

# Prompt para Projeto de Análise de Gráficos de Trading com IA

## Contexto e Objetivo

Você é um especialista em desenvolvimento de aplicações financeiras e análise técnica de trading. Seu objetivo é criar uma aplicação web completa para análise profissional de gráficos de trading que utiliza inteligência artificial para fornecer análises precisas e recomendações de investimento.

## Especificações do Projeto

### 1. Funcionalidades Principais

#### Análise de Gráficos com IA

- Implementar análise automatizada de gráficos de ativos financeiros usando IA
- Identificar padrões técnicos clássicos como: martelo, doji, engolfo, triângulos, suporte/resistência, rompimentos, bandeiras, cunhas
- Detectar tendências de alta, baixa e lateralização
- Calcular e interpretar indicadores técnicos: RSI, MACD, Médias Móveis, Bollinger Bands, Estocástico, Williams %R, ADX
- Fornecer análise de volume e momentum
- Gerar sinais de compra, venda ou espera com base na análise técnica

#### Sistema de Recomendações

- Análise deve ser realizada como se fosse um trader experiente e bem-sucedido
- Fornecer previsões de alta ou baixa com percentual de confiança
- Explicar o raciocínio por trás de cada recomendação
- Incluir pontos de entrada, stop loss e take profit sugeridos
- Alertas automáticos para oportunidades de trading

#### Interface de Usuário

- Dashboard moderno, profissional e responsivo
- Compatibilidade total com desktop e mobile
- Animações fluidas e transições suaves

- Gráficos interativos com ferramentas de desenho (linhas de tendência, suporte/resistência)
- Seleção de ativos: ações brasileiras, ETFs, FIIs, criptomoedas, forex
- Múltiplos tipos de visualização: candlestick, linha, área, OHLC
- Timeframes variados: 1min, 5min, 15min, 1h, 4h, 1d, 1w, 1m

## 2. Arquitetura Técnica

### Backend

- **Framework:** Flask (Python) ou Node.js/Express
- **IA/ML:**
  - OpenAI GPT-4o mini para análise de texto (\$0.15/1M tokens input)
  - Bibliotecas Python: pandas, numpy, ta-lib, scikit-learn
  - Processamento de imagens de gráficos com OpenCV
- **APIs de Dados:**
  - **Gratuita:** brapi.dev (15.000 req/mês, dados BR)
  - **Paga:** Polygon.io (\$29/mês para dados internacionais)
  - **Backup:** Alpha Vantage, Yahoo Finance
- **Banco de Dados:** PostgreSQL ou MongoDB
- **Cache:** Redis para otimização

### Frontend

- **Framework:** React.js com TypeScript
- **Gráficos:** TradingView Charting Library ou Chart.js
- **UI/UX:** Material-UI ou Tailwind CSS
- **Estado:** Redux ou Zustand
- **Animações:** Framer Motion

### Infraestrutura

- **Hospedagem:** Vercel (frontend) + Railway/Heroku (backend)
- **CDN:** Cloudflare para performance
- **Monitoramento:** Sentry para logs de erro

## 3. Estratégia de Custos Otimizada

### Opções Gratuitas (Recomendadas para MVP)

- **Dados:** brapi.dev (gratuito, 15k req/mês)
- **IA:** Ollama com modelos locais (Llama 3.1, Mistral)
- **Hospedagem:** Vercel (frontend) + Railway (backend) - planos gratuitos

- **Banco:** PostgreSQL gratuito (Supabase/Neon)

## Opções Pagas (Para Escala)

- **Dados Premium:**
- brapi.dev Startup: R\$ 599,90/mês (150k req/mês)
- Polygon.io Starter: \$29/mês (dados internacionais)
- **IA Premium:** OpenAI GPT-4o mini: ~\$10-50/mês (uso moderado)
- **Hospedagem Premium:** \$20-100/mês dependendo do tráfego

## 4. Implementação da Análise com IA

### Processamento de Dados

```
# Exemplo de estrutura para análise
def analisar_ativo(symbol, timeframe):
    # 1. Obter dados históricos
    dados = obter_dados_historicos(symbol, timeframe)

    # 2. Calcular indicadores técnicos
    indicadores = calcular_indicadores(dados)

    # 3. Identificar padrões
    padroes = identificar_padroes_candlestick(dados)

    # 4. Análise com IA
    contexto = preparar_contexto_ia(dados, indicadores, padroes)
    analise_ia = chamar_ia_analise(contexto)

    # 5. Gerar recomendação
    return gerar_recomendacao(analise_ia, indicadores)
```

