El Libro Definitivo del SRE de Pagos Globales

De Operador de NOC a Staff Engineer en un Mundo Adquirente. Un manual completo para dominar la confiabilidad extrema en sistemas de pago global, desde ISO 8583 hasta arquitecturas active-active, observabilidad elite y respuesta a incidentes críticos.

Fundamentos del Ecosistema de Pagos

Como SRE en pagos, no solo operamos máquinas; operamos dinero en movimiento. Debemos entender el "quién es quién" en cada transacción. La confiabilidad no termina en nuestro data center; se extiende a través de toda la cadena de procesamiento.

Cliente

Titular de la tarjeta que inicia la transacción en el comercio.

Comercio

Establecimiento que acepta el pago mediante terminal POS o e-commerce.

Adquirente

Procesa pagos en nombre del comercio y conecta con la red de pago.

Red (Visa/MC)

Autopista global que enruta mensajes entre adquirente y emisor.

Emisor

Banco que emite la tarjeta y autoriza la transacción del cliente.

Acrónimos Críticos

Identificadores

- **PAN:** Número de cuenta principal (Primary Account Number)
- **BIN:** Número de identificación del banco (Bank Identification Number)
- **STAN:** Número de auditoría de traza del sistema (Systems Trace Audit Number)
- **RRN:** Número de referencia de recuperación (Retrieval Reference Number)

Autenticación

- **ARQC:** Código de autenticación de solicitud (Authorisation Request Cryptogram)
- **ATC:** Contador de transacciones de aplicación (Application Transaction Counter)
- **TVR:** Valor de verificación de terminal (Terminal Verification Results)

Procesamiento

- **DE:** Elemento de datos (Data Element) en ISO 8583
- **MTI:** Indicador de tipo de mensaje (Message Type Indicator)
- **STIP:** Procesamiento en modo suplente (Stand-In Processing)

Actores Adicionales del Ecosistema

Processor (Procesador)

Servicio que puede actuar como intermediario entre el adquirente y la red. Algunos adquirentes externalizan la conectividad a procesadores especializados que manejan la traducción de protocolos y la integración con múltiples redes.

Gateway (Pasarela)

Sistema que facilita pagos en e-commerce, agregando capas de seguridad como 3D Secure. El gateway recibe el pago del navegador, autentica al cliente y comunica con el adquirente usando APIs REST en lugar de ISO 8583.

Interfaces Técnicas del Ecosistema

ISO 8583

Protocolo binario para autorizaciones en tiempo real entre POS, adquirente, red y emisor.

Archivos Batch

Procesamiento por lotes para clearing y settlement al final del día.

APIs REST

Reporting, consultas y operaciones administrativas para comercios y emisores.

ISO 8583 Operativo Real

ISO 8583 es el latín de los pagos. No necesitas ser un cardadador de mainframe, pero debes poder leer un mensaje hexdump como si fuera un libro abierto durante un incidente. Un mensaje tiene tres partes principales: MTI (Message Type Indicator), Bitmap(s) y Data Elements.

Ejemplo de Mensaje ISO 8583 en Hexdump

```
0200 7020800000000000 16 4543123456789012 001000 000000001500 0626123045 123456
|    |                |  |                |      |            |         |
|    |                |  |                |      |            |         +-- DE 11: STAN
|    |                |  |                |      |            +-- DE 7: Fecha/Hora
|    |                |  |                |      +-- DE 4: Monto ($15.00)
|    |                |  |                +-- DE 2: PAN (454312...1234)
|    |                |  +-- DE 3: Código de procesamiento (001000 = compra)
|    |                +-- Bitmap primario (70 20 80 00...)
|    +-- MTI: 0200 (Solicitud de autorización)
+-- MTI: 0200 (Solicitud de autorización)
```

