Plano de Monitoramento – Bot Cripto Telegram

# 1. Introdução

Este Plano de Monitoramento define métricas, alertas e rotinas de acompanhamento para garantir o bom funcionamento do Bot Cripto Telegram. Ele complementa o Runbook Operacional, focando no que deve ser observado continuamente em produção.

# 2. Métricas Principais

- \*\*Tempo de Resposta\*\*: tempo médio de execução de workflows (<3s alvo).  
- \*\*Latência de Alertas\*\*: diferença entre condição de preço e envio da notificação (<60s alvo).  
- \*\*Taxa de Falhas\*\*: % de execuções com erro em workflows (WF\_Error\_Handler).  
- \*\*Taxa de Acertos da IA\*\*: proporção de respostas corretas/úteis do agente IA.  
- \*\*Uso de APIs\*\*: número de chamadas Binance, OpenAI, RSS por período (diário/horário).  
- \*\*Cache Hit Ratio (Redis)\*\*: % de requisições respondidas pelo cache em vez de API externa.

# 3. Alertas Automáticos

- \*\*Uptime Webhook Telegram\*\*: alerta se o endpoint estiver indisponível >1 min.  
- \*\*Taxa de Erros Alta\*\*: alerta se >5% das execuções falharem em 10 min.  
- \*\*Redis Inacessível\*\*: alerta se não houver resposta do cache em 2 tentativas seguidas.  
- \*\*Supabase/Postgres Inacessível\*\*: alerta se consultas falharem por mais de 5 min.  
- \*\*Consumo de API Excedido\*\*: alerta se chamadas acumuladas > limite diário definido.  
- \*\*Custos Elevados\*\*: alerta se custo estimado do OpenAI ultrapassar orçamento diário.

# 4. Ferramentas de Monitoramento

- \*\*UptimeRobot\*\* ou Healthchecks.io: monitoramento do webhook principal (WF\_Bot\_Principal).  
- \*\*Logs do Supabase\*\*: tabelas BotLog e ErrorState para auditoria de falhas e execuções.  
- \*\*N8N Execution List\*\*: painel nativo para identificar workflows falhos.  
- \*\*Redis Monitor (opcional)\*\*: redis-cli monitor ou dashboard Upstash.  
- \*\*Grafana/Prometheus (opcional)\*\*: centralização de métricas avançadas.

# 5. Periodicidade de Checagem

- \*\*A cada 1 min\*\*: uptime do webhook (via UptimeRobot).  
- \*\*A cada 5 min\*\*: disponibilidade Redis e Supabase.  
- \*\*A cada hora\*\*: consumo de APIs (tabela ApiUsage).  
- \*\*Diário\*\*: revisão de número de alertas disparados e consumo OpenAI.  
- \*\*Semanal\*\*: análise dos erros registrados em ErrorState, ajuste de prompts IA.  
- \*\*Mensal\*\*: auditoria completa de logs (BotLog) e verificação de custos.

# 6. Ações Corretivas

- \*\*Workflow travado\*\* → reiniciar workflow no painel N8N.  
- \*\*API rate limited\*\* → aumentar TTL do cache Redis temporariamente.  
- \*\*Redis offline\*\* → fallback para Supabase PriceCache até Redis voltar.  
- \*\*Erros repetidos da IA\*\* → revisar prompts de sistema, ajustar fluxos.  
- \*\*Supabase indisponível\*\* → pausar alertas até banco voltar, usar cache temporário.

# 7. Conclusão

Com este plano de monitoramento, o operador do Bot Cripto Telegram terá visibilidade contínua do sistema, acionando medidas corretivas rápidas sempre que forem detectados problemas. Isso garante estabilidade e confiança para o uso diário.