**🧠 Componentes Clave de la Aplicación AML + Auditoría Interna Inteligente**

**1. 🔍 Módulo de Detección de Blanqueo (AML clásico)**

* Análisis de transacciones con IA (Isolation Forest, reglas de umbral, análisis de redes).
* Panel con KPIs:
  + Volumen inusual de operaciones.
  + Entidades de alto riesgo.
  + Países no cooperantes (según FATF/GAFI).
  + Frecuencia de alertas generadas por cliente.
* Visualización con grafos (relaciones entre cuentas, origen/destino de fondos).

**2. 🧩 Módulo de KPI & Ataques a Negocio / Compliance**

* KPIs que vigilen integridad financiera:
  + Variación anómala de saldos.
  + Modificaciones en archivos de clientes.
  + Accesos no autorizados a bases de datos o archivos críticos.
* Análisis forense de impacto: ¿qué KPIs se vieron afectados tras un incidente?

**3. 🧑‍💼 Auditoría Interna**

* Detección de:
  + Cambios en configuraciones clave o archivos sensibles.
  + Escaladas de privilegios internas.
  + Acciones atípicas del personal con acceso privilegiado.
* Logs firmados + alertas automatizadas (Sysmon, AuditD, Wazuh o AIDE si es Linux).

**4. 📁 Repositorio de Excepciones / Casos Especiales**

* Base de datos editable con:
  + Casos reales de auditoría que no encajan claramente en la normativa.
  + Preguntas frecuentes del equipo de cumplimiento.
  + Resoluciones propuestas por IA basadas en NLP (GPT-like).
* Sistema de búsqueda semántica para encontrar precedentes similares.

**5. 🧾 Informes Ejecutivos HTML/PDF**

* Resumen ejecutivo con gráficas y visualizaciones de KPIs.
* Identificación de áreas críticas o atacadas.
* Recomendaciones y estado del cumplimiento.
* Exportación profesional para auditores externos.

**🧰 Tecnologías Recomendadas**

| **Componente** | **Tecnología** |
| --- | --- |
| Backend AML | Python + FastAPI |
| IA | scikit-learn / PyCaret / LLM (IA generativa para excepciones) |
| Visualización | Streamlit / Dash + Plotly + NetworkX |
| Auditoría | Wazuh / AuditD / File Integrity Monitor |
| Repositorio excepciones | MongoDB + NLP (con embeddings y recuperación semántica) |
| Informes | Jinja2 + wkhtmltopdf |
| Alertas | Telegram Bot API / Email (SMTP) |
| Seguridad | TLS, JWT, AES256, control de acceso granular |

## ✅ RESUMEN DE LO HECHO HASTA AHORA

### 🧠 Objetivo del proyecto:

Desarrollar una aplicación profesional de **detección de blanqueo de capitales (AML)** con IA y ciberseguridad, bajo el sello de Menarguez-IA Solutions.

### 📦 FASE 1 – Backend básico con FastAPI

* Creado el archivo main.py con endpoints para:
  + Registrar transacciones.
  + Generar alertas automáticas si el monto es sospechoso.
  + Consultar KPIs del sistema.
  + Añadir y consultar excepciones regulatorias.

### 🛠️ FASE 2 – Entorno virtual en Linux (Kali)

* Has creado un entorno con:

bash

CopiarEditar

python3 -m venv venv

source venv/bin/activate

pip install fastapi uvicorn

* Has lanzado el servidor en http://127.0.0.1:8010/docs.

### 🧾 FASE 3 – Script automático

* Se generó el archivo lanzar\_aml.sh para automatizar el arranque del entorno y el servidor:

bash

CopiarEditar

chmod +x lanzar\_aml.sh

./lanzar\_aml.sh

### 🗃️ FASE 4 – Persistencia real con SQLite

* Ahora tu API **guarda los datos en una base SQLite (aml\_data.db)**.
* Se añadió SQLAlchemy al backend:
  + Transacciones, alertas y excepciones quedan guardadas.
  + Al cerrar o reiniciar el servidor no se pierde nada.
  + Puedes analizar la base con herramientas como DB Browser for SQLite.