PDR – Bot Cripto Telegram

# 1. Visão Geral

O projeto consiste em um bot pessoal de Telegram para acompanhamento de criptomoedas.  
Objetivo: prover consultas rápidas de preço, alertas customizados, gerenciamento de portfólio real e notícias do mercado, com interface em linguagem natural mediada por um agente de IA (LangChain + MCP).  
  
Escopo inicial: uso individual, sem foco em escalabilidade massiva, mas com arquitetura limpa que permita crescer se necessário.

# 2. Objetivos do Produto

- Obter preços em tempo real de qualquer token listado na Binance, com variação 24h.  
- Criar e gerenciar alertas de preço personalizados.  
- Registrar trades reais (via input manual ou API key da Binance) e acompanhar posições, custo médio, PnL realizado/não realizado.  
- Fornecer notícias de criptomoedas com curadoria e sentimento de mercado.  
- Permitir interação natural com IA, sem depender apenas de comandos rígidos.  
- Garantir baixo tempo de resposta (<3s por requisição) e alertas rápidos (<60s).

# 3. Público-Alvo

- Usuário único (criador do projeto).  
- Perfil: entusiasta de automação, estudante de IA e cripto.  
- Objetivo: estudo + uso prático no dia a dia.

# 4. Requisitos Funcionais

1. Consulta de Preço  
 - Comando `/preco BTC` ou variações em linguagem natural.  
 - Dados retornados: preço atual em BRL, variação 24h, volume.  
 - Fonte: Binance Spot API.  
 - Cache Redis: 5–10s.  
  
2. Alertas de Preço  
 - Criar, listar, deletar alertas.  
 - Exemplo: `/alerta add BTC abaixo 250000`.  
 - Execução: Cron job a cada 1–2min.  
 - Persistência: tabela alerts (Supabase).  
  
3. Portfólio Real  
 - Registro manual: `/comprar BTC 0.05 a 248000`.  
 - Registro de venda: `/vender BTC 0.02 a 255000`.  
 - Consulta posição: `/posicao`, `/pnl`.  
 - Estrutura de dados: tabela trades (Supabase).  
 - Cálculo: custo médio móvel, PnL realizado/não realizado.  
 - Fase futura: integração via API Key read-only da Binance.  
  
4. Notícias  
 - Comando `/news [N]`.  
 - Fontes: RSS (CoinDesk, CoinTelegraph) + opção futura CryptoCompare.  
 - Cache Redis: 2–5min.  
 - Curadoria: IA resume e classifica sentimento (bullish, bearish, neutro).  
  
5. Interação via IA  
 - Usuário pode escrever em linguagem natural.  
 - Agente IA (LangChain + MCP) decide qual ferramenta usar.  
 - Exemplo: “me avisa se o solana cair abaixo de 150 reais e já mostra o preço agora”.  
  
6. Gestão de Erros  
 - Workflow dedicado (WF\_Error\_Handler).  
 - Registro em logs (Supabase).  
 - Notificação de falhas via Telegram.

# 5. Requisitos Não Funcionais

- Tempo de resposta: <3s por interação.  
- Latência de alertas: <60s.  
- Uptime: 99% (dependente do host do n8n).  
- Segurança:  
 - API keys no Vault do n8n.  
 - Webhook Telegram protegido.  
 - Sanitização de inputs.  
- Armazenamento eficiente:  
 - Redis para cache volátil.  
 - Supabase/Postgres para dados persistentes.

# 6. Arquitetura do Sistema

- Frontend: Telegram Bot.  
- Orquestração: n8n.  
- Persistência: Supabase/Postgres.  
- Cache: Redis.  
- APIs externas:  
 - Binance Spot API (preço, volume).  
 - RSS Feeds (notícias).  
 - Futuro: TAAPI.io (análise técnica), CryptoCompare (notícias avançadas).  
- IA: LangChain + MCP (agente orquestrador).

# 7. Fluxo de Dados (exemplo: preço do BTC)

1. Usuário envia `/preco BTC` no Telegram.  
2. n8n (Telegram Trigger) → mensagem vai pro Agente IA.  
3. Agente identifica intenção: get\_price(BTC).  
4. Workflow checa Redis (px:spot:BTCBRL).  
5. Se cache expirado → chama Binance API → atualiza Redis.  
6. Resposta formatada → Telegram.

# 8. Futuras Extensões

- Análise técnica (RSI, MACD) via TAAPI.io.  
- Integração direta com carteira Binance (API Key read-only).  
- Dashboard web para visualização do portfólio.  
- Suporte multiusuário.  
- Backtesting de estratégias.