Códigos de Respuesta (DE39) Más Comunes

Código	Estado	Significado Operativo
00	Aprobado	Transacción autorizada exitosamente
05	Rechazado	No honrar - decisión del emisor
14	Rechazado	Formato de tarjeta inválido
41	Rechazado	Tarjeta perdida - bloqueo inmediato
43	Rechazado	Tarjeta robada - alerta de seguridad
51	Rechazado	Fondos insuficientes en cuenta
54	Rechazado	Tarjeta vencida - renovación necesaria
55	Aprobado	Pin correcto - autenticación exitosa
57	Rechazado	Transacción no permitida al titular
58	Rechazado	Transacción no permitida al terminal
61	Rechazado	Límite de monto excedido
62	Rechazado	Restricción especial - ver emisor
63	Rechazado	Seguridad violada - contacto manual
65	Rechazado	Límite de intentos excedido
68	Rechazado	Respuesta tardía - timeout de red
75	Rechazado	Límite de reintentos alcanzado
76	Rechazado	Reverso no encontrado - inconsistencia
77	Rechazado	Reverso inválido - datos corruptos
78	Rechazado	Ya reversado - duplicación detectada
79	Rechazado	Reverso de reverso - operación inválida
80	Rechazado	Fecha/hora inválida - sincronización
81	Rechazado	MAC inválido - firma corrupta
82	Rechazado	Registro de certificado inválido
83	Rechazado	Autenticación de datos fallida
84	Rechazado	Identificador de aplicación inválido
85	Aprobado	No hay acción - estado normal
86	Rechazado	MAC no verificado - seguridad comprometida
87	Rechazado	MAC no verificado - intento de fraude
88	Rechazado	MAC no verificado - clave corrupta
89	Rechazado	MAC no verificado - origen desconocido
90	Rechazado	Corte en curso - procesamiento temporal
91	Rechazado	Emisor no disponible - retry necesario
92	Rechazado	Ruta no encontrada - configuración
93	Rechazado	Viola las leyes - transacción ilegal
94	Rechazado	Duplicado - transacción repetida
95	Aprobado	Reembolso completado - reverso exitoso
96	Rechazado	Mal funcionamiento - error del sistema

Mini Taller: Captura y Decodificación

01

Capturar Tráfico

Usa Wireshark o tcpdump para capturar tráfico TCP en el puerto del switch de autorización (generalmente 5000-6000).

02

Identificar Mensajes

Filtra por el puerto y protocolo ISO 8583. Los mensajes comienzan con el MTI (4 bytes) seguido del bitmap.

03

Decodificar Hexdump

Usa herramientas online como ISO8583 Decoder o implementa un parser en Python con la librería iso8583.

04

Analizar Campos

Verifica el DE39 para el código de respuesta y el STAN para correlación con logs del sistema.

Arquitectura de Alta Disponibilidad

No hablamos de tener 3 réplicas de un microservicio; hablamos de arquitecturas geo-distribuidas con tiempos de conmutación de milisegundos. La disponibilidad de 99.99%+ es obligatoria para sistemas de pago.

Active-Active Real vs. Activo-Pasivo

Activo-Pasivo (Tradicional)

Un data center primario maneja todo el tráfico. El secundario está esperando. En un desastre, hay que hacer un cambio de DNS, mover bases de datos, etc. Esto puede tomar minutos. **InaceptABLE para pagos de alto valor.**

Active-Active Real (Nuestro Objetivo)

Ambos centros de datos reciben tráfico simultáneamente. Cada uno tiene su propio conjunto de aplicaciones y bases de datos. El desafío es el estado y la consistencia. **Objetivo: conmutación en milisegundos.**

Stand-In Processing (STIP)

Cuando la red no puede contactar al emisor, puede aprobar la transacción en "modo suplente" basándose en reglas y límites predefinidos. El adquirente también tiene su propio modo stand-in interno para fallos de red.

Timeout de Emisor

El emisor no responde en el tiempo esperado (generalmente 2-3 segundos).

Verificar Configuración STIP

¿Está habilitado STIP para este BIN? ¿El monto es menor que el límite pre-aprobado?

Aprobar con Riesgo

El adquirente asume el riesgo y aprueba la transacción. Se registra para reconciliación posterior.

Reconciliación

Cuando el emisor esté disponible, se valida la transacción y se ajusta el riesgo.

Chaos Engineering en Pagos

Probar la resiliencia inyectando fallos controlados. Herramientas como Gremlin o Chaos Mesh permiten simular desconexiones de emisores, retrasos en respuestas y saturación de conexiones.

Desconectar Emisor Simulado

Verificar que el sistema activa STIP automáticamente y mantiene el servicio.

Retrasar Respuestas

Simular latencia de red alta y verificar que los timeouts y reintentos funcionan correctamente.

Saturar Conexiones

Probar el comportamiento bajo carga extrema y verificar que el rate limiting protege el sistema.

MTBF y MTTR: Cálculo con Datos Reales

MTBF (Mean Time Between Failures)

Tiempo promedio entre fallos. Para un sistema Tier-1:

$$MTBF = \frac{\text{Tiempo total de operación}}{\text{Número de fallos}}$$

Ejemplo: 30 días × 24h = 720h operación. 3 fallos = **MTBF = 240h**

MTTR (Mean Time To Recover)

Tiempo promedio para recuperar el servicio:

$$MTTR = \frac{\text{Tiempo total de recuperación}}{\text{Número de fallos}}$$

Ejemplo: 15min + 30min + 45min = 90min. **MTTR = 30min**

Modelo Económico del Downtime

10K
Transacciones/Minuto
Volumen en hora pico

80%
Tasa de Aprobación
Transacciones exitosas

$50
Valor Promedio
Por transacción

1.5%
Comisión
Ingreso para adquirente

$6K
Costo/Minuto
Solo comisiones

$180K
30 Minutos
Costo total

Observabilidad Elite

Métricas, logs y trazas no son tres cosas separadas. Son la Santa Trinidad para entender un sistema distribuido. En pagos, la "experiencia del usuario" es una transacción exitosa y rápida.

