# Arquitetura de Observabilidade SRE para Mobilidade Corporativa Android Enterprise (Datadog + Azure + Integrações Microsoft/Cisco/Salesforce)

Arquiteto de Soluções Sênior (SRE & Observabilidade)

06 de fevereiro de 2026

## Sumário

## Resumo Executivo de Valor

**Problema:** falta de correlação fim a fim (Dispositivo → Rede → Identidade → Aplicação) torna o suporte reativo, aumenta o tempo de diagnóstico e dificulta mensurar a disponibilidade percebida.

**Solução:** centralizar telemetria no Datadog com ingestão Azure-native e operação SRE (SLIs/SLOs + automação), habilitando correlação multi-camada e governança por Error Budget.

**Metas executivas (baseline a validar): redução de 60% no MTTR** e **redução de 40% na abertura de tickets** (suporte mobilidade).

**Benefícios esperados (alto nível):**

- **Diagnóstico mais rápido**: correlação determinística reduz troca de contexto e tempo de triagem.
- **Operação proativa**: SLOs e burn rate antecipam degradações antes do pico de tickets.
- **Governança e eficiência**: padrões de dados, retenção e logs-to-metrics reduzem ruído e custo por sinal.

**Métricas-chave (resumo):**

| Métrica | Meta |
|---|---|
| MTTR (incidentes mobilidade) | $-60\%$ |
| Redução de tickets (suporte mobilidade) | $-40\%$ |
| Cobertura de correlação (`IntuneDeviceId`) | $> 95\%$ |
| SLO mensal de conectividade VPN | $\geq 99.0\%$ |

# Trilha 1: Fundamentos e Justificativa

## Visão Executiva e Estratégica

### Contexto executivo (síntese)

O detalhamento de problema, solução, metas e benefícios está consolidado no **Resumo Executivo de Valor** (antes do sumário). Nesta trilha, o foco passa a ser a fundamentação, a definição do serviço e as decisões de arquitetura/operabilidade.

### Fundamentação bibliográfica de autoridade

Além das documentações de produto, as metodologias aplicadas nesta iniciativa são fundamentadas em obras de referência:

- **SRE (Error Budgets e SLIs/SLOs)**: o uso de SLIs/SLOs, Error Budgets e Burn Rate como mecanismo de governança de confiabilidade segue a abordagem consolidada em *Site Reliability Engineering* [7].
- **Cultura de entrega (automação e monitoramento contínuo)**: a automação operacional e a observabilidade como pilares de agilidade e estabilidade operam em consonância com práticas descritas em *The DevOps Handbook* [9].
- **Arquitetura cloud (pilares e trade-offs)**: o desenho e as decisões arquiteturais (Confiabilidade, Segurança e Otimização de Custos) são alinhados ao *Azure Well-Architected Framework* [14].

### Definição do serviço e jornadas críticas

Para fins de SRE, o "serviço" observado nesta proposta é o **Serviço de Mobilidade Corporativa Android Enterprise** (dispositivos gerenciados com acesso a aplicações corporativas), com as seguintes jornadas como referência de experiência do usuário:

- **Acesso (identidade)**: autenticação via Entra ID e aplicação de políticas (ex.: CA).
- **Conectividade segura**: estabelecimento e manutenção de sessão Cisco VPN.
- **Saúde e conformidade**: sincronização, compliance e capacidade de receber políticas/comandos (Intune).
- **Aplicação crítica**: autenticação e uso do Salesforce (telemetria conforme disponível).

### Escopo e exclusões (in-scope / out-of-scope)

**In-scope:**

- Dispositivos Android Enterprise **ativos** (enrollment ativo e elegíveis para políticas), e suas interações com Intune/Entra/VPN/Salesforce.
- Observabilidade da **plataforma de ingestão e processamento** (Azure + Datadog) e sua qualidade de dados.

**Out-of-scope (tratados como exclusões ou categorias separadas):**

- Eventos estritamente de comportamento do usuário (ex.: senha incorreta), e dispositivos descomissionados/perdidos.
- Indisponibilidade de operadora/last-mile quando não houver evidência de falha na cadeia corporativa (medida separadamente quando possível).

### Premissas de validade e "freshness" dos sinais

Como as fontes possuem atrasos naturais (ex.: *sync* do Intune/Graph e economia de energia do Android), SLIs/SLOs devem declarar **janelas de observação** e **critérios mínimos de atualidade** (*freshness*) por fonte, evitando interpretar "ausência de sinal" como falha do serviço.

### Alinhamento com Azure Well-Architected Framework e SRE (Google)

- **Reliability**: desenho com resiliência (Geo-DR, DLQ, retries), estado tri-valorado e governança por SLO/Error Budget [7].
- **Security**: RBAC, Managed Identities, proteção de PII (LGPD) na entrada, auditoria e segregação [14].
- **Cost Optimization**: Flex Logs, logs-to-metrics, retenções por criticidade e guardrails de cardinalidade [14].
- **Operational Excellence**: runbooks, automação (auto-healing), integração com ITSM e critérios de escalonamento [9].
- **Performance Efficiency**: ingestão/processing serverless e *delta queries* (Microsoft Graph) para reduzir overhead [14].

## 1. Contexto e Problema de Negócio

### Fragmentação de visibilidade

O ecossistema de mobilidade corporativa (aprox. 5.000 tablets Android Enterprise) depende de sistemas que geram sinais operacionais relevantes, porém isolados:

- **Intune**: saúde, inventário e conformidade.

- **Entra ID**: autenticação, políticas de acesso e sinais de identidade.
- **Cisco VPN**: conectividade, sessões e motivos de falha.
- **Salesforce**: aplicação crítica de negócio.

A ausência de correlação determinística (*Device* → *Network* → *Identity* → *Application*) aumenta o tempo de diagnóstico e dificulta afirmar, com precisão, a disponibilidade do serviço percebida pelo usuário.

### Modelo reativo e impacto operacional

O modelo atual é reativo: incidentes são frequentemente identificados por tickets/contatos de usuários. Isso eleva o **MTTR**, multiplica o esforço manual (troca de contexto entre consoles) e amplia o risco reputacional (impacto em força de vendas e atendimento).

**Caso de uso (mobilidade em campo, exemplo narrativo):** um vendedor em rota tenta registrar um credenciamento no Salesforce e falha (VPN/intermitência/CA), abrindo chamado sem evidência correlacionada; o suporte alterna entre consoles (Intune/Entra/VPN) e o diagnóstico chega tarde.

Com a plataforma, o NOC vê em um único dashboard a jornada *Device* → *VPN* → *Identity* → *Salesforce*, com `join_confidence` e motivo de falha, e aciona o runbook correto (ou automação) antes de o impacto se espalhar.

## 2. Visão da Solução

### Plataforma unificada de observabilidade

A solução consolida telemetria no **Datadog** como plano único de observabilidade (logs, métricas, monitores, SLOs, dashboards), recebendo dados por uma camada de ingestão **Azure-native** (Event Hubs, Functions, Container Apps) e integrações com **Microsoft Graph**, **Intune/Entra Diagnostic Settings**, **Cisco syslog** e sinais de aplicação (Salesforce).

### Correlação multi-camada e transição para SRE

O desenho foca em **correlação multi-camada** e mudança operacional para SRE:

- **SLIs/SLOs** orientados ao usuário.
- **Error Budgets** para governar mudanças (release, políticas, rollout) e priorização.
- **Automação** para reduzir ruído e padronizar resposta.

**O Diferencial da Solução — Identity Stitching via SCEP (SAN + `IntuneDeviceId`):** ao usar o `IntuneDeviceId` no SAN do certificado, a correlação entre VPN (Cisco) e inventário (Intune) torna-se determinística sem depender de identificadores de hardware (restritos por privacidade no Android 12+), reduzindo ambiguidade no diagnóstico e aumentando a confiabilidade dos SLIs/SLOs.

### Mecanismo de correlação (stitching) e qualidade do *join*

A correlação multi-camada deve ser tratada como um produto com **contrato de dados**:

- **Chave primária (determinística)**: `IntuneDeviceId` presente no SAN do certificado SCEP.
- **Chaves alternativas (fallback)**: UPN/usuário (Entra), *device name* (com normalização), e/ou outros identificadores corporativos **somente quando documentados** e com risco conhecido de colisão.
- **Confiança do join**: cada evento correlacionado recebe um atributo `join_confidence` (`high`/`medium`/`low`) para separar análises e evitar conclusões incorretas.

