INFORME DE COSTES Y CALIDAD EN EL PROCESO DE ALMACENAMIENTO Y PROCESADO DE DATOS

Clemente Rodriguez
Ibai Garcia
Aritz Basarrate

1. Introducción

En este informe se presentan los costes asociados al almacenamiento y procesado de datos, así como una evaluación de la calidad del proceso. Se analizan los principales factores que influyen en la eficiencia del sistema y se proponen mejoras para optimizar los recursos.

2. Objetivos

- Evaluar los costes operativos y de infraestructura.
- Analizar la calidad del almacenamiento y procesado de datos.
- Identificar áreas de mejora y posibles optimizaciones.

3. Introducción

3.1. Costes de almacenamiento

3.1.1. Infraestructura

Descripción: Máquina virtual (VM) que aloja Docker; dentro, un contenedor MySQL para las tablas de stocks y cryptos; un volumen persistente montado en el host VM para datos; opcionalmente un snapshot/backups a almacenamiento en bloque o objeto externo (por ejemplo, servicio cloud S3/Block Storage).

Justificación: La VM aporta aislamiento y control; Docker facilita despliegue reproducible; el volumen persistente evita pérdidas ante recreación de contenedores; backups externos son imprescindibles para recuperación ante fallo total del host.

3.1.2. Consumo energético

Componentes: consumo de la VM (parte proporcional del servidor físico), dispositivos de almacenamiento y refrigeración en el centro de datos.

Estimación y justificación: para una VM de tamaño moderado (1-2 vCPU, 4-8 GB RAM) el consumo incremental es pequeño, pero se debe contabilizar como fracción del coste del rack/CPD. Incluir coste energético evita subestimar operaciones continuas 24/7, especialmente si la VM se mantiene encendida para ingestión en tiempo real o jobs nocturnos.

3.1.3. Licencias y software

Elementos: MySQL (versión comunitaria gratuita o versión empresarial con soporte), sistema operativo de la VM (Linux libre o pago), herramientas de backup/monitoring y soluciones de copia externa.

Justificación: elegir MySQL Community reduce costes iniciales; sin embargo, si se requieren características empresariales (replicación avanzada, soporte), hay que prever licencias y SLAs que incrementan costes. Herramientas de backup y monitorización con soporte comercial aportan garantía de recuperación y operación estable.

3.1.4. Mantenimiento

Gastos: soporte técnico (horas de administración), actualizaciones de software, parches de seguridad, pruebas de restore y pruebas de integridad de datos.

Justificación: el coste humano de DBA/DevOps es significativo. Planificar horas mensuales (por ejemplo 4–8 h/mes para un setup pequeño) asegura disponibilidad y cumplimiento normativo; las pruebas regulares reducen el riesgo de pérdida de datos y penalizaciones regulatorias.

3.2. Costes de procesado

3.2.1. Capacidad de cómputo

Uso previsto: preprocesado ETL (limpieza, normalización), cálculo de indicadores financieros, entrenamiento de modelos de ML ligeros y consultas analíticas. Infraestructura: CPU para ETL y consultas; GPU opcional para entrenamiento si los modelos superan capacidad CPU.

Justificación: las cargas ETL y consultas pueden correr en la misma VM si son puntuales; para entrenamientos repetidos o modelos más complejos, añadir instancias con más CPU/RAM o acceso a GPU reduce tiempos y mejora rendimiento, justificando el coste adicional.

3.2.2. Tiempo de procesamiento

Impacto: tiempo de ejecución de jobs ETL y de entrenamiento afecta a la latencia de entrega de resultados y a coste por hora de la infraestructura.

Justificación: optimizar pipelines (vectorización, indexado en MySQL, particionado de tablas) reduce tiempo de procesamiento y por tanto coste. Medir tiempos promedio y picos permite dimensionar infraestructuras evitando sobreaprovisionamiento.

3.2.3. Costes de escalabilidad

Gastos adicionales: replicación de VM o contenedores, balanceo, almacenamiento adicional, licencias extra y mayor consumo energético.

Justificación: la escalabilidad horizontal es necesaria si aumenta la frecuencia de ingestión o crecen volúmenes de datos; planificar costes evita interrupciones en picos de mercado. Opciones “on demand” en cloud permiten pagar sólo cuando se escala, pero pueden salir más caras si el uso es constante.

4. Evaluación de calidad

4.1. Calidad del almacenamiento

4.1.1. Disponibilidad

Métrica: objetivo SLA expresado en porcentaje de tiempo (por ejemplo 99.9% mensual).

Justificación: para análisis financiero intradiario puede requerirse alta disponibilidad; para proyectos batch nocturnos, SLA más relajado es suficiente y más barato. La disponibilidad se mejora con réplicas y con almacenamiento redundante.

4.1.2. Redundancia

Métodos: snapshots regulares del volumen; replicación MySQL maestro-esclavo o cluster; backups fuera del host (almacenamiento en objeto); pruebas de restore.

Justificación: múltiples capas de respaldo reducen riesgo de pérdida por fallos de disco, corrupción o error humano; las pruebas de restore validan la integridad de los backups.

4.1.3. Seguridad

Medidas: cifrado en reposo del volumen; cifrado TLS para conexiones a MySQL; control de accesos (roles, usuarios con privilegios mínimos); firewall a nivel de VM; rotación de credenciales; logging y auditoría.