SLIs (Service Level Indicators) Específicos de Pagos

Authorization Success Rate (ASR)

Porcentaje de autorizaciones aprobadas sobre solicitudes totales. Es nuestro SLI más importante. Una caída en ASR es una caída de ingresos.

Issuer/Network Latency (P99)

Tiempo que tarda la red + emisor en responder. Medido desde que enviamos el mensaje hasta que recibimos la respuesta.

Approval Ratio

Similar al ASR, pero específico por emisor o BIN. Si un banco empieza a rechazar todo, debemos aislarlo.

Reversal Ratio

Transacciones reversadas sobre aprobadas. Un ratio alto indica problemas de timeouts o bugs en POS.

Network Connectivity Health

Latencia y errores de socket hacia cada red (Visa, MC, Amex) y emisores directos.

Switch Processing Latency (P99)

Nuestro propio tiempo de procesamiento interno desde que recibimos el mensaje hasta que lo enviamos a la red.

Dashboards Visuales en Grafana

1

ASR Global

Gráfica de serie temporal mostrando ASR en la última hora con línea de objetivo (SLO) y línea de alerta (burn rate).

2

Latencia de Red

Heatmap mostrando latencia P50, P95, P99 por cada red (Visa, MC). Ayuda a ver degradación en los carriles.

3

Top Emisores

Tabla con top 5 emisores con menor ASR en los últimos 5 minutos. Permite ver si un banco específico está caído.

4

Volumen de Transacciones

Métrica de negocio mostrando TPS vs. volumen aprobado. La distancia entre ambas líneas es el dinero que no estamos ganando.

Alertas Buenas vs. Malas

❌ Mala Alerta

`CPU > 80%`

¿Y qué? El CPU puede estar alto por un proceso batch legítimo. No indica un problema funcional real.

✅ Buena Alerta

`rate(errors[1h]) / rate(requests[1h]) > 0.001 * 14.4`

Ratio de error en última hora consumiendo todo error budget mensual en menos de 2 horas (burn rate).

Mini Recetario de PromQL para Pagos

```
# ASR por Emisor
sum by(issuer) (rate(requests{status="approved"}[1h]))
/ sum by(issuer) (rate(requests[1h]))

# Latencia P99 por Red
histogram_quantile(0.99,
  sum by(le, network) (rate(latency_bucket[1h]))
)

# Tasa de Reversos por Comercio
sum by(merchant) (rate(reversals[1h]))
/ sum by(merchant) (rate(approvals[1h]))

# Burn Rate de Error Budget
sum by(service) (rate(errors{service="auth-switch"}[1h]))
/ sum by(service) (rate(requests{service="auth-switch"}[1h]))
> 0.001 * 14.4
```

Predictive Monitoring con Holt-Winters

Detectar anomalías estacionales usando modelos de suavizamiento exponencial. Por ejemplo, una caída de ASR en hora valle que normalmente no debería ocurrir puede ser un problema silencioso.

1

Modelo Estacional

Entrena con datos históricos de 30 días para capturar patrones diarios y semanales.

2

Umbral Dinámico

Calcula bandas de confianza (±2σ) alrededor de la predicción.

3

Alerta Anomalía

Si el valor real cae fuera de las bandas, dispara alerta de problema silencioso.

SLO Engineering y Estrategia

Los SLOs no son solo números. Son la herramienta de gestión que equilibra la innovación (lanzar features) con la confiabilidad. El error budget es el combustible que permite innovar sin comprometer la estabilidad.

Definición de SLOs por Criticidad

SLO Estricto

Core Authorizations

99.95% de disponibilidad en ventana mensual. Cualquier transacción de autorización en tiempo real.

SLO Medio

Batch y Reportes

99.5% de éxito en ventana de 24 horas. Deben correr antes del inicio del siguiente ciclo de liquidación.

SLO Bajo

Paneles de Administración

99% de disponibilidad. Pueden caer sin que un comercio deje de vender.

Error Budgets y Política de Burn Rate

El error budget es el 1 - SLO. Si nuestro SLO es 99.9%, tenemos un 0.1% de tiempo de error permitido al mes. Ese es nuestro presupuesto para gastar en incidentes.

