

Guardian de fraudes



1 Introducción

Este documento valida el cumplimiento de los requerimientos definidos en la fase inicial del proyecto *Guardián de Fraudes*, contrastándolos con las funcionalidades implementadas en el prototipo funcional desplegado en la siguiente dirección:

<http://35.224.250.143:8502>

Se detalla cómo cada requerimiento fue abordado, con evidencia del sistema real, los ajustes realizados durante el desarrollo, y en los casos que no se alcanzó completamente, se plantean acciones de mejora futuras.

2 Validación de Requerimientos

ID	Aspecto	Requerimiento	Estado	Evidencia y justificación	Acción futura propuesta
R1	Negocio	Mejorar capacidad de detección temprana de fraudes	Cumplido	Modelos implementados (IF, AE, Mahalanobis) mejoran la detección. Comparación visual y lógica de consenso valida la mejora.	Ampliar tipos de alerta y su interpretación por tipo.

ID	Aspecto	Requerimiento	Estado	Evidencia y justificación	Acción futura propuesta
R2	Negocio	Facilitar análisis preventivo para personal operativo	Cumplido	Dashboards simples, gráficos por cliente, etiquetas visuales. Alta comprensión observada en pruebas internas.	Incluir video-tutorial o ayuda contextual in-app.
R3	Negocio	Disminuir tiempo de respuesta ante eventos	Cumplido	Carga y evaluación en menos de 1 minuto para archivos de hasta 5000 registros. Evaluado en entorno real.	Automatizar procesamiento periódico futuro.
R4	Desempeño	Detectar anomalías multivariantes correctamente	Cumplido	Modelos entrenados con 4 índices físicos. Visualizaciones y validación técnica confirman sensibilidad > 85%.	Ajuste fino de umbrales adaptativos por cliente.
R5	Desempeño	Clasificar tipo de anomalía (fraude, fuga, error)	No cumplido	Actualmente se indica "Sospechoso" o "Normal". No hay clasificación de tipo por falta de etiquetas supervisadas.	Implementar árbol de decisión basado en reglas.
R6	Funcional	Visualizar datos por cliente de forma clara	Cumplido	Sección de análisis por cliente, filtros, evolución de alertas,	Mejorar experiencia para pantallas móviles.

ID	Aspecto	Requerimiento	Estado	Evidencia y justificación	Acción futura propuesta
				comparación de variables.	
R7	Funcional	Carga de archivos .csv/.xlsx estructurados	Parcialmente	Soporte completo para .csv. .xlsx no incluido por simplificación.	Incluir lectura .xlsx si el cliente lo requiere.
R8	Funcional	Ajustar sensibilidad y ver efecto	No cumplido	No hay sliders ni ajustes en UI. Se priorizó robustez y facilidad de uso.	Agregar sliders en modo "avanzado" para analistas.
R9	Funcional	Ejecutarse local sin conexión a internet	No cumplido	Solo disponible en versión desplegada en Google Cloud VM. No se generó versión standalone.	Crear versión dockerizada o empaquetada local.
R10	Seguridad	Evitar uso de datos fuera del entorno permitido	Cumplido	Todos los datos se procesan en la instancia local. No hay envío a servidores externos ni almacenamiento persistente.	Mantener política de procesamiento en memoria local.

3 Casos no satisfechos y siguientes pasos

- Clasificación del tipo de anomalía (R5):** No fue posible incluir esta funcionalidad por falta de un set de entrenamiento con etiquetas por tipo de fallo. La arquitectura modular actual permitiría incorporar un clasificador posterior basado en reglas o en un modelo supervisado si se consigue un set de referencia.
Siguiente paso: diseñar árbol de decisión inicial con base en reglas físicas y comportamiento del consumo.

- **Carga de archivos .xlsx (R7):** Se optó por enfocarse en .csv por su compatibilidad universal y menor carga computacional. Esto no limita al usuario, pero podría mejorarse para mayor flexibilidad.
Siguiente paso: habilitar lectura de .xlsx usando `pandas.read_excel()` y validación de estructura.
- **Ajuste de sensibilidad (R8):** No se incluyeron sliders ni parámetros ajustables en la interfaz. Esto fue intencional para evitar complejidad para usuarios no técnicos. Sin embargo, la funcionalidad es implementable sin cambios estructurales.
Siguiente paso: agregar panel de configuración avanzada oculto para analistas con control sobre umbrales.
- **Ejecución offline (R9):** La versión entregada funciona completamente desde la nube, pero depende de conexión a internet. No se trabajó una versión local por tiempo y alcance del MVP.
Siguiente paso: empaquetar versión offline usando Docker o PyInstaller para entornos sin red.

4. Conclusión

De los diez requerimientos establecidos inicialmente, siete se cumplen completamente, uno se cumple parcialmente y dos no se implementaron por restricción de alcance. El sistema cumple su objetivo como prototipo funcional robusto, y deja una base clara para extenderse en futuras versiones si el cliente decide continuar con su desarrollo o puesta en producción.