Justificación: datos financieros son sensibles y pueden tener implicaciones legales; medidas de seguridad minimizan riesgos de fugas y cumplen normativas y buenas prácticas.

4.2. Calidad del procesado

4.2.1. Precisión

Métrica: validación de outputs con datos de referencia, tests unitarios en pipelines y métricas de ML (RMSE, MAE, Sharpe ratio si aplica); controles de calidad en transformaciones (checksums, validación de esquema).

Justificación: asegurar que los cálculos (p. ej., retornos, indicadores) son correctos evita decisiones erróneas y pérdida de valor del proyecto; pruebas automatizadas detectan regresiones al cambiar código o datos.

4.2.2. Velocidad

Métrica: tiempo medio de ETL, latencia de consultas críticas y tiempo de entrenamiento por epoch.

Justificación: tiempos cortos permiten respuestas casi en tiempo real en análisis; identificar cuellos de botella (I/O, CPU, índices) permite optimizar y reducir costes operativos.

4.2.3. Fiabilidad

Métrica: tasa de fallos de jobs, reproducibilidad de resultados y consistencia temporal entre ejecuciones.

Medidas: versionado de esquemas y datos, logging estructurado, retry policies y alertas.

Justificación: la fiabilidad asegura que procesos automatizados no requieren intervención humana constante; reduce tiempo de resolución de incidentes y costes asociados.

5. Análisis y propuestas de mejora

5.1. Objetivos de mejora

Fortalecer seguridad y gobernanza de datos: proteger datos sensibles y cumplir normativas.

Medir y optimizar costes operativos: trazar coste por job y por MB almacenado para decisiones informadas.

5.2. Propuestas concretas de mejora (técnicas y operativas)

5.2.1. Alta disponibilidad y backups automatizados

Qué hacer: configurar replicación MySQL maestro-esclavo o activar clustering ligero; crear backups automáticos diarios incrementales a almacenamiento de objetos y snapshots semanales.

Beneficio: reducción del RTO/RPO; menor riesgo frente a fallo del host.

Impacto en coste: almacenamiento adicional y pequeñas horas de administración; coste justificado por la reducción de riesgo.

5.2.2. Volúmenes persistentes y separación de responsabilidades

Qué hacer: mover datos MySQL a volúmenes montados en la infraestructura de la VM con IOPS garantizados o usar un almacenamiento en bloque gestionado; separar servicios (DB, ETL, experimentación) en contenedores distintos.

Beneficio: mejor rendimiento I/O, menor impacto entre procesos; facilidad para escalar componentes de forma independiente.

Impacto en coste: mayor coste de almacenamiento con IOPS garantizadas; mejora significativa en latencias ETL.

5.2.3. Automatizar pipeline ETL reproducible

Qué hacer: crear pipelines con una herramienta ligera (Airflow, Prefect Core, o scripts containerizados con versionado) que: validen esquemas, apliquen transforms y escriban auditoría por lote.

Beneficio: reproducibilidad, detección temprana de errores en datos, menor intervención manual.

Impacto en coste: esfuerzo inicial de desarrollo; ahorro medio-largo plazo en tiempo de operación.

6. Conclusiones

Resumen ejecutivo: La arquitectura actual (CSV → MySQL en Docker sobre una VM) es apta para prototipos y experimentación, pero insuficiente para cargas crecientes o producción continua; requiere mejoras en resiliencia, automatización y monitorización para soportar IA aplicada a datos financieros.

Riesgos principales identificados: dependencia de un único host; backups y restores poco automatizados; posibles cuellos de botella I/O; ausencia de pipelines reproducibles; visibilidad limitada del coste operativo.

Beneficios esperados de las mejoras propuestas: replicación y backups automatizados reducen RTO/RPO y riesgo de pérdida de datos; separación de componentes y volúmenes con IOPS garantizados mejora latencias y fiabilidad; pipelines ETL reproducibles y monitorización permiten optimizar costes y detectar errores precozmente; uso puntual de recursos acelerados reduce drásticamente tiempos de entrenamiento sin mantener coste fijo.

Prioridades operativas: 1) implementar replicación y backups automáticos; 2) desplegar monitorización y paneles de coste/rendimiento; 3) optimizar MySQL (índices, particionado) y construir pipelines ETL versionados; 4) aplicar cifrado y controles de acceso; 5) habilitar escalado puntual con instancias GPU cuando sea necesario.

KPIs críticos a mantener: disponibilidad objetivo ≥ 99.9%; RTO ≤ 1 hora; RPO ≤ 24 horas; reducción del 30% en tiempo medio de ETL; p95 de consultas críticas < 500 ms; tasa de fallos de jobs < 1% mensual.

Trade-offs y justificación económica: invertir en almacenamiento con IOPS y en automatización eleva costes fijos modestos pero reduce costes operativos y riesgos mayores; usar GPUs sólo bajo demanda optimiza coste/beneficio para entrenamientos pesados.

Próximos pasos inmediatos: activar backups automatizados y replicación básica, desplegar métricas esenciales, y ejecutar un ciclo de pruebas de restore y de performance para validar supuestos de coste y SLA.

Conclusión final: Con ajustes priorizados en resiliencia, monitorización y automatización, la plataforma pasará de un prototipo vulnerable a una base sólida y escalable para proyectos IA financieros, manteniendo un equilibrio entre coste, rendimiento y seguridad.