1

Burn Rate < 1x

Consumo bajo

Monitoreo normal. Sin acciones especiales.

2

Burn Rate 1x-2x

>2% en 6h

Ticket de investigación. Observar tendencias.

3

Burn Rate 2x-6x

>5% en 3h

Alerta a equipo. Posible incidente P3/P4.

4

Burn Rate 6x-14.4x

>10% en 1h

Página al SRE de guardia (P1). Congelar lanzamientos.

5

Burn Rate > 14.4x

>5% en 10m

Página CRÍTICA. Se está quemando el budget muy rápido.

Matriz de Decisión de Error Budget

Consumo	Acción Técnica	Acción de Negocio
10%	Revisión de cambios recientes	Monitoreo aumentado
25%	Freeze de lanzamientos	Reunión con producto
50%	Reunión de emergencia	Comunicación a dirección
75%	Activar DR plan	Comunicación ejecutiva
100%	Investigación profunda	Revisión de SLA

SLA Real con Cliente Grande

Acuerdo de Nivel de Servicio

- Disponibilidad: 99.95% mensual
- Latencia P99: < 500ms
- Tiempo de recuperación: < 30min
- Notificación de incidente: < 15min

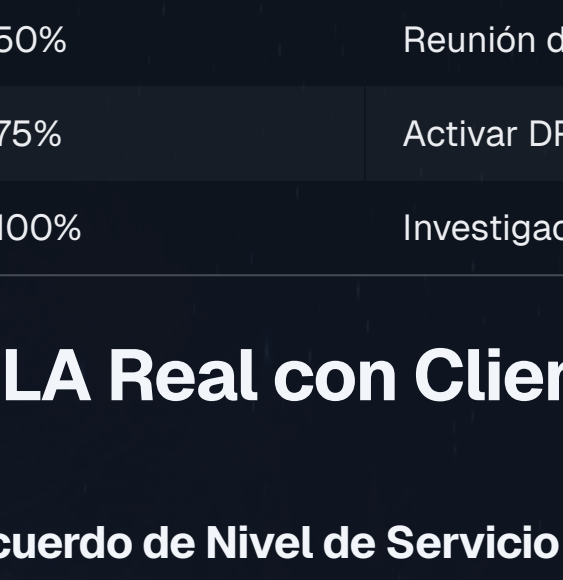
Traducción a SLOs Técnicos

- ASR > 99.95%
- Latencia de red P99 < 250ms
- MTTR < 30min
- Time to Acknowledge < 5min

Penalizaciones

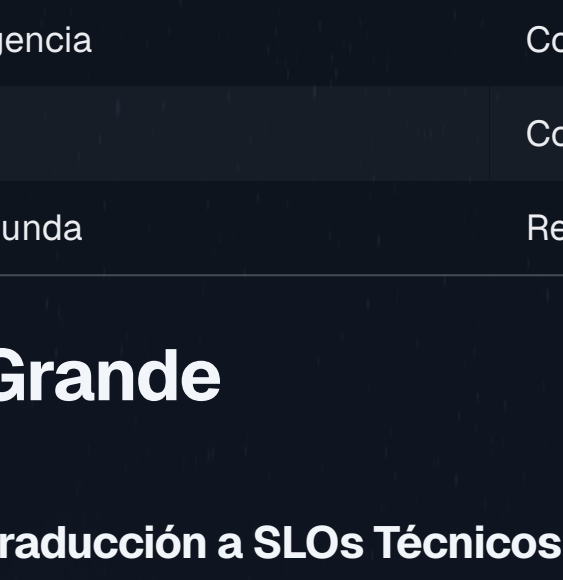
- 99.9% - 99.95%: 5% descuento
- 99.5% - 99.9%: 15% descuento
- < 99.5%: 30% descuento
- Incidente P0: 20% descuento

