Reporte de Auditoría: Framework UNCASE (SCSF)

Fecha: 28 de Febrero de 2026
Auditor: Senior ML Engineer & Business Strategist
Estado del Sistema: Producción / Desarrollo Avanzado

Resumen Ejecutivo

Este documento presenta una auditoría crítica, estricta y de alto estándar del pipeline de la plataforma UNCASE. Se evalúan 14 dimensiones clave, incluyendo privacidad, matemáticas, estabilidad y visión de negocio, con el fin de identificar cuellos de botella técnicos y oportunidades estratégicas.

---

1. Privacidad (Privacy)

Estado: Robusto pero con dependencias críticas.

- Hallazgos: La implementación de `PIIScanner` es sólida, combinando heurísticas rápidas (Regex) con modelos NER avanzados (Presidio). El `PrivacyInterceptor` actúa como un firewall efectivo para tráfico de LLMs.
- Crítica: La implementación de Privacidad Diferencial (DP-SGD) en `LoraPipeline` es experimental y depende de parches manuales sobre el optimizador de `SFTTrainer`. Esto es frágil ante actualizaciones de librerías base.
- Riesgo: La anonimización depende de listas blancas (`bypass_words`), lo que introduce riesgo de filtración en dominios no previstos.

2. Cumplimiento (Compliance)

Estado: Alto.

- Hallazgos: El modelo `AuditLogModel` es exhaustivo: rastrea actor, acción, recurso, IP y contexto. Es fundamental para estándares GDPR/CCPA.
- Trazabilidad: La cadena `Seed -> Conversation -> QualityReport` asegura que cada dato sintético tiene un origen auditable.
- Crítica: Falta un mecanismo de "Derecho al Olvido" (borrado en cascada) automatizado para datos sensibles del usuario.

3. Estabilidad y Mantenibilidad (Stability & Maintainability)

Estado: Muy Alta (Código), Media (Dependencias).

- Hallazgos: Código limpio con `typing` estricto, `dataclasses` y manejo de excepciones granular. El uso de `structlog` facilita la observabilidad.
- Crítica: El parser de respuestas (`_parse_llm_response`) utiliza Regex como fallback. Esto es propenso a errores ante la naturaleza estocástica de los LLMs. La dependencia de proveedores externos vía `LiteLLM` introduce riesgos de disponibilidad ajenos al sistema.

4. Riesgos (Risks)

Estado: Riesgo Medio-Alto en Validez Semántica.

- Hallazgos:

* Alucinación no detectada: Las métricas son mayoritariamente sintácticas. Un diálogo puede ser aprobado si contiene palabras clave, aunque su significado sea incorrecto.

* Inyección de Prompts: La construcción de prompts inyecta texto de semillas directamente, lo que podría permitir ataques de *jailbreak* si las semillas no están curadas.

5. Validez de Métricas de Calidad (Validity of quality metrics)

Estado: Superficial / Heurístico.

- Fidelidad Factual: Se limita a búsqueda de palabras clave (`_context_presence`) y flujo de texto. No hay validación semántica real.
- Coherencia Dialógica: Basada en Similitud de Jaccard. Penaliza diálogos naturales con baja superposición léxica pero alta coherencia de significado.
- ROUGE-L: Inadecuado para evaluar la dinámica de un diálogo multi-turno; es una métrica de resumen, no de interacción.

6. Corrección Matemática (Correctness of Mathematics)

Estado: Correcta pero Simplista.

- Hallazgos: Los cálculos de promedios ponderados y métricas de diversidad son aritméticamente exactos.
- Crítica: El uso de `min()` en `compute_composite_score` es extremadamente severo. Una sola métrica heurística ruidosa que falle descarta todo el trabajo, aumentando el costo operativo por desperdicio de tokens.

7. Calidad del Proceso (Quality of the Process)

Estado: Excelente Diseño Arquitectónico.

- Hallazgos: El flujo circular `Seed -> Gen -> Eval -> Feedback -> Train` es de vanguardia.
- Punto Fuerte: La funcionalidad `generate_with_feedback` es un diferenciador competitivo clave que permite la mejora iterativa sin intervención humana constante.

8. Auditabilidad y Trazabilidad (Auditability and Traceability)

Estado: Excelente.

- Hallazgos: Cada artefacto (conversación, modelo, reporte) mantiene metadatos completos del experimento (modelo usado, temperatura, escenario). Los registros de auditoría son inmutables y detallados.

9. Áreas de Mejora (Areas of improvement)

- Validación Semántica: Integrar modelos "Juez" (LLM-as-a-Judge) o embeddings de similitud coseno.
- Robustez de Entrenamiento: Formalizar la integración de DP-SGD mediante un bucle de entrenamiento propio.
- Seguridad: Sanitizar entradas de semillas para prevenir ataques de inyección en la generación.

10. Soluciones (Solutions)

- Paralelización: Migrar los bucles `for` de generación y evaluación a `asyncio.gather` para reducir tiempos de espera.
- Refactorización DP-SGD: Utilizar integraciones nativas o custom loops que no dependan de parches internos de librerías de terceros.

11. Mejores Prácticas de Código (Code best practices)

Estado: Sobresaliente.

- Uso consistente de Pydantic para esquemas.
- Arquitectura modular que facilita el testing unitario.
- Documentación interna clara y tipado exhaustivo.

12. Idoneidad Open Source (Open Source Suitability)

Estado: Alta.

- Estructura modular con extras opcionales (`[ml]`, `[privacy]`) que facilita la instalación.
- Cumple con estándares de calidad que invitan a la colaboración de la comunidad.

13. Escalabilidad (Scalability)

Estado: Deficiente (Cuello de Botella).

- Crítica: El procesamiento es secuencial. Generar datasets grandes (10,000+ ejemplos) es inviable con el orquestador actual debido a la latencia acumulada de las llamadas a API.

14. Extensibilidad (Extensibility)

Estado: Muy Buena.

- El diseño basado en interfaces (`BaseMetric`, `BaseGenerator`) permite añadir soporte para nuevos modelos o métricas con mínimo esfuerzo.

Reinos Adicionales

15. Seguridad del Modelo (Model Security & Safety)

- Hallazgo: Falta detección de toxicidad y sesgos en el dataset sintético.
- Recomendación: Integrar filtros de seguridad post-generación antes de la fase de entrenamiento (Layer 4).

16. Eficiencia de Costos (Cost Efficiency)

- Hallazgo: El bucle de feedback es costoso en tokens.
- Recomendación: Utilizar modelos pequeños (SLMs) para el filtrado inicial y reservar los modelos grandes para la generación final y el "juicio" de calidad.

17. Experiencia de Desarrollo (DX)

- Hallazgo: Configuración compleja vía archivos de entorno y código manual.
- Recomendación: Desarrollar una CLI interactiva o interfaz visual para la gestión de semillas y monitoreo del pipeline en tiempo real.

Recomendaciones Estratégicas (Business & Strategy)

1. Diferenciación de Producto: Evolucionar de "herramienta de scripting" a "plataforma de Data Ops". El valor real está en la garantía de calidad de los datos sintéticos, no solo en su generación.
2. Monetización de Privacidad: El módulo de privacidad es lo suficientemente robusto para ser un producto independiente o un middleware empresarial.
3. Optimización Operativa: La paralelización de la generación no es solo una mejora técnica, es una necesidad comercial para reducir el *Time-to-Market* de los modelos entrenados.

Firma:

Senior ML Engineer & Data Scientist / Business Strategist Agent