**Identity Stitching determinístico (entendimento e configuração em 3 camadas):** este é o mecanismo central que elimina ambiguidade entre logs de rede e gestão do dispositivo. Diferentemente de correlações por IP (volátil) ou por usuário (múltiplos dispositivos), utiliza um identificador **imutável** injetado **criptograficamente** no certificado do dispositivo.

**Camada 1 — Intune (padronização do certificado SCEP):**

- No perfil SCEP (Android Enterprise), configurar **SAN** com atributo **URI** e valor `ID:Microsoft Endpoint Manager:GUID:{{DeviceID}}` (ou variação equivalente), garantindo que `{{DeviceID}}` seja substituído pelo GUID real.
- **Requisito crítico de PKI (NDES/CA):** template com **"Supply in the request"** em *Subject Name*, para que a CA não ignore o SAN enviado pelo Intune.

**Camada 2 — Cisco ASA/FTD (mapeamento e extração no handshake VPN):**

- Criar **attribute-map** que lê `subjectAltName`, aplica Regex e mapeia o GUID para um atributo visível em logs (ex.: `IETF-Radius-Class`).
- Associar o mapa ao *tunnel-group* (connection profile) dos tablets e garantir **logging class vpn** no nível adequado.

**Camada 3 — Datadog (pipeline + `join_confidence`):**

- Pipeline Cisco: extrair `intune_device_id` do campo mapeado e remapear para `device.id`.
- Classificar confiança do *join*:
  - **High**: log contém `intune_device_id` (derivado do certificado/SAN).

– **Medium**: sem device id, mas com `user.id` correlacionável (ex.: UPN) com sinais de Entra na mesma janela.

– **Low**: correlação apenas por IP/tempo (maior risco de colisão, e.g., CGNAT).

**Resultado operacional:** em incidentes de VPN, filtrar por `join_confidence:high` reduz risco de ação sobre o usuário/dispositivo errado e aumenta a precisão de *postmortems* e automações.

**Controles recomendados:**

- Validação do template SCEP (SAN obrigatório e formato padronizado).
- Monitoramento de **cobertura de correlação** (% de eventos VPN com `IntuneDeviceId` válido) e alarmes quando cair abaixo do patamar definido.
- Tratamento explícito de rotação/expiração de certificado (prevenção de "quebra silenciosa" do stitching).

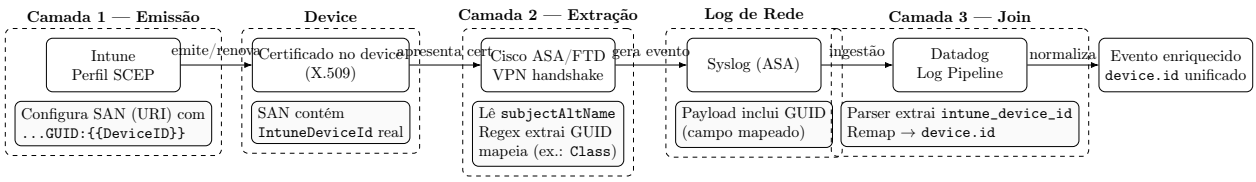


Figura 1: Identity stitching determinístico: o `{{DeviceID}}` do perfil SCEP vira `IntuneDeviceId` no SAN do certificado, é extraído pelo Cisco ASA e chega ao Datadog para remapping em `device.id`.

# Trilha 2: Engenharia e Arquitetura

# 3. Arquitetura Técnica (Detalhada)

## 3.1 Visão em camadas

**Camada de fontes (sinais):** Intune/Entra (logs e inventário), Cisco VPN (syslog/eventos), Salesforce (sinais de aplicação e autenticação), e metadados de PKI/SCEP.

**Camada de ingestão (Azure):**

- **Azure Event Hubs** com **Geo-Disaster Recovery** (continuidade e replay).
- **Azure Functions** para **delta queries** e extrações periódicas via Microsoft Graph.
- **Azure Container Apps** para processamento/forwarding (substituição de VMs), escalável com KEDA.

**Camada de processamento (Datadog):**

- **Log Pipelines** (parsing/normalização/enrichment).
- **Reference Tables** (enrichment de contexto: BU, região, modelo, operadora, etc.).
- **Sensitive Data Scanner** e políticas de redaction/masking.
- **Flex Logs** e **Logs-to-Metrics** para otimização (FinOps) e priorização de indexação.

**Camada de ação (auto-healing): Logic Apps** (ou automações equivalentes) para remediação, criação/atualização de incidentes e integrações ITSM.

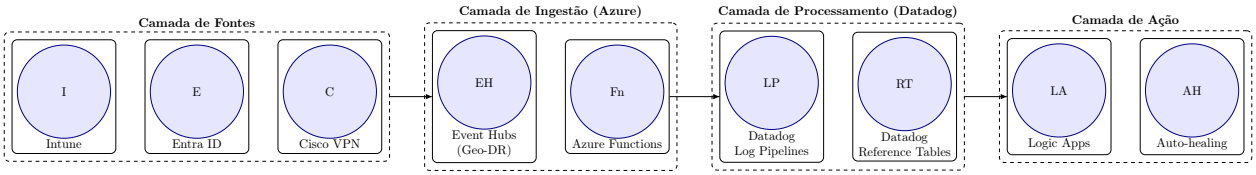


Figura 2: Fluxo horizontal simplificado em 4 camadas: Fontes → Ingestão (Azure) → Processamento (Datadog) → Ação (Auto-healing).

### 3.2 Fluxos de dados (end-to-end)

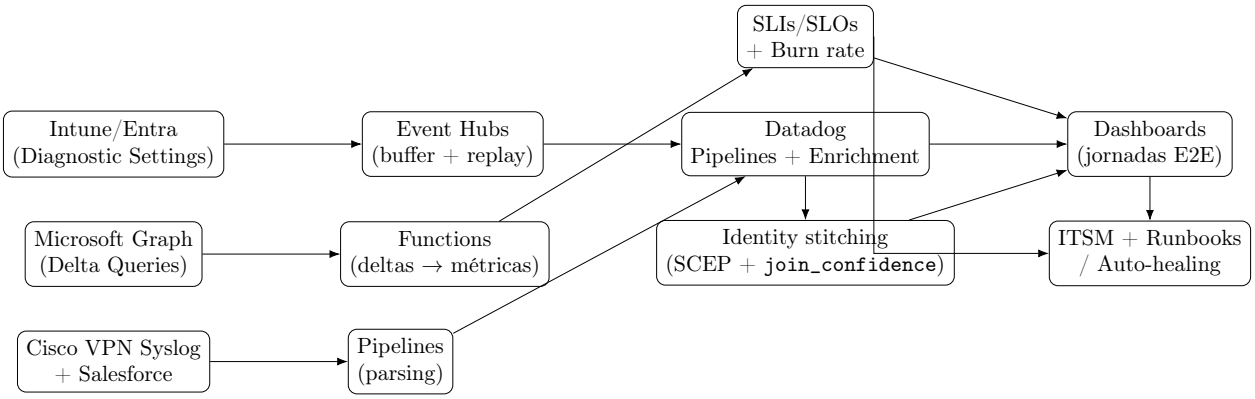


Figura 3: Fluxo E2E: sinais → ingestão/processamento → correlação → SLIs/SLOs → ação.

**Quadro resumo de ingestão (comparativo):**

| Fonte | Transporte | Processador | Destino | Valor de Negócio |
|---|---|---|---|---|
| Intune/Entra (Diagnostic Settings) | Event Hubs | Container Apps (consumers) | Datadog Logs/Pipelines | Visibilidade unificada, auditoria e detecção precoce de falhas (identidade e compliance). |
| Microsoft Graph (delta queries) | HTTPS (Graph API) + Table Storage (estado) | Azure Functions | Datadog Métricas/Events | Saúde de frota e SLIs/SLOs com baixo overhead (somente deltas). |

| | | | | |
|---|---|---|---|---|
| Cisco VPN (syslog) | Syslog/CEF | Datadog Pipeline (parsing) + stitching | Datadog Logs/Métricas | Medir disponibilidade real de conectividade e correlacionar falhas por device/usuário/região. |
| Salesforce (sinais disponíveis) | API/Logs (conforme fonte) | Normalização/enriquecimento | Datadog (Logs/Métricas) | SLIs de aplicação crítica e correlação com identidade/rede. |

**Fluxo A: Intune/Entra Diagnostic Settings → Event Hubs → Datadog**

1. Logs operacionais/auditoria são exportados via Diagnostic Settings para Event Hubs.
2. Consumer(s) em Container Apps consomem e encaminham ao Datadog.
3. Pipelines normalizam campos (timestamp, tenant, category), extraem atributos e aplicam proteção de PII.