SLOs para Batch de Liquidación



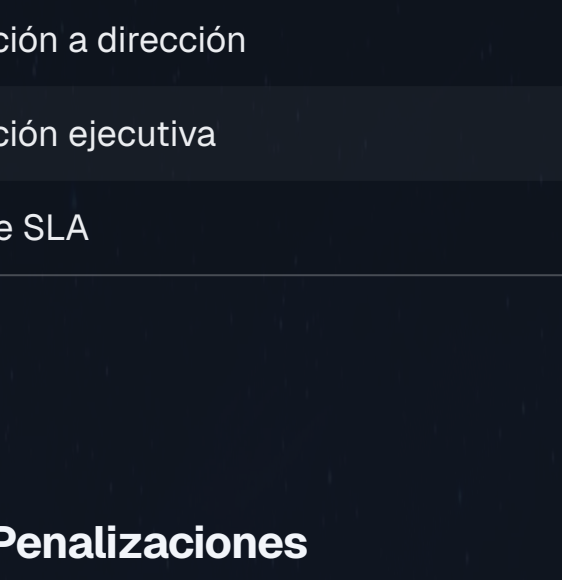
Tiempo de Procesamiento

Batch debe completarse en < 2h



Tasa de Éxito de Archivos

Archivos procesados sin errores



Integridad de Datos

Sumas y totales coinciden

Logging y Tracing Distribuido

Si no puedes correlacionar una transacción a través de 10 sistemas, no puedes arreglarla cuando falla. El formato JSON estandarizado y el tracing distribuido son obligatorios para la observabilidad moderna.

Formato JSON Obligatorio

Cada servicio debe loguear en JSON con campos estandarizados. El `trace_id` es para los ingenieros. El `transaction_id` (STAN) es para el negocio y el soporte. Ambos son esenciales.

```
{
  "timestamp": "2024-05-20T14:32:10.123Z",
  "level": "ERROR",
  "service": "authorization-switch",
  "trace_id": "abc-123-def-456-ghi",
  "span_id": "def-456",
  "transaction_id": "STAN-123456",
  "message": "Timeout esperando respuesta de Visa",
  "error_code": "TIMEOUT_NETWORK",
  "duration_ms": 2500,
  "metadata": {
    "bin": "454312",
    "network": "VISA",
    "mti": "0210"
  }
}
```

Tracing Distribuido con OpenTelemetry

Entrada HTTP/TCP

El POS nos contacta. Inicio de la traza.

Validación y Enriquecimiento

Parseo del mensaje y búsqueda del BIN.

Criptografía

Llamada al HSM para verificar PIN o generar código.

Llamada a la Red

Envío del mensaje a Visa (incluye tiempo de red y Visa+Emisor).

Persistencia

Guardar el resultado en la base de datos de transacciones.

Respuesta al POS

Envío de la respuesta al terminal.

Objetivos por Span

End-to-End

500ms

Span A a F (P99)

Llamada a Red

250ms

Span D (P99)

Procesamiento Interno

20ms

Span B (P99)

Propagación de Contexto W3C TraceContext

El contexto de traza se propaga a través de sistemas legacy que no lo soportan inyectando el `trace_id` en un campo reservado de ISO 8583 (por ejemplo, DE 48 o un campo privado).

1

Sistema Moderno

OpenTelemetry inyecta `traceparent` en cabecera HTTP.

2

Legacy ISO 8583

Extrae `trace_id` y lo inyecta en DE 48 (campo privado).

3

Sistema Legacy

Lee `trace_id` de DE 48 y lo propaga en logs.

Correlación de Logs entre Sistemas

Usando el mismo `transaction_id` (STAN), podemos correlacionar logs del switch, la base de datos y el HSM para reconstruir la historia completa de una transacción.

1 **Switch: Inicio**

2024-05-20T14:32:10.123Z - Recibida solicitud STAN-123456

2 **HSM: Firma**

2024-05-20T14:32:10.150Z - Solicitud de firma para STAN-123456

3 **HSM: Respuesta**

2024-05-20T14:32:10.180Z - Firma generada para STAN-123456

4 **Switch: Envío**

2024-05-20T14:32:10.200Z - Mensaje enviado a Visa STAN-123456

5 **BD: Persistencia**

2024-05-20T14:32:10.250Z - Transacción STAN-123456 guardada

6 **Switch: Fin**

2024-05-20T14:32:10.300Z - Respuesta enviada STAN-123456

Buenas Prácticas de Logging para PCI DSS

Enmascarar PAN

Siempre mostrar solo los 6 primeros y 4 últimos dígitos. Nunca loguear el PAN completo.

No Loguear PIN

El PIN nunca debe aparecer en logs. Si es necesario para debugging, usar hash irreversible.

Rotación de Logs

Rotar logs diariamente y cifrar en reposo. Retención según políticas de cumplimiento.

Acceso Restringido

Solo personal autorizado con necesidad de negocio puede acceder a logs con datos sensibles.

Incident Response Real

Cuando suena el pájaro (PagerDuty/Opsgenie), el pánico es el enemigo. El proceso es tu amigo. La severidad se basa en impacto económico, no en la gravedad técnica del error.

Severidades Basadas en Impacto Económico

P0 - Todo se está quemando

Caída total del servicio. Imposible vender. Múltiples comercios afectados. **> $50K/hora**. Respuesta inmediata. Guarida de guerra. IC asignado. Comunicación a CEO en < 15min.

