# Contexto

O setor agrícola brasileiro representa um dos pilares fundamentais da economia nacional, movimentando aproximadamente R\$ 700 bilhões anuais em crédito rural e sendo responsável por mais de 25% do PIB do país. Pequenos e médios produtores rurais, que compõem a maioria dos estabelecimentos agropecuários brasileiros, enfrentam sistematicamente o desafio do descasamento temporal entre investimentos necessários para o plantio e o retorno financeiro da comercialização da safra. Durante os 6 a 12 meses que compreendem um ciclo produtivo típico, estes produtores necessitam de capital para aquisição de insumos essenciais como sementes certificadas, defensivos agrícolas, fertilizantes e eventualmente locação de maquinário, mas só conseguem gerar receita após a colheita e venda dos produtos.

O sistema bancário tradicional, embora ofereça linhas de crédito rural, apresenta limitações significativas que dificultam o acesso ao capital por parte destes produtores. As taxas de juros praticadas pelas instituições financeiras convencionais variam entre 18% e 25% ao ano para este segmento, valores considerados elevados quando comparados à rentabilidade típica da atividade agrícola. Além disso, as exigências de garantias são complexas e muitas vezes inacessíveis para pequenos produtores, incluindo avaliações de propriedades, seguros rurais e comprovações burocráticas extensas que podem levar meses para serem processadas. Este cenário cria um gargalo estrutural que limita a capacidade de expansão e modernização da agricultura familiar e de médio porte.

Paralelamente, o mercado financeiro brasileiro tem testemunhado o crescimento exponencial de investidores pessoa física em busca de alternativas de investimento que ofereçam retornos superiores aos produtos tradicionais de renda fixa. Investidores têm buscado diversificar suas carteiras com produtos alternativos que possam oferecer rendimentos ainda mais atrativos e diferenciação de risco, incluindo fundos imobiliários, criptomoedas e investimentos em fintechs. Simultaneamente, o conceito de empréstimos peer-to-peer tem ganhado tração global, permitindo que pessoas físicas emprestem diretamente para outras pessoas ou empresas, eliminando intermediários bancários e oferecendo melhores condições tanto para tomadores quanto para investidores.

A convergência de tecnologias emergentes como blockchain e machine learning tem criado novas possibilidades para mitigação de riscos e automação de processos financeiros no setor agrícola. Smart contracts baseados em blockchain podem automatizar liberações de pagamento com base em critérios pré-estabelecidos, garantindo transparência total das operações, pois todas as transações e condições ficam registradas de forma imutável. Além disso, os dados de crédito podem ser avaliados considerando a reputação de cada produtor, permitindo análises de histórico de empréstimos, adimplência e cumprimento de pagamentos. À medida que a reputação do produtor cresce com pagamentos realizados corretamente, seus limites de crédito disponíveis podem aumentar, criando um sistema dinâmico de confiança.

Com base nessa reputação, também é possível classificar o risco para o investidor que está disponibilizando o capital, fornecendo uma visão clara do perfil de risco associado a cada operação. Algoritmos de inteligência artificial podem combinar dados históricos de safras, preços de commodities e comportamento de crédito para ajustar automaticamente a avaliação de risco e propor condições de empréstimo mais seguras e personalizadas. Sistemas de análise antifraude podem validar informações de

produtores através de bases de dados públicas como INCRA e Receita Federal, oferecendo camadas adicionais de segurança e transparência para investidores.

# Proposta: Marketplace de Crédito Agrícola P2P (agricultores ↔ investidores)

Plataforma marketplace que conecta produtores rurais (tomadores) a investidores (provedores de capital). Produtores tokenizam parte da colheita como garantia (recibos, notas de armazenagem, NFTs) e solicitam empréstimos para financiar insumos até a venda da safra. Investidores avaliam propostas com métricas de risco (score on-chain/off-chain, preços de commodities via oráculos) e decidem financiar total ou parcialmente operações.

## Objetivos principais

- Reduzir o custo do crédito para pequenos e médios produtores rurais.
- Conectar diretamente produtores e investidores, eliminando intermediários.
- Oferecer alternativas de investimento com rendimento ajustado ao risco.
- Garantir transparência e imutabilidade das operações via blockchain.
- Automatizar pagamentos, liquidações e análises de risco com smart contracts e oráculos.
- Integrar meios de pagamento (Pix, gateways, stablecoins) para flexibilidade.