**Fluxo B: Microsoft Graph Delta Queries (inventário/estado) → métricas**

1. Azure Function executa a cada 15 minutos e consulta **apenas deltas** (mudanças) usando *deltaLink*.
2. Mudanças são convertidas em métricas/atributos para Datadog.
3. O *deltaLink* é persistido (Table Storage) e monitorado (health widget) para evitar *stale state*.

**Fluxo C: Cisco VPN syslog → parsing → correlação (SCEP)**

1. Eventos de VPN (sucesso/falha) entram via syslog (ou transporte equivalente) e são parseados.
2. O certificado do dispositivo (SCEP) expõe identificadores no SAN, incluindo `IntuneDeviceId`.
3. O Datadog realiza **identity stitching**: sessão VPN *joins* com dispositivo Intune, usuário (Entra) e contexto corporativo.

**Meta-observabilidade (qualidade e saúde do pipeline)**

Além de observar as fontes, a proposta deve observar a **plataforma de observabilidade** (ingestão e processamento) para garantir confiança nos SLIs/SLOs.

**Sinais mínimos (por fonte e por pipeline):**

- **Freshness end-to-end**: idade do último evento processado (P50/P95) e *watermark* por partição.
- **Lag/backlog**: profundidade de fila/atraso por consumer group (Event Hubs).
- **Taxa de erro**: falhas de parse, falhas de envio ao Datadog, throttling (429) e retries.

- **DLQ/Poison messages**: volume e categoria de erros (para correção de schema/parsers).
- **Cardinalidade de atributos/tags**: guardrails para evitar explosão de cardinalidade e degradação de custo/performance.

### 3.3 Pontos de resiliência e pontos críticos de falha

**Resiliência (design):**

- Event Hubs com retenção estendida para **replay** e isolamento por **consumer groups**.
- Processing em Container Apps com **min replicas** e auto-scaling por backlog.
- Retry exponencial e circuit breaker em chamadas ao Graph (mitiga HTTP 429).
- DLQ/erros de parsing segregados (observabilidade do pipeline).

**Pontos críticos (a governar):**

- **Expiração do deltaLink** (risco de `HTTP 410 Gone` após inatividade prolongada).
- **Dependência de APIs externas** (Graph, Intune, Salesforce, Cisco).
- **Falha de correlação SCEP** (certificados fora do padrão, SAN ausente, ou inconsistência de IDs).
- **Doze Mode** (Android) causando falsos positivos de "dispositivo inativo".

## 4. Modelo de Confiabilidade (SRE)

### 4.1 Conceitos e governança por Error Budgets

O modelo estabelece SLOs orientados ao usuário. O orçamento de erro (mensal) governa:

- **priorização** (incidentes vs. roadmap),
- **controle de mudanças** (rollouts, policies, versões),
- **automação** (remediações autorizadas por impacto).

### 4.2 Estado tri-valorado (Healthy / Unknown / Down)

Para reduzir ruído e refletir incerteza observacional (ex.: Doze Mode), adota-se estado tri-valorado:

- **Healthy**: sinais recentes e coerentes (sync/uso).

- **Unknown**: ausência parcial de sinais (provável economia de energia).
- **Down**: ausência consistente e impactante (exige ação).

**Calibração do modelo tri-valorado (estratégia SRE):** o objetivo é mitigar falsos positivos gerados por *Doze Mode* e otimizações agressivas de bateria (ex.: One UI). Em vez de tratar disponibilidade como binária, o modelo introduz uma zona de amortecimento (**Unknown**) baseada em **evidências cruzadas**.

**Janelas recomendadas (baseline a calibrar por telemetria):**

- **0–8h**: operação normal do ciclo de check-in/policy do Intune (dispositivo tende a permanecer **Healthy**).
- **8–24h**: janela onde o Intune pode ficar "mudo" por Doze Mode, mas o usuário ainda pode estar ativo (candidato a **Unknown**).
- **>24h**: ausência prolongada; aumenta probabilidade de falha real (candidato a **Down**), salvo evidência alternativa forte.

**Evidência cruzada ("pulso" fora do agente MDM):**

- **Identidade (Entra ID):** sucessos em `SignInLogs` (`ResultType == 0`) e, quando aplicável, `NonInteractiveUserSignInLogs` (renovação silenciosa de token) sugerem dispositivo ligado e conectado.
- **Rede (Cisco VPN):** presença de sessão ativa/recente (ex.: evento de estabelecimento `722022` sem evento de encerramento correspondente e/ou tráfego recente quando disponível) confirma conectividade corporativa.
- **Aplicação crítica (Salesforce):** sucesso de autenticação/uso (quando instrumentado) indica disponibilidade operacional percebida pelo usuário.

**Regra operacional de alertas: Unknown** deve ser um estado de *aviso silencioso* (dashboard/relatório), sem pager; **Down** é candidato a alertas acionáveis e impacto em burn rate.

**Resumo da decisão (pseudocódigo):**

```
IF intune_last_sync < 8h:
  status = HEALTHY
ELIF 8h <= intune_last_sync <= 24h:
  IF entra_login_success < 24h OR vpn_session_active:
    status = UNKNOWN
  ELSE:
    status = DOWN
ELIF intune_last_sync > 24h:
  IF entra_login_success < 24h OR vpn_session_active:
    status = UNKNOWN
  ELSE:
    status = DOWN

# Observao: >7 dias em DOWN -> decommissioned (excluir do denominador do SLO)
```

### 4.3 SLIs e SLOs (com coerência matemática)

**Nota de coerência:** todo SLI que dependa de logs/sync deve declarar (i) a população elegível (denominador), e (ii) critérios de *freshness* do dado; quando esses critérios não forem atendidos, o estado deve transitar para **Unknown** (e não para falha do serviço).

#### SLI 0: Freshness do pipeline (fontes → Datadog)

$$\text{Pipeline Freshness} = \text{P95}(\texttt{now} - \texttt{event\_timestamp\_source})$$

**Objetivo:** manter a P95 de idade do sinal abaixo de um limiar definido por fonte (ex.: minutos para logs de VPN e dezenas de minutos para inventário via Graph), com alarmes específicos por pipeline (Event Hubs, Function, Container Apps).

#### SLI 1: Fleet Compliance Health (Intune)

$$\text{Fleet Compliance Health} = \frac{\text{count(devices where compliance\_state} = \texttt{compliant})}{\text{count(devices)}} \times 100\%$$

**Regra de escopo (denominador):** considerar apenas dispositivos "vivos" (ex.: `last_sync_datetime > now() - 8h`, enrollment ativo e OS Android), evitando poluição por dispositivos órfãos.

**Critério de Unknown:** `last_sync_datetime` entre 8h e 24h **e** presença de sinal alternativo recente (ex.: último login Entra ou sessão VPN $\leq$ 24h), sugerindo Doze Mode/uso intermitente.

**Critério de Down:** `last_sync_datetime > 24h` **e** ausência de sinais alternativos (login Entra e VPN) no mesmo período, indicando provável indisponibilidade operacional do device (ou perda de gerenciamento).

**SLO sugerido:** $\geq 95\%$ (mensal), sujeito a baseline e segmentação por BU/região.

**Mitigação de falsos positivos de *Unknown* em tablets Samsung (One UI/Knox):** a **One UI** pode ser agressiva ao encerrar processos em segundo plano para economia de bateria, o que frequentemente interrompe o app/agente do Intune (Company Portal) e aumenta *Unknown* por atraso de `last_sync_datetime`.

**Solução automatizada (recomendada para escala):** usar **OEMConfig** da Samsung via **Knox Service Plugin (KSP)**.

- Criar um perfil de configuração no Intune com o app **Knox Service Plugin**.
- No schema do KSP, configurar **Battery optimization allowlist** (ou **Unmonitored apps**) e adicionar o package: `com.microsoft.windowsintune.companyportal`.
- (Opcional) adicionar também `com.cisco.anyconnect.vpn.android.avf` para reduzir quedas de VPN em standby.