P1 - Incendio en una habitación

Degradación severa. Un emisor importante caído. Un lote de liquidación fallido. **> $10K/hora**. Respuesta inmediata. Equipo completo. Comunicación a dirección.

P2 - Huele a humo

Error funcional en feature no crítico. Latencia elevada para subconjunto pequeño. **Bajo impacto**. Durante horas hábiles. Parche en próximo release.

P3 - Humo de cigarrillo

Error cosmetico en UI. Alerta ruidosa pero no indica problema real. **Sin impacto**. Backlog del equipo.

Runbook de Respuesta a Incidentes P0

01

Detección

Alerta de burn rate salta. Ingeniero de guardia recibe el pájaro.

02

Agradecimiento y Triage (0-5 min)

El on-call ACK la alerta. Se une al canal de incidentes. Evalúa impacto: ¿Es P0? ¿Está todo caído o solo una región? Si es P0, se declara.

03

Comunicación Inicial (5-10 min)

El Incident Commander (IC) se nombra. Publica mensaje: "Investigando incidente P0 que afecta autorizaciones. Update en 15min." Crea documento de telemetría.

04

Mitigación (En adelante)

El equipo de troubleshooting busca causa o mitigación más rápida. El IC asigna tareas. Prioridad: restaurar el servicio, no encontrar causa raíz.

05

Resolución

Se declara que el servicio ha vuelto a la normalidad. IC publica: "Incidente resuelto. Servicio nominal. Postmortem en 48h."

06

Postmortem (48h)

Reunión sin culpa. Análisis de línea de tiempo, causa raíz, y acciones con dueños y fechas para que no vuelva a pasar.

Organigrama de Roles en Incidente P0

Incident Commander (IC)

Coordina el incidente, asigna tareas, comunica estado. No debuggea. Deja de debuggear para coordinar.

Scribe

Lleva la línea de tiempo del incidente en el documento de telemetría. Registra decisiones y acciones.

Comms Lead

Comunica con stakeholders externos (negocio, clientes, ejecutivos). Mantiene actualizaciones claras.

Technical Lead

Lidera el equipo de troubleshooting. Identifica causa raíz. Propone soluciones técnicas de mitigación.

Plantillas de Comunicación

Slack - Update Inicial

> **Incidente P0 - Autorizaciones Degradadas**
>
> Estamos investigando un incidente P0 que afecta las autorizaciones de pago. ASR global cayó del 92% al 70%. Equipo completo trabajando. Próximo update en 15 minutos.
>
> 🔴 Estado: Degradado

Correo Ejecutivo - Update 1

> **Asunto: Incidente P0 - Impacto en Autorizaciones**
>
> Estimado equipo ejecutivo,
>
> Estamos gestionando un incidente P0 que afecta las autorizaciones de pago desde las 14:30 UTC. ASR global está en 70% (normal: 92%). Causa raiz en investigación.
>
> Equipo completo trabajando. Próximo update en 30 minutos.
>
> Saludos,
> SRE Team

Checklist: Cosas que NUNCA Debes Hacer

- **Reiniciar servicios sin consenso**

  Puede empeorar el problema o perder evidencia forense.

- **No comunicar**

  El silencio genera pánico. Update cada 15min aunque no haya novedades.

- **Hacer cambios sin rollback plan**

  Cada mitigación debe tener un plan B para revertir si empeora.

- **Asignar culpables durante el incidente**

  El postmortem sin culpa es después. Ahora, solo resolver.

- **Ignorar el runbook**

  Los runbooks existen porque alguien ya pasó por esto. Síguelos.

Incidentes Simulados

Aquí es donde se forjan los seniors. Vamos a caminar por el valle de la muerte cinco veces. Cada incidente incluye contexto, síntomas, debugging paso a paso y lecciones aprendidas.

Incidente 1: El Emisor Silencioso

1

Contexto

Jueves 3:00 PM, hora pico en Brasil. Procesamos transacciones para un gran banco emisor "Banco do Brasil".

2

Síntomas

ASR global cae del 92% al 70% en 2 minutos. Alerta de burn rate P1 salta.

3

Métricas

ASR para "Banco do Brasil" es 0%. Latencia hacia ese banco es 5 segundos y luego timeout.

4

Logs

Error: Connection timeout a socket del emisor después de 2500ms. IOException al leer respuesta.

5

Decisión Correcta

Activar modo Stand-In para ese emisor si tenemos límites pre-acordados. IC comunica: "Banco do Brasil no responde. Activamos STIP."

6

Aprendizaje

La dependencia de terceros es nuestro mayor riesgo. Necesitamos mecanismos automáticos de STIP y comunicación proactiva con la red.

