Runbook Operacional – Bot Cripto Telegram

# 1. Introdução

Este Runbook Operacional descreve os procedimentos para testes, implantação, monitoramento e manutenção do Bot Cripto Telegram. O objetivo é garantir que o sistema esteja sempre disponível, seguro e confiável.

# 2. Testes e Garantia de Qualidade

Antes da implantação final, o sistema deve ser submetido a testes técnicos e de aceitação:

- \*\*Tempo de Resposta (<3s):\*\* medir início e fim de cada workflow no N8N.  
- \*\*Latência de Alerta (<60s):\*\* criar alerta de teste próximo ao preço atual e medir tempo de disparo.  
- \*\*Uptime (99%):\*\* monitorar endpoint do webhook principal via UptimeRobot ou similar.  
- \*\*Zero Falsos Alertas:\*\* rodar soak tests com alertas acima/abaixo do preço atual.  
- \*\*Testes de Aceitação:\*\* validar todos os comandos (/preco, /alerta, /comprar, /vender, /posicao, /pnl, /news).

# 3. Procedimentos de Implantação

Checklist para colocar o bot em produção:

1. \*\*Configuração do Ambiente\*\*  
 - Instância N8N ativa (Docker/Hostinger/Portainer).  
 - Projeto Supabase criado, esquema Prisma aplicado.  
 - Redis disponível (Upstash ou self-hosted).  
  
2. \*\*Configuração de Credenciais\*\*  
 - Todas as chaves armazenadas no N8N Vault (Telegram, Binance, OpenAI, Supabase, etc.).  
 - Configuração de dupla credencial: Supabase API + Postgres pooler.  
 - Variáveis de ambiente para Redis (URL, token).  
  
3. \*\*Segurança\*\*  
 - Webhook do Telegram protegido com token secreto.  
 - Sanitização de inputs do usuário antes de queries SQL ou chamadas HTTP.  
  
4. \*\*Ativação de Workflows\*\*  
 - Ativar WF\_Error\_Handler e testar notificação.  
 - Ativar WF\_Monitor\_Alertas (cron).  
 - Ativar WF\_Bot\_Principal e registrar webhook no Telegram.  
  
5. \*\*Testes Finais (Go/No-Go)\*\*  
 - Executar casos de teste da seção 2.  
 - Validar logs de inicialização no Supabase (BotLog, ErrorState).

# 4. Monitoramento e Operação

- \*\*Uptime:\*\* monitorar URL do webhook via UptimeRobot.  
- \*\*Alertas de Erro:\*\* WF\_Error\_Handler envia mensagens no Telegram em caso de falha.  
- \*\*Logs:\*\* revisar periodicamente tabelas BotLog, ApiUsage e ErrorState no Supabase.  
- \*\*APIs:\*\* acompanhar limites de taxa no Binance e no OpenAI.

# 5. Gerenciamento de Riscos

- \*\*Falha de APIs externas:\*\* fallback para cache Redis/Supabase.  
- \*\*Rate limiting:\*\* cache agressivo (TTL curto), monitoramento ApiUsage.  
- \*\*Erros de interpretação da IA:\*\* logs em ErrorState para ajuste de prompts.  
- \*\*Perda de dados:\*\* rely em backups automáticos do Supabase.

# 6. Futuras Melhorias Operacionais

- Monitoramento centralizado via Grafana/Prometheus (conectando Supabase e Redis).  
- Relatórios automáticos de uso enviados diariamente no Telegram.  
- Script de automação de deploy (CI/CD simples).  
- Expansão para múltiplos usuários (já suportado pelo schema).