**Solução manual (quando KSP não se aplica, ex.: BYOD/Work Profile):** orientar o usuário/suporte a configurar o Company Portal como **Irrestrito/Não restrito** em **Configurações → Aplicativos → Portal da Empresa → Bateria**.

**Validação na observabilidade (Datadog):**

- Correlacionar nível de bateria (quando disponível via Graph) com atraso de sync para diferenciar hibernação (software) vs. desligamento (bateria).
- Manter **Unknown** entre 8h e 24h e promover para **Down** somente com sinais corroborantes (ex.: falha de login no Entra) ou atraso $> 24$h.

### SLI 2: VPN Availability (Cisco) com exclusões de erro

**Definição (notação descritiva):** conexões bem-sucedidas / (todas as tentativas de conexão − tentativas excluídas por motivos que não caracterizam falha da infraestrutura), multiplicado por 100.

$$\text{VPN Availability} = \frac{\text{Conexões bem-sucedidas}}{\text{Tentativas totais} - \text{Tentativas excluídas}} \times 100\%$$

**Exclusões do denominador (ajustar à taxonomia de `reason` adotada):**

| `reason` (excluído) | Justificativa |
|---|---|
| `wrong_password` | Erro de credencial (comportamento do usuário/identidade), não indisponibilidade da infraestrutura VPN. |
| `expired_certificate` | Certificado expirado/fora de conformidade (PKI/SCEP/política), mede governança de credenciais, não a saúde do concentrador. |
| `disconnect_by_user` | Desconexão voluntária (evento esperado), não falha de conectividade. |
| `user_cancelled` | Operação cancelada pelo usuário/dispositivo (evento esperado), não falha do serviço. |
| `profile_misconfiguration` | Falha por configuração/política (ex.: perfil, CA, deploy), deve ser monitorada à parte como qualidade de mudança. |

**Racional técnico das exclusões:** protege a integridade do SLI ao medir **confiabilidade da infraestrutura VPN** (rede/concentradores) e não fatores externos (comportamento do usuário, governança de credenciais ou erros de rollout).

**Critério de Unknown:** volume insuficiente de tentativas na janela (ex.: $< 30$ tentativas/15 min por região/concentrator) **ou** degradação de *freshness* do pipeline de VPN (ex.: P95 de atraso $> 15$ min), tornando a medição inconclusiva.

**Critério de Down:** degradação sustentada com amostra suficiente (ex.: $\geq 5$ min com `VPN Availability` abaixo do SLO por região/concentrator) **e** aumento correlacionado de falhas não-excluídas (ex.: `reason:timeout`, `reason:handshake_failed`).

**SLO sugerido:** $\geq 99.0\%$ (mensal), com alertas de Burn Rate (fast/slow).

#### SLI 3: Acesso ao Salesforce (experiência do usuário)

SLI definido por sucesso de autenticação/uso (conforme telemetria disponível):

- **Disponibilidade lógica**: taxa de sucesso de login e carregamento inicial.
- **Latência**: P95 para etapas críticas (quando instrumentável).

**Critério de Unknown:** indisponibilidade/atraso de telemetria de aplicação (ex.: fonte/API sem eventos recentes acima do limiar de *freshness*) **enquanto** sinais de identidade (Entra) e conectividade (VPN) permanecem saudáveis.

**Critério de Down:** taxa de falha de login/uso acima do limiar por janela (ex.: $\geq 5$ min) **com** volume mínimo de amostras, ou aumento sustentado de erros de autenticação/HTTP (conforme sinal disponível), indicando indisponibilidade percebida pelo usuário.

**SLO sugerido:** $\geq 99.5\%$ (mensal).

### 4.4 Burn Rate Alerts (Fast/Slow)

Definir dois níveis por SLO (exemplo genérico):

- **Fast burn**: consumo acelerado do budget (sinal de incidente severo em curso).
- **Slow burn**: degradação gradual (sinal de tendência/"morte lenta").

## Trilha 3: Governança, Risco e Conformidade

## 5. Segurança e Compliance (LGPD)

### 5.1 Princípios: privacidade por design e minimização de dados

A arquitetura trata logs como dados de alto risco: coleta mínima necessária, proteção na entrada e controles de acesso por necessidade.

### 5.2 Detecção e proteção de PII

**PII detection patterns** (exemplos a ajustar por contexto): CPF, e-mail pessoal, identificadores diretos, tokens/segredos.

**Técnicas:**

- **Hashing** irreversível para identificadores sensíveis (quando necessário para correlação estatística).
- **Redaction** (remoção) para segredos/credenciais.
- **Partial masking** (ex.: `***@***`) para e-mails.

### 5.3 Controles e auditoria

- **RBAC** por função (SRE, Segurança, Suporte N1/N2, Gestão).
- **Managed Identities** para acesso a Key Vault e recursos Azure (evita chaves estáticas).
- **Auditoria automatizada** (acesso, mudanças em pipelines/monitores, exportações).
- **Proteção contra vazamento de logs**: bloqueio/alerta para padrões sensíveis e governança de retenção.

### 5.4 Classificação de dados, retenção e minimização

Para conformidade com LGPD e para reduzir risco operacional, recomenda-se formalizar uma política de dados por tipo de sinal:

| Fonte/tipo | Risco | Governança (exemplos) |
|---|---|---|
| Logs de autenticação (Entra) | Alto | Minimizar atributos; mascarar identificadores sensíveis; restringir acesso a times específicos; retenção curta para conteúdo detalhado e agregações para métricas. |
| Eventos de VPN (Cisco) | Médio | Evitar PII em payload; manter atributos necessários para troubleshooting; aplicar redaction para campos livres; limitar exportações. |
| Inventário/estado (Intune/Graph) | Médio | Persistir apenas atributos necessários a SLIs; evitar armazenar campos não usados; versionar schema; limitar retenção de dados brutos. |
| Telemetria de aplicação (Salesforce) | Variável | Preferir agregações; separar índices por criticidade; controlar acesso por necessidade de negócio; rotacionar segredos/tokens. |

## 6. Automação Operacional

### 6.1 Integração ITSM e critérios de escalonamento

Integração com ITSM deve:

- deduplicar incidentes (uma causa raiz, múltiplos sintomas),
- anexar evidências (logs/metrics/sessões correlacionadas),

- aplicar severidade baseada em impacto (SLO/Burn Rate + volume regional).

### 6.2 Playbook: dispositivo *stale*

**Gatilho:** dispositivo em estado **Down** (sem sync e sem uso) acima do limiar definido.

**Ações automatizadas (Logic Apps):**

1. Verificar sinais alternativos (último login Entra, última sessão VPN, atividade Salesforce) para distinguir **Unknown** vs. **Down**.
2. Se elegível, acionar sync/remediação remota (Intune) e notificar usuário (ex.: Teams).
3. Se persistir por $> 7$ dias: abrir ticket para coleta/recuperação e excluir do denominador de SLO (*decommissioned*).

### 6.3 Playbook: instabilidade regional de VPN

**Gatilho:** aumento anômalo de falhas de VPN em uma região/operadora.

**Ações:**

1. Agrupar falhas por **região**, **concentrator**, **ISP** e **reason** (exclusões preservadas).
2. Criar incidente único "Falha Regional" com evidências e percentil de impacto.
3. Se burn rate fast: escalar para Networking & Segurança e acionar comunicação executiva.

## 7. Plano de Implementação

### 7.1 Fase 1: Hardening da Infraestrutura

**Capacidades que passam a existir:**

- **Capacidade de reprocessar backlog sem perda de dados e sem VM**: ingestão resiliente (replay) em Event Hubs + processamento elástico em Azure Container Apps.

#### 7.1.1 Setup de Azure Event Hubs (Geo-DR)

**Objetivos:**

- Criar namespace com Geo-Disaster Recovery (Geo-DR) e retenção para replay.
- Isolar workloads por *consumer groups*.

- Implementar autorização via Managed Identity e RBAC.

**Atividades técnicas:**

| ID | Atividade | Responsável |
|---|---|---|
| 1.1 | Provisionamento do Event Hub Namespace (Standard Tier). | Cloud Engineer |
| 1.2 | Configuração de Geo-Disaster Recovery pairing. | Cloud Engineer |
| 1.3 | Criação de Event Hubs dedicados (ex.: `intune-logs`, `entra-logs`, `graph-metrics`). | Cloud Engineer |
| 1.4 | Configuração de Consumer Groups para isolamento de workloads. | Cloud Engineer |
| 1.5 | Setup de Managed Identity e permissões RBAC. | Security Engineer |

**Entregáveis:**

- Event Hub Namespace em modo Geo-DR.
- Runbook de Disaster Recovery (failover + rollback).
- Templates IaC (Terraform/Bicep) versionados.