Incidente 2: Tormenta de Reintentos Black Friday

1

Contexto

Cyber Monday, 12:00 PM. Tráfico 3 veces el normal.

2

Síntomas

TPS se dispara a 10 veces lo normal. Servicios dan errores 503. Base de datos muestra alto número de locks.

3

Métricas

Gráfico de TPS muestra pico anómalo en forma de diente de sierra.

4

Logs

Miles de entradas con mismo STAN pero diferentes timestamps. Un comercio reintentando.

5

Decisión Correcta

Implementar rate limiter a nivel de firewall para ese comercio. Llamar a soporte del comercio para arreglar configuración de POS.

6

Aprendizaje

Los clientes mal configurados pueden ser armas de destrucción masiva. Debemos tener rate limiting por cliente y fomentar buenas prácticas.

Incidente 3: Desbordamiento Contador STAN

1

Contexto

1 de enero, 00:01 AM. Comienza el nuevo año.

2

Síntomas

Todas las transacciones nuevas fallan con error genérico. Tasa de error 100%.

3

Métricas

No hay timeout, son errores inmediatos. Latencia interna baja.

4

Logs

ERROR: duplicate key value violates unique constraint "transactions_pkey". STAN se reinició a 0.

5

Decisión Correcta

Modificar lógica de generación del STAN para incluir año o prefijo (ej. 2400001 para 2024). Implementar hotfix con prefijo de año.

6

Aprendizaje

Los sistemas heredados tienen suposiciones que son trampas explosivas. El testing de fecha de fin de año es obligatorio.

Incidente 4: Error de Red que No Era de Red

1

Contexto

Migración de data center a la nube.

2

Síntomas

0.1% de transacciones fallan con "MAC Invalid". Error intermitente.

3

Métricas

Error de seguridad, no timeout. Checksums TCP/IP correctos.

4

Logs

ERROR: MAC verification failed for message with STAN X. Byte en campo no crítico cambió.

5

Causa Real

Nuevo firewall en nube tenía DPI para tráfico financiero, intentaba "normalizar" caracteres. Corrompió firma criptográfica.

6

Decisión Correcta

Deshabilitar DPI para puerto específico del tráfico de pagos. Tráfico debe ser tratado como binario opaco.

7

Aprendizaje

Nunca asumas que la red es inocente. El camino del mensaje está lleno de duendes que pueden "mejorar" el tráfico de formas que destruyen la criptografía.

Incidente 5: Liquidación que No Cuadra

1

Contexto

Lunes 8:00 AM. Equipo de finanzas reporta archivo de liquidación del domingo no cuadra. Diferencia de $1M.

2

Síntomas

No hay caída, no hay alertas técnicas. Es un problema de datos.

3

Debugging

Extraer suma total de transacciones aprobadas del domingo. Comparar con total del archivo. Buscar transacciones en BD pero no en archivo.

4

Causa Real

Bug en proceso batch de liquidación. Al encontrar error de formato en una transacción, abortó inclusión de todas las del lote pequeño.

5

Decisión

Ejecutar proceso de conciliación manual. Generar archivo complementario (delta) para enviar a cámara de compensación.

6

Aprendizaje

La confiabilidad no es solo "el sistema está arriba". Es también "los datos son correctos". Los procesos batch requieren monitoreo de calidad de datos.

Incidente 6: Fallo de HSM por Agotamiento de Claves

1

Contexto

Miércoles 2:00 PM. Tráfico normal en hora valle.

2

Síntomas

Transacciones empiezan a fallar con "MAC Invalid" y "PIN Verification Failed". ASR cae 15 puntos.

3

Métricas

Error rate en HSM sube. Métrica "remaining keys" en HSM muestra 0.

4

Logs

ERROR: HSM unable to sign. No more keys available. Error en todos los servicios que usan HSM.

5

Decisión Correcta

Activar rotación automática de claves. Implementar alerta de "remaining keys < 10%" para detección temprana.

6

Aprendizaje

El HSM es un componente crítico que puede degradarse sin caer. Monitorear métricas de capacidad (claves, sesiones) es tan importante como disponibilidad.

Roadmap de Carrera: De NOC a Staff SRE

Este es tu camino de batalla. Aquí tienes la hoja de ruta ejecutable para evolucionar de operador de NOC a Staff SRE en 24 meses, con habilidades técnicas, proyectos recomendados y métricas personales para cada etapa.