## 3. Estrutura de Banco de Dados

Diagrama de Entidade-Relacionamento (DER)

```
erDiagram
        PRODUTORES {
                UUID produtor_id PK
                VARCHAR nome
                VARCHAR cpf_cnpj
                VARCHAR endereco_wallet
                VARCHAR regiao
                TIMESTAMP criado_em
                TIMESTAMP atualizado_em
        }
        INVESTIDORES {
                UUID investidor_id PK
                VARCHAR nome
                VARCHAR email
                VARCHAR endereco_wallet
                TIMESTAMP criado_em
                TIMESTAMP atualizado_em
        PROPOSTAS {
                UUID proposta_id PK
                UUID produtor_id FK
                DECIMAL valor_solicitado
                VARCHAR moeda
```

```
DATE data_vencimento
        VARCHAR commodity
        VARCHAR status
        TEXT termos_ipfs_hash
        TIMESTAMP criado_em
        TIMESTAMP atualizado_em
}
EMPRESTIMOS {
        UUID emprestimo_id PK
        UUID proposta_id FK
        UUID investidor_id FK
        DECIMAL valor_financiado
        DECIMAL valor_outstanding
        VARCHAR status
        TIMESTAMP financiado_em
        TIMESTAMP atualizado_em
GARANTIAS_TOKEN {
        UUID garantia_id PK
        UUID proposta_id FK
        UUID produtor_id FK
        VARCHAR token_uri
        VARCHAR tipo_token
        DECIMAL quantidade
        VARCHAR unidade
        VARCHAR informacao_armazenagem
        TIMESTAMP criado_em
        TIMESTAMP atualizado_em
}
REEMBOLSOS {
       UUID reembolso_id PK
        UUID emprestimo_id FK
        DECIMAL valor
        TIMESTAMP pago_em
        VARCHAR metodo
        VARCHAR tx_hash
}
PONTUACOES {
        UUID pontuacao_id PK
        UUID produtor_id FK
        INT valor_pontuacao
        DECIMAL pd
        VARCHAR versao_modelo
        TEXT snapshot_features_ipfs
        TIMESTAMP calculado_em
}
TRANSACOES {
        UUID transacao_id PK
        UUID emprestimo_id FK
        VARCHAR from_wallet
        VARCHAR to_wallet
        DECIMAL valor
        VARCHAR moeda
```

```
VARCHAR status
VARCHAR tipo
TIMESTAMP criado_em
}

PRODUTORES ||--o{ PROPOSTAS : cria
PRODUTORES ||--o{ GARANTIAS_TOKEN : possui
PROPOSTAS ||--o{ EMPRESTIMOS : gera
INVESTIDORES ||--o{ EMPRESTIMOS : financia
EMPRESTIMOS ||--o{ REEMBOLSOS : inclui
PRODUTORES ||--o{ PONTUACOES : tem
EMPRESTIMOS ||--o{ TRANSACOES : registra
```

## Descrição das Entidades

### **PRODUTORES**

- Armazena informações dos produtores (tomadores de empréstimo).
  - produtor\_id (PK): Identificador único do produtor (UUID).
  - nome: Nome ou razão social do produtor (VARCHAR).
  - cpf cnpj: Documento fiscal (CPF ou CNPJ) (VARCHAR).
  - endereco\_wallet: Endereço de carteira on-chain vinculado ao produtor (VARCHAR).
  - regiao: Município / estado da produção (VARCHAR).
  - criado\_em / atualizado\_em: Data e hora de criação e atualização do registro (TIMESTAMP).

#### **INVESTIDORES**

- Dados dos investidores que fornecem capital.
  - investidor\_id (PK): Identificador único do investidor (UUID).
  - nome: Nome ou razão social (VARCHAR).
  - email: Contato principal (VARCHAR).
  - endereco\_wallet: Endereço on-chain do investidor (VARCHAR).
  - criado\_em / atualizado\_em: Timestamps (TIMESTAMP).

### **PROPOSTAS**

- Propostas de empréstimo criadas pelos produtores.
  - proposta id (PK): Identificador único da proposta (UUID).
  - produtor\_id (FK): Referência ao produtor que criou a proposta (UUID).
  - valor solicitado: Montante requerido (DECIMAL).
  - moeda: Moeda do empréstimo (ex.: BRL, USDT, USDC) (VARCHAR).
  - data\_vencimento: Data prevista para liquidação (DATE).
  - commodity: Produto agrícola que lastreia a proposta (ex.: soja, milho) (VARCHAR).
  - status: Estado da proposta (ABERTA / FINANCIADA / CANCELADA / VENCIDA) (VARCHAR).
  - termos\_ipfs\_hash: Hash ou link no IPFS com termos e documentos (TEXT).
  - criado\_em / atualizado\_em: Timestamps (TIMESTAMP).

#### **EMPRESTIMOS**

- Empréstimos efetivamente financiados a partir das propostas.
  - emprestimo\_id (PK): Identificador único do empréstimo (UUID).
  - proposta\_id (FK): Referência à proposta base (UUID).
  - investidor id (FK): Investidor que financiou (UUID).
  - valor\_financiado: Valor efetivamente liberado (DECIMAL).
  - valor outstanding: Saldo devedor atual (DECIMAL).
  - status: Estado do empréstimo (ATIVO / LIQUIDADO / INADIMPLENTE) (VARCHAR).
  - financiado em / atualizado em: Timestamps (TIMESTAMP).

## **GARANTIAS\_TOKEN**

- Representação tokenizada das garantias (NFTs, recibos de armazenagem).