**Critérios de aceite (exemplos):**

- Failover manual validado.
- Latência de ingestão P95 dentro de limiar acordado.
- Throughput suportado para o cenário de frota.

### 7.1.2 Migração para Azure Container Apps (processamento/forwarding)

**Objetivos:**

- Eliminar dependência de VMs Linux (redução de patching e *ops overhead*).
- Auto-scaling baseado em KEDA (backlog do Event Hub).
- Separar segredos (DD API Key) no Key Vault via Managed Identity.

**Atividades técnicas:**

| ID | Atividade | Responsável |
|---|---|---|
| 1.6 | Containerização do forwarder/agent (imagem no ACR). | DevOps Engineer |
| 1.7 | Provisionamento do Azure Container Apps Environment. | Cloud Engineer |
| 1.8 | Configuração de scaler KEDA para trigger do Event Hub. | DevOps Engineer |

| | | |
|---|---|---|
| 1.9 | Implementação de health probes (liveness/readiness). | DevOps Engineer |
| 1.10 | Managed Identity para acesso ao Key Vault (segredos Datadog). | Security Engineer |

**Configuração de exemplo (YAML - Azure Container Apps):**

```
properties:
  configuration:
    activeRevisionsMode: Single
    ingress:
      external: false
      targetPort: 8080
    secrets:
      - name: datadog-api-key
        keyVaultUrl: https://<keyvault>.vault.azure.net/secrets/dd-api-key
    registries:
      - server: <acr>.azurecr.io
        identity: <managed-identity-id>
  template:
    containers:
      - image: <acr>.azurecr.io/datadog-agent:7.50.0
        name: datadog-forwarder
        env:
          - name: DD_API_KEY
            secretRef: datadog-api-key
          - name: DD_SITE
            value: datadoghq.com
        resources:
          cpu: 1.0
          memory: 2Gi
    scale:
      minReplicas: 2
      maxReplicas: 10
      rules:
        - name: eventhub-scaler
          custom:
            type: azure-eventhub
            metadata:
              eventHubName: intune-logs
              consumerGroup: datadog-consumer
              unprocessedEventThreshold: '100'
```

**Critérios de aceite (exemplos):**

- Auto-scaling validado por backlog/eventos.
- Deploy com zero downtime (revisões).
- Telemetria do container visível no Datadog.

### 7.1.3 Delta Queries (Microsoft Graph) com fallback automático

**Objetivos:**

- Reduzir custo/risco de full scans (delta queries).
- Mitigar throttling (HTTP 429) e lidar com expiração de *deltaLink* (HTTP 410 Gone).
- Persistir estado em Table Storage e publicar *health signals* (freshness, fallback rate).

**Lógica revisada (revisão.md):** detectar `HTTP 410 Gone` / `ResourceNotFound` e executar fallback automático para **full sync**, com evento de visibilidade operacional.

**Pseudocódigo (Python) – versão com resiliência:**

```
import time

GRAPH_SCOPE = "https://graph.microsoft.com/.default"
DELTA_STATE_PK = "partition"
DELTA_STATE_RK = "devicesDelta"


def run_intune_inventory_delta_sync(timer_context, state_table, log):
    # 1) Ler estado (deltaLink)
    state = state_table.get(partition_key=DELTA_STATE_PK, row_key=DELTA_STATE_RK)  #
    pode ser None
    delta_link = state.get("deltaLink") if state else None

    graph = get_authenticated_graph_client(scope=GRAPH_SCOPE)

    was_full_sync = False
    processed_count = 0

    try:
        # 2) Buscar pgina inicial
        if not delta_link:
            log.info("First run detected. Performing full sync (delta endpoint without
    deltaLink).")
            page = graph.intune.managed_devices_delta()
        else:
            log.info(f"Attempting delta sync with link: {delta_link}")
            page = graph.request(delta_link)

        # 3) Processar paginao
        while page is not None:
            for device in page.items:
                send_to_datadog_metrics(device)
                processed_count += 1

            page = page.next_page()  # None quando terminar

        new_delta_link = page.delta_link if page else graph.last_delta_link
```

```
except GraphHttpError as ex:
    # deltaLink invlido/expirado: Microsoft Graph retorna 410 Gone (ou
ResourceNotFound)
    if ex.status_code == 410 or ex.error_code == "ResourceNotFound":
        log.warning(f"Delta link expired or invalid. Falling back to full sync.
Error: {ex}")

        # Resetar estado e executar full sync
        if state:
            state_table.delete(partition_key=DELTA_STATE_PK, row_key=DELTA_STATE_RK
)

        page = graph.intune.managed_devices_delta()  # full sync via delta()
        was_full_sync = True

        while page is not None:
            for device in page.items:
                send_to_datadog_metrics(device)
                processed_count += 1
            page = page.next_page()

        new_delta_link = page.delta_link if page else graph.last_delta_link

        send_datadog_event({
            "title": "Intune Delta Sync - Fallback to Full Sync",
            "text": "Delta link expired. Performed full inventory sync.",
            "alert_type": "warning",
            "tags": ["service:intune", "sync_type:full_fallback"],
        })
    else:
        raise

# 4) Persistir novo estado
state_table.upsert({
    "partitionKey": DELTA_STATE_PK,
    "rowKey": DELTA_STATE_RK,
    "deltaLink": new_delta_link,
    "lastSyncTime": utcnow_iso(),
    "devicesProcessed": processed_count,
    "wasFullSync": was_full_sync,
})

log.info(f"Sync completed. Processed {processed_count} devices. Full sync: {
was_full_sync}")
```

**Entregáveis adicionais:**

- Widget "Delta Sync Health" (last success, devices processed, flag de fallback).
- Alertas de *freshness* e de fallback (anormalidade).

## 7.2 Fase 2: Observabilidade (fontes, pipelines e correlação)

**Capacidades que passam a existir:**

- **Capacidade de ver, em um único dashboard, Device → VPN → Identity → Salesforce com** `join_confidence`: correlação determinística via SCEP e enriquecimento padronizado em pipelines/reference tables.

### 7.2.1 Diagnostic Settings (Intune/Entra ID) → Event Hubs

Para ativar o envio de logs do Microsoft Intune e do Entra ID diretamente para o Azure Event Hub, utiliza-se o recurso de **Diagnostic Settings** (Configurações de Diagnóstico) presente em ambos os serviços. Este processo desacopla a geração do log do seu consumo, permitindo que o Datadog ingira esses dados em tempo real.

**1. Pré-requisitos**

- **Azure Event Hub Namespace**: deve existir previamente no Azure. Recomenda-se o nível **Standard** (ou superior) para suportar *Geo-Disaster Recovery* e escala.
- **Permissões**: perfil **Intune Service Administrator** (ou Global Admin) e permissão de escrita no namespace do Event Hub (ex.: *Owner* ou *Contributor*).

**2. Configuração no Microsoft Intune (logs operacionais e auditoria)**

1. Acesse o **Intune admin center**.
2. Navegue em **Reports** → **Diagnostic settings**.
3. Selecione **Add diagnostic setting** e defina um nome (ex.: `intune-to-eventhub-prod`).
4. Marque as categorias essenciais:
   - **AuditLogs**: mudanças em políticas, atribuições e ações administrativas.
   - **OperationalLogs**: sucessos/falhas de enrollment e eventos operacionais relevantes.
   - **DeviceComplianceOrg** (recomendado): relatórios de conformidade.
   - **IntuneDevices**: inventário de dispositivos (pode ter latência inicial maior que logs operacionais).
5. Em **Destination details**, selecione **Stream to an event hub** e escolha **Subscription**, **Event Hub Namespace** e a política (preferencialmente uma política com permissão *Send*).
6. Clique em **Save**.

**3. Configuração no Microsoft Entra ID (logs de login e auditoria)**

1. Acesse o **Microsoft Entra admin center**.

2. Navegue em **Identity** → **Monitoring & health** → **Diagnostic settings**.
3. Selecione **Add diagnostic setting**.
4. Marque as categorias:
   - **SignInLogs**: autenticações, falhas de MFA e erros de Acesso Condicional.
   - **AuditLogs**: mudanças em usuários, grupos e políticas.
   - **NonInteractiveUserSignInLogs** (opcional): renovações silenciosas de token em mobile.
5. Selecione **Stream to an event hub** e aponte para o namespace e um Event Hub (tópico) dedicado (recomendado separar de Intune para facilitar isolamento no processamento).
6. Clique em **Save**.

