URL = "amqp://cola:5672"

# Claves API externas

CLAVE\_VIRUSTOTAL = "VT\_XXXXXXXX"

CLAVE\_TWILIO\_SID = "ACXXXXXXXXXXXXXXXX"

CLAVE\_TWILIO\_TOKEN = "XXXXXXXXXXXXXXXX"

NUMERO\_WHATSAPP\_REMISOR = "whatsapp:+1415XXXXXXX"

# Seguridad JWT

SECRETO\_JWT = "clave\_jwt\_ultra\_secreta"

ALGORITMO\_JWT = "HS256"

EXPIRACION\_JWT\_MINUTOS = 60

# MinIO o S3

S3\_ENDPOINT = "http://minio:9000"

S3\_BUCKET = "sentinel-artefactos"

S3\_USUARIO = "minioadmin"

S3\_CLAVE = "minioadmin"

# Entorno

ENTORNO = "desarrollo" # o "produccion"

**📌 Notas clave para implementación**

* **Docker Compose + Helm** preparados para entornos locales y K8s.
* **Tokens** pueden ampliarse sin tocar lógica principal.
* **Documentación técnica viva** (guía + AADR).
* **Multicanal de comunicación (WhatsApp/Telegram)**.
* **Panel Vue.js desacoplado y consumiendo solo APIs.**
* **Listo para integrar observabilidad, tracing y políticas OPA.**

¿Quieres que te genere esta estructura en carpetas reales (como .zip) o en un .docx para presentarla como anexo técnico del proyecto? También puedo construir automáticamente una plantilla base con Dockerfiles, README y .env. ¿Qué prefieres?