Etapa 1: NOC Operator (El Centinela)

Habilidades Técnicas

- Dominar dashboards (Grafana)
- Saber leer logs (Kibana/Loki)
- Seguir runbooks al pie de la letra
- Conocimientos básicos de Linux y redes

Habilidades de Negocio

- Entender impacto de una caída
- Comunicar estado de incidente claramente
- Triage básico de alertas

Proyectos Recomendados

- Mejorar runbook existente
- Crear script de automatización
- Documentar proceso manual

Etapa 2: SRE (El Bombero)

Habilidades Técnicas

- Programación (Python/Go)
- Infraestructura como código (Terraform)
- Gestión de contenedores (K8s)
- Conocimiento profundo de ISO 8583

Habilidades de Negocio

- Participar en postmortems
- Entender SLOs de servicios
- Colaborar con equipos de desarrollo

Proyectos Recomendados

- Liderar migración a Kubernetes
- Construir dashboard de ASR en tiempo real
- Automatizar proceso manual

Etapa 3: Senior SRE (El Arquitecto de Confiabilidad)

Habilidades Técnicas

- Diseño de sistemas distribuidos
- Estrategias de resiliencia (circuit breakers, bulkheads)
- Definición de SLOs y error budgets

Habilidades de Negocio

- Influir en roadmap de producto
- Embajador de confiabilidad
- Traducir necesidades de negocio

Proyectos Recomendados

- Diseñar estrategia active-active
- Liderar postmortem de incidente P0
- Implementar error budget policy

Etapa 4: Staff SRE (El Multiplicador)

Habilidades Técnicas

- Visión técnica a largo plazo
- Conocimiento del ecosistema completo
- Resolver problemas abstractos

Habilidades de Negocio

- Liderazgo sin autoridad
- Mentorear a otros SREs
- Traducir necesidades en estrategias

Proyectos Recomendados

- Definir estrategia de observabilidad
- Crear programa de game days
- Liderar integración post-adquisición

Plan de Estudio Autodidacta

Fundamentos Sistemas Distribuidos

"Designing Data-Intensive Applications" - Martin Kleppmann. CAP theorem, timeouts, circuit breakers, bulkheads, idempotencia.

ISO 8583

Especificación ISO 8583, guías de Visa/Mastercard, Wireshark con plugin ISO 8583 para análisis de tráfico.

SRE y Confiabilidad

"Site Reliability Engineering" y "The Site Reliability Workbook" - Google. SLIs, SLOs, Error Budgets, Toil, Postmortems.

Observabilidad

Prometheus docs, OpenTelemetry docs, "Prometheus deep dive" - CNCF, "Practical Alerting" - Grafana Labs.

Arquitectura de Pagos

Visa Developer Center, Mastercard Developers, guías de integración, flujo de autorización, clearing vs. settlement, 3D Secure.

Programación para SREs

"The Go Programming Language" - Donovan & Kernighan, Tour of Go, escribir proxy TCP que entienda ISO 8583.

Mapa de Calor de Habilidades

Checklist de Guardia SRE

- **¿Hay incidente activo o degradación?**
  Verificar dashboards y alertas pendientes
- **¿Hay cambios planificados en producción?**
  Confirmar rollback plan disponible
- **¿Los dashboards muestran métricas normales?**
  ASR, latencia, error budget dentro de rangos
- **¿He leído runbooks actualizados?**
  Servicios core y procedimientos de emergencia
- **¿Tengo acceso a todos los sistemas?**
  VPN, consolas, servidores de logs, páginas
- **¿El teléfono tiene batería?**
  Aplicación de páginas funciona correctamente

Plantilla de Postmortem

```
Título: [Fecha] - [Breve descripción]
Estado: Borrador | Final
Dueño: [Nombre]
Participantes: [Nombres]

Resumen del Impacto:
- Duración: [Tiempo]
- Usuarios afectados: [Porcentaje/tipo]
- Transacciones fallidas: [Número/impacto económico]

Línea de Tiempo (UTC):
- [Hora] Evento A
- [Hora] Alerta disparada
- [Hora] Ingeniero asignado
- [Hora] Acción de mitigación X
- [Hora] Servicio restaurado

Causa Raíz:
[Explicación técnica detallada]

Por Qué el Incidente Fue Severo (5 Whys):
1. Por qué? ...
2. Por qué? ...

Acciones:
- Corto Plazo: Arreglar el bug. (Dueño: A, Fecha: DD/MM)
- Mediano Plazo: Mejorar alerta. (Dueño: B, Fecha: DD/MM)
- Largo Plazo: Rediseñar componente. (Dueño: C, Fecha: DD/MM)
```

> *"Cuando el teléfono suene a las 3 a.m. y veas un error DE39: 91, no entres en pánico. Respira. Sigue el proceso. Y recuerda que detrás de cada uno de esos mensajes, hay una persona tratando de comprar un café, pagar una cena o hacer una transferencia para llegar a fin de mes. Nuestro trabajo es asegurar que esa experiencia sea invisible y confiable."*