**Notas importantes de arquitetura**

- **Região**: preferir Event Hub na mesma região dos recursos de origem para reduzir latência e custos de tráfego *cross-region*.
- **Latência**: após configurar, os primeiros eventos podem levar de 15 a 30 minutos para aparecer no Event Hub (especialmente no início).

| ID | Atividade | Responsável |
|---|---|---|
| 2.1 | Configurar Diagnostic Settings no Intune (operacional/audit/compliance). | Cloud Engineer |
| 2.2 | Configurar Diagnostic Settings no Entra ID (SignInLogs/AuditLogs). | Identity Engineer |
| 2.3 | Validar fluxo fim a fim no Event Hub Explorer e no Datadog. | Cloud Engineer |

**Logs a capturar (baseline):**

| Fonte | Categoria | Evento crítico | Uso |
|---|---|---|---|
| Intune | IntuneOperationalLogs | Falha de enrollment | Detectar falhas massivas de registro |
| Intune | IntuneDeviceComplianceOrg | Mudança de compliance | Monitorar conformidade |
| Intune | IntuneAuditLogs | Mudança de policy | Auditoria/controle de mudanças |
| Entra ID | SignInLogs | Falhas (ResultType $\neq 0$) | Detectar bloqueios/CA |
| Entra ID | NonInteractiveUserSignInLogs | Refresh token (Salesforce) | Monitorar sessões silenciosas |

### 7.2.2 Log Pipelines no Datadog (normalização e enrichment)

**Objetivos:** normalizar campos, extrair atributos semânticos (Device/User/Error) e enriquecer com metadados de negócio.

| ID | Atividade | Responsável |
|---|---|---|
| 2.4 | Pipeline "Intune Logs" com Grok/JSON conforme origem. | SRE Engineer |
| 2.5 | Pipeline "Entra ID Logs" com parsing JSON e normalização. | SRE Engineer |
| 2.6 | Pipeline "Cisco Syslog" com parsing CEF/Grok (ASA). | Network Engineer |
| 2.7 | Remappers para unificar `user.id` e `device.id`. | SRE Engineer |

**Exemplo de parser (Cisco ASA):**

```
# Extrao de Syslog ID, Sesso VPN e Usurio
%{CISCO_REASON}%{SPACE}%{DATA:syslog_id}:%{SPACE}%{GREEDYDATA:message}

# Extrao do Device ID do certificado (campo SAN)
%{DATA}ID:Microsoft Endpoint Manager:GUID:%{UUID:intune_device_id}

# Parsing de disconnect reason
User:%{SPACE}%{USERNAME:vpn_user},%{SPACE}Reason:%{SPACE}%{DATA:disconnect_reason}
```

**Exemplo de remapping (JSON):**

```
{
  "name": "Unified User Identity",
  "processors": [
    {
      "type": "attribute-remapper",
      "sources": ["userPrincipalName", "CSCO_USER_NAME", "SourceUserName"],
      "target": "user.id",
      "override_on_conflict": false
    },
    {
      "type": "attribute-remapper",
      "sources": ["DeviceId", "intune_device_id", "deviceDetail.deviceId"],
      "target": "device.id",
      "override_on_conflict": false
    }
  ]
}
```

### 7.2.3 Identity stitching via SCEP (SAN + IntuneDeviceId)

**Objetivo:** correlação determinística entre logs Cisco VPN e inventário Intune, injetando `IntuneDeviceId` no SAN do certificado e extraindo esse identificador em logs.

**Como configurar o perfil SCEP no Intune para injetar `IntuneDeviceId` no SAN**

1. Criar/editar um perfil de certificado **SCEP** para **Android Enterprise** (a lógica vale também para iOS/Windows).
2. Em **Subject alternative name (SAN)**, adicionar um atributo:
   - **Tipo**: **URI**.
   - **Valor** (recomendado para extração via Regex no firewall):
     - Opção A: `IntuneDeviceId://{{DeviceID}}`
     - Opção B: `ID:Microsoft Endpoint Manager:GUID:{{DeviceID}}`
3. Confirmar que o token `{{DeviceID}}` será substituído pelo GUID real do dispositivo durante a emissão.

**Pré-requisito crítico (PKI / NDES / CA):** garantir que o template de certificado na CA aceite o SAN enviado na requisição. Em ambientes Windows CA/NDES, isso normalmente implica configurar o template com **"Supply in the request"** em *Subject Name*. Caso o template esteja como *Build from this Active Directory information*, a CA pode ignorar o SAN do Intune e o `IntuneDeviceId` não aparecerá no certificado.

**Validação:** após o deploy, o certificado no dispositivo deve conter, nas extensões SAN, a string completa (ex.: `IntuneDeviceId://a1b2c3d4-...`).

| ID | Atividade | Responsável |
|---|---|---|
| 2.8 | Ajustar perfil SCEP no Intune para incluir `{{DeviceID}}` no SAN. | Intune Admin |
| 2.9 | Configurar Cisco ASA para extrair e logar SAN do certificado. | Network Engineer |
| 2.10 | Validar taxa de correlação (meta: $> 95\%$) e criar dashboard de coverage. | SRE Engineer |

**Exemplo (Intune SCEP):**

```
{
  "subjectAlternativeNameType": "URI",
  "subjectAlternativeNameValue": "ID:Microsoft Endpoint Manager:GUID:{{DeviceID}}"
}
```

**Exemplo (Cisco ASA - logging/AAA mapping):**

```
ldap attribute-map CertificateMap
 map-name subjectAltName IETF-Radius-Class
 map-value subjectAltName "ID:Microsoft Endpoint Manager:GUID:(.*)" \1

logging enable
logging trap informational
logging host inside <syslog-server>
logging class vpn
```

**Configuração detalhada (Cisco ASA/FTD) para extrair e logar o `IntuneDeviceId` do certificado**

O objetivo é extrair o valor do **Subject Alternative Name (SAN)** do certificado SCEP apresentado na autenticação VPN e mapear esse valor para um atributo de sessão que apareça nos logs (Syslog/Accounting), permitindo correlação determinística entre rede e dispositivo.

**1. Pré-requisito (Intune/SCEP)**

- Garantir que o perfil SCEP injete o ID no SAN (formato recomendado em URI), por exemplo: `ID:Microsoft Endpoint Manager:GUID:{{DeviceID}}`.

**2. Cisco ASA (CLI) — Attribute Map com Regex**

1. Criar um **attribute-map** e capturar somente o GUID com Regex (grupo `(.*)`), armazenando no atributo mapeado (ex.: `IETF-Radius-Class`).

```
! Criar um mapa de atributos para ler o certificado (SAN)
ldap attribute-map CertificateMap
  map-name subjectAltName IETF-Radius-Class
  map-value subjectAltName "ID:Microsoft Endpoint Manager:GUID:(.*)" \1
```

**3. Aplicar o mapeamento ao Connection Profile (Tunnel Group)**

- Associar o mapa ao *tunnel-group* utilizado pelos dispositivos Android.
- Em FTD, aplicar via políticas/GUI ou FlexConfig (conforme padrão do ambiente), mantendo o mesmo conceito de mapeamento e extração do SAN.

```
! Exemplo: aplicar no tunnel-group de Android (ajuste nomes conforme ambiente)
tunnel-group <Nome_do_Tunnel_Group_Android> general-attributes
  authentication-server-group <Seu_Grupo_AAA>
  ldap-attribute-map CertificateMap
```

**4. Logging para capturar o atributo**

1. Garantir **logging class vpn** e nível adequado (ex.: `informational`) para registrar eventos de sessão (start/stop) que carreguem o atributo mapeado.

```
logging enable
logging trap informational
logging host inside <IP_do_Forwarder_Syslog>
logging class vpn
```

**Resultado esperado nos logs (conceitual):**

- **Antes**: logs trazem apenas usuário e IP, sem vínculo determinístico com o device.

- **Depois**: logs passam a incluir o atributo mapeado (ex.: `Class` / `IETF-Radius-Class`) contendo o GUID, permitindo que o pipeline Datadog extraia `intune_device_id` e realize o *join* com inventário Intune.