### Prompt para IA de Análise

Você é um trader profissional com 20 anos de experiência e histórico de sucesso no mercado financeiro. Analise os seguintes dados técnicos:

DADOS DO ATIVO: {symbol}  
- Preço atual: {preco\_atual}  
- Variação 24h: {variacao\_24h}  
- Volume: {volume}

INDICADORES TÉCNICOS:  
- RSI(14): {rsi}  
- MACD: {macd}  
- Médias Móveis: {mas}  
- Bollinger Bands: {bb}

PADRÕES IDENTIFICADOS: {padroes}

HISTÓRICO RECENTE: {dados\_historicos}

Forneça uma análise completa incluindo:

1. Tendência atual (alta/baixa/lateral)
2. Força da tendência (1-10)
3. Padrões técnicos relevantes
4. Níveis de suporte e resistência
5. Recomendação: COMPRA/VENDA/ESPERA
6. Confiança da análise (%)
7. Pontos de entrada, stop loss e take profit
8. Justificativa técnica detalhada

## 5. Funcionalidades Avançadas

### Análise Multi-Timeframe

- Correlação entre diferentes timeframes
- Confirmação de sinais em múltiplos períodos
- Análise de divergências

### Portfolio Tracking

- Acompanhamento de carteira
- Cálculo de performance
- Relatórios de P&L

### Alertas Inteligentes

- Notificações push/email
- Alertas baseados em padrões
- Monitoramento 24/7

### Backtesting

- Teste de estratégias históricas
- Métricas de performance
- Otimização de parâmetros

## 6. Cronograma de Desenvolvimento

### Fase 1 (2-3 semanas): MVP

- Setup básico do projeto

- Integração com APIs gratuitas
- Interface básica de gráficos
- Análise simples com IA

## **Fase 2 (2-3 semanas): Funcionalidades Core**

- Indicadores técnicos completos
- Identificação de padrões
- Sistema de recomendações
- Interface responsiva

## **Fase 3 (2-3 semanas): Funcionalidades Avançadas**

- Multi-timeframe
- Alertas
- Portfolio tracking
- Otimizações de performance

## **Fase 4 (1-2 semanas): Deploy e Testes**

- Deploy em produção
- Testes de carga
- Ajustes finais
- Documentação

# **7. Considerações de Segurança**

## **Proteção de APIs**

- Rate limiting
- Autenticação JWT
- Criptografia de dados sensíveis
- Validação de inputs

## **Compliance**

- Disclaimers legais sobre investimentos
- Termos de uso claros
- Política de privacidade
- Não constituir aconselhamento financeiro

## 8. Métricas de Sucesso

### Técnicas

- Latência < 2 segundos para análises
- Uptime > 99.5%
- Precisão de análises > 70%

### Negócio

- Taxa de retenção de usuários
- Engagement com recomendações
- Feedback qualitativo dos usuários

## Instruções de Execução

1. **Comece pelo MVP:** Implemente primeiro as funcionalidades essenciais usando apenas recursos gratuitos
2. **Teste extensivamente:** Valide cada componente antes de avançar
3. **Otimize custos:** Monitore uso de APIs e IA para manter dentro do orçamento
4. **Documente tudo:** Mantenha documentação técnica atualizada
5. **Feedback contínuo:** Colete feedback de usuários para melhorias

## Recursos Adicionais

### Bibliotecas Python Recomendadas

- `yfinance` : Dados gratuitos do Yahoo Finance
- `ta-lib` : Indicadores técnicos
- `pandas-ta` : Análise técnica com pandas
- `mplfinance` : Visualização de gráficos financeiros
- `ccxt` : APIs de exchanges de criptomoedas

### Ferramentas de Desenvolvimento

- **IDE:** VS Code com extensões Python/React
- **Versionamento:** Git + GitHub
- **CI/CD:** GitHub Actions
- **Monitoramento:** Sentry + LogRocket

## Referências Técnicas

- Documentação TradingView Charting Library
- Documentação APIs: brapi.dev, Polygon.io, OpenAI
- Livros: "Technical Analysis of the Financial Markets" - John Murphy

---

**Importante:** Este projeto deve ser desenvolvido com foco na experiência do usuário e precisão das análises. Sempre inclua disclaimers apropriados sobre riscos de investimento e não constitua aconselhamento financeiro personalizado.