#### 7.2.4 Reference Tables (enrichment determinístico)

**Objetivo:** enriquecer logs e métricas com metadados (departamento, região, cost center, VIP), reduzindo *joins* em tempo de consulta.

**Schema mínimo sugerido:**

| Coluna | Tipo | Fonte | Exemplo |
|---|---|---|---|
| `device_id` | string | IntuneDeviceId | `a1b2c3d4-...` |
| `user_principal_name` | string | Graph API | `john.doe@corp.com` |
| `department` | string | Entra (User) | Sales |
| `cost_center` | string | Atributo corporativo | CC-5001 |
| `region` | string | Office/Geo | LATAM-BR |
| `vip_status` | boolean | Regra de negócio | true |

#### 7.2.5 Segurança/LGPD no pipeline (Sensitive Data Scanner)

Para conformidade com a **LGPD**, recomenda-se implementar **Privacy by Design** usando o **Datadog Sensitive Data Scanner (SDS)** na **camada de ingestão**, antes de indexar/armazenar logs. O objetivo é interceptar, classificar e transformar PII em tempo real, preservando correlação operacional sem expor dados sensíveis.

**Estratégia de ações (por tipo de dado):**

- **Hashing (SHA-256)**: ideal para identificadores únicos (ex.: CPF, IDs) quando é necessário correlacionar ocorrências sem revelar o valor.
- **Redaction**: suprimir totalmente o conteúdo (ex.: segredos/credenciais) quando não há valor operacional.
- **Partial redaction/masking**: manter parte do dado (ex.: IP mantendo /24; serial com máscara) para contexto e troubleshooting.
- **Unmask controlado (quando aplicável)**: permitir revelação somente a perfis específicos (RBAC) para auditoria, evitando acesso amplo.

**Regras recomendadas para o projeto (Android Enterprise), com regex PCRE:**

**PII a ofuscar (exemplos):**

| Tipo | Regex | Ação | Exemplo |
|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| CPF | `\d{3}.\d{3}.\d{3}-\d{2}` | Hash SHA-256 | `123.456.789-00` | → `a3c8f9...` |
| E-mail pessoal | `...@(?!corp.com)...` | Redact | `user@gmail.com` | → `***@***` |
| IP completo | `(\d+.){3}\d+` | Manter /24 | `192.168.1.50` | → `192.168.1.0/24` |
| Serial | `S/N:\s*([A-Z0-9]{10,})` | Partial mask | `ABC****567` | |

**Configuração (exemplo JSON):**

```
{
  "name": "LGPD - PII Protection for Mobile Logs",
  "is_enabled": true,
  "scanning_rules": [
    {
      "name": "CPF Hash",
      "pattern": "\\d{3}\\.\\d{3}\\.\\d{3}-\\d{2}",
      "action": { "type": "hash", "algorithm": "sha256" },
      "tags": ["pii:cpf", "compliance:lgpd"]
    },
    {
      "name": "Email Redaction",
      "pattern": "\\b[A-Za-z0-9._%+-]+@(?!corp\\.com)[A-Za-z0-9.-]+\\.[A-Z|a-z]{2,}\\b
    ",
      "action": { "type": "redact", "replacement": "***@***" },
      "tags": ["pii:email"]
    }
  ]
}
```

**Detecção avançada (opcional):** para logs não estruturados, considerar a regra **Human Name Scanner** (classificador de ML). Recomenda-se iniciar em modo de **observação** (logar matches sem transformar) para calibrar falsos positivos e, depois, aplicar a transformação (hash/redaction) conforme política.

**Governança e auditoria contínua:**

- **Auditoria automatizada**: rodar semanalmente uma query para detectar vazamentos residuais (*potential_violations*) e alertar SecOps quando $> 0$.
- **RBAC**: restringir quem pode alterar regras do scanner e quem pode ver dados potencialmente mascarados (quando aplicável).
- **Audit Trail**: habilitar trilha de auditoria no Datadog para rastrear alterações em regras/pipelines e políticas de retenção.

### 7.3 Fase 3: SLO Modeling (com exclusões e estado tri-valorado)

**Capacidades que passam a existir:**

- **SLOs com estado tri-valorado operando em produção com runbooks associados**: monitores orientados a jornada, com critérios explícitos de **Unknown** vs. **Down** e governança por burn rate/error budget.

**Correções incorporadas (revisão.md):**

- Denominadores explicitamente definidos ("dispositivos vivos") para evitar poluição por órfãos.
- Exclusões documentadas do SLI de VPN (ex.: `wrong_password`, `expired_certificate`).
- Estado tri-valorado para absorver incerteza (Doze Mode) sem inflar falhas.

### 7.4 Plano de Testes e Validação (inclui Chaos Engineering)

| Cenário | Método | Critério de aceite |
|---|---|---|
| Delta link expiration forçado | Deletar/invalidar `deltaLink` no Table Storage e simular 410 | Function deve fazer fallback full sync e sinalizar evento/alerta |
| Doze Mode simulado | Tablets com Battery Saver por período prolongado | Dispositivos devem transitar para **Unknown** (não **Down**) |
| PII leakage test | Injetar log com CPF sintético | Dado deve ser ofuscado antes de indexar e auditoria deve passar |
| SCEP certificate rollover | Renovação de certificados no piloto | Correlação Device ID deve permanecer funcional |

### 7.5 Governança e Operação (RACI e cadência)

**Matriz RACI (baseline):**

| Atividade | NOC | SecOps | Eng. Mobile | SRE | Gestão |
|---|---|---|---|---|---|
| Monitoramento de SLOs | R | I | I | A | C |
| Resposta a alertas de compliance | I | R/A | I | C | I |
| Gestão de perfis Intune | I | C | R/A | C | I |
| Otimização de custos/volume | C | I | I | R/A | C |
| Aprovação de mudanças | I | I | I | C | R/A |

**Cadência operacional (baseline):**

| Reunião | Frequência | Participantes | Objetivo |
|---|---|---|---|
| SLO Review | Semanal | SRE, NOC Lead | Análise de consumo de error budget |

| | | | |
|---|---|---|---|
| Postmortem | Ad-hoc | Times afetados | RCA, ações corretivas e aprendizado |
| Cost Optimization | Mensal | SRE, FinOps | Revisão de indexação, Flex Logs, cardinalidade |
| Roadmap Técnico | Trimestral | Liderança TI | Alinhamento de evolução da plataforma |

### 7.6 Próximos passos e aprovação

- Aprovação de escopo (fontes, retenção, RBAC e requisitos LGPD).
- Designação de equipe (SRE/DevOps/Backend/Segurança/Networking/Intune).
- Acesso a ambientes Azure e Datadog provisionado.
- App registration/identidade com permissões Graph e segregação por ambientes.
- Definição do piloto (10% $\rightarrow$ 50% $\rightarrow$ 100%)*ecritriosderollback*.

## 8. Riscos Técnicos e Mitigações

| Risco | Impacto | Mitigação / Fallback |
|---|---|---|
| **Expiração do deltaLink / falha de Delta Query** | Lacuna em inventário/métricas; SLIs degradados por dados desatualizados. | Tratar `HTTP 410 Gone` / `syncStateNotFound` com **fallback automático** para full sync; respeitar `Retry-After` e backoff exponencial para `429`; SLI de *freshness* e monitor de *last success*. |
| **Expiração/rollover de certificado SCEP** | Quebra do stitching (perda de `IntuneDeviceId` no SAN); queda de `join_confidence:high`. | Monitorar **cobertura de correlação** (% de sessões VPN com `device.id`) e alertar se $< 95\%$; testar **SCEP rollover** no piloto (Chaos); manter correlação probabilística como último recurso e marcar `join_confidence:low`. |
| **Falha de correlação SCEP (template/SAN/parsers)** | Joins incorretos ou ausentes (VPN $\leftrightarrow$ device); diagnósticos ambíguos. | Validar template SCEP e SAN obrigatórios; testes automatizados de parsing do SAN; dashboards/monitores por `join_confidence`; runbook para rotação/expiração e drift de formato. |

| | | |
|---|---|---|
| **Limite de throughput do Event Hubs** | Backlog/latência; risco de rejeição em picos ("tempestade" de logins). | Dimensionar partições para pico (regra de 1 MB/s por partição); habilitar **Auto-inflate** (Standard); isolar consumidores via **consumer groups**; alertas de backlog/lag e escala via KEDA. |
| **Custos imprevisíveis de logs Cisco** | Estouro de orçamento (indexação/armazenamento); ruído operacional. | Index exclusion filters para logs verbosos/baixo valor; **Flex Logs** para retenção barata; **logs-to-metrics** (Generate Metrics) para tendências e descarte do log bruto quando possível; guardrails de cardinalidade. |
| **Dependência do formato de syslog ASA** | Quebra de parsers (Grok) após upgrade; perda de campos críticos (reason/device id). | Preferir CEF quando disponível; monitorar *parsing failures* (tags/erros); versionar parsers e validar em *staging* antes de firmware upgrades. |
| **Inflação de falsos positivos (Doze Mode / One UI)** | Alertas indevidos; erosão de confiança no monitoramento. | Estado tri-valorado; evidência cruzada (Entra/VPN/Salesforce); mitigação Samsung via KSP/OEMConfig; janelas adaptativas por perfil de uso; monitores de anomalia por percentual de frota. |
| **Vazamento de PII** | Risco legal e reputacional (LGPD). | Sensitive Data Scanner na entrada; hashing/redaction/masking; RBAC e audit trail; segregação de índices/retention; auditoria automatizada (*potential_violations*) e testes *canary*. |
| **Dependência excessiva de APIs externas** | Degradação por throttling/outage (Graph/Salesforce). | Cache/estado; filas + retry; *graceful degradation* (Unknown); limites de taxa; alarmes de *freshness* e fallback documentado. |

## 9. Governança de dados e tags (FinOps sem estimativas)

Nesta proposta, "FinOps" é tratado como **governança de volume, retenção, indexação e cardinalidade** (sem detalhamento de custos), visando manter a observabilidade sustentável e previsível.

### Política de tags e cardinalidade (guardrails)

**Princípios:**

- **Tags estáveis e de baixa cardinalidade** para filtros e rateio lógico (ex.: `service`, `source`, `region`, `business_unit`, `env`).
- **Evitar tags por dispositivo/usuário** (ex.: `device_id`, UPN) quando houver risco de alta cardinalidade; preferir atributos consultáveis/fields e agregações.
- **Taxonomia padronizada** e versionada (dicionário de dados) para evitar drift entre pipelines.

### Indexação, retenção e logs-to-metrics (por criticidade)

- Classificar sinais em **críticos** (investigação forense/segurança) vs. **operacionais volumétricos** (sucessos rotineiros), com retenções e indexação distintas.
- Aplicar **logs-to-metrics** para SLIs/SLOs (agregações) quando o objetivo for tendência/saúde, mantendo logs detalhados apenas quando indispensável.
- Definir **limites e alarmes** de volume e de cardinalidade por fonte para prevenir degradação (e.g., novos campos/tags inesperados).

## 10. Indicadores de Sucesso (KPIs e SLOs)

Esta seção consolida os KPIs e SLOs do serviço, incluindo definição operacional e critério de medição.

| **Indicador** | **Meta** | **Como medir** |
|---|---|---|
| MTTR (incidentes mobilidade) | $-60\%$ | Tempo entre detecção (monitor) e mitigação/fechamento (ITSM) por severidade. |
| Redução de tickets | $-40\%$ | Volume mensal (categoria mobilidade/VPN/Salesforce) normalizado por base ativa. |
| Cobertura de correlação de identidade | $> 95\%$ | Percentual de eventos Cisco/Intune/Entra com `IntuneDeviceId` correlacionado (SCEP stitching). |
| Atingimento de SLO (VPN) | $\geq 99.0\%$ | SLO Datadog mensal com exclusões documentadas. |
| Error Budget saudável | $> 50\%$ restante (meio do mês) | Consumo proporcional (burn rate) vs. budget esperado para período. |

# 11. Runbooks Operacionais (Referência)

### Runbook R1: Delta Sync (Graph) falhando

**Sintoma:** queda de freshness do inventário, falhas recorrentes em execução de Function.

**Checklist:**

1. Verificar erros `HTTP 410 Gone` / `ResourceNotFound` (delta expirado).
2. Confirmar saúde de autenticação (Managed Identity/credenciais) e throttling (429).
3. Acionar **fallback full sync** e reset do estado (Table Storage), preservando evidência.
4. Após recuperação, validar taxa de processamento e atualizar dashboard "Delta Sync Health".

### Runbook R2: Vazamento potencial de PII em logs

**Sintoma:** alerta do scanner de dados sensíveis.

**Resposta:**

1. Bloquear/limitar acesso (RBAC) ao índice afetado.
2. Aplicar redaction imediata no pipeline e reprocessar/restringir retenção conforme política.
3. Abrir incidente de segurança e registrar trilha de auditoria (quem acessou / exportou).
4. Ajustar padrões (regex) e testes automatizados com payloads sintéticos.

### Runbook R3: Instabilidade regional VPN (correlacionada)

**Sintoma:** aumento anômalo em falhas por região/operadora.

**Resposta:**

1. Confirmar que motivos excluídos (wrong_password, etc.) não dominam o denominador.
2. Identificar concentrator(s) e ISP(s) com maior contribuição.
3. Se fast burn: comunicar stakeholder(s), escalar NOC/Networking e acionar mitigação (rota alternativa/capacidade).
4. Encerrar com *postmortem* e ação preventiva (capacity, vendor escalation, tuning de alertas).

# 12. Evolução e Roadmap Futuro

- **Device Risk Scoring**: score de risco por device (comportamento, compliance, falhas recorrentes).

- **Análise preditiva**: detecção antecipada de falhas (séries temporais, anomalias, forecast de saturação).
- **Integração com Mobile Threat Defense (MTD)**: enriquecer sinais de segurança e resposta automatizada.
- **Expansão para iOS/macOS**: generalização do modelo SRE para outras plataformas.
- **Observabilidade corporativa unificada**: padronização de tagging, SLOs e governança para outros domínios (rede, apps, identidade).

## Referências

## Referências


[1] ANDROID DEVELOPERS. *Android Enterprise documentation*. Disponível em: https://developer.android.com/work. Acesso em: 05 fev. 2026.

[2] CISCO SYSTEMS. *Cisco Secure Firewall ASA Series Syslog Messages Guide*. Disponível em: https://www.cisco.com/c/en/us/td/docs/security/asa/syslog/asa-syslog.html. Acesso em: 05 fev. 2026.

[3] CISCO SYSTEMS. *Configure ASA Syslog message forwarding*. Disponível em: https://www.cisco.com/c/en/us/support/docs/security/pix-500-series-security-appliances/63884-config-asa-00.html. Acesso em: 05 fev. 2026.

[4] DATADOG, INC. *Azure Event Hub integration*. Disponível em: https://docs.datadoghq.com/integrations/azure-event-hub/. Acesso em: 05 fev. 2026.

[5] DATADOG, INC. *Log collection and configuration*. Disponível em: https://docs.datadoghq.com/logs/log_collection/. Acesso em: 05 fev. 2026.

[6] DATADOG, INC. *Service Level Objectives (SLOs)*. Disponível em: https://docs.datadoghq.com/service_management/service_level_objectives/. Acesso em: 05 fev. 2026.

[7] BEYER, B. et al. *Site Reliability Engineering*. Sebastopol: O'Reilly Media, 2016.

[8] BEYER, B. et al. *The Site Reliability Workbook*. Sebastopol: O'Reilly Media, 2018.

[9] KIM, G. et al. *The DevOps Handbook*. Portland: IT Revolution, 2016.

[10] MICROSOFT. *Microsoft Intune – Send logs to Azure Monitor*. Disponível em: https://learn.microsoft.com/intune/intune-service/fundamentals/review-logs-using-azure-monitor. Acesso em: 05 fev. 2026.

[11] MICROSOFT. *Microsoft Graph REST API v1.0 – managedDevice resource*. Disponível em: https://learn.microsoft.com/graph/api/resources/intune-devices-manageddevice. Acesso em: 05 fev. 2026.

[12] MICROSOFT. *Microsoft Entra ID – Diagnostic settings and sign-in logs*. Disponível em: https://learn.microsoft.com/entra/identity/monitoring-health/. Acesso em: 05 fev. 2026.

[13] MICROSOFT. *Azure Event Hubs documentation*. Disponível em: https://learn.microsoft.com/azure/event-hubs/. Acesso em: 05 fev. 2026.

[14] MICROSOFT. *Azure Well-Architected Framework*. Disponível em: https://learn.microsoft.com/azure/architecture/framework/. Acesso em: 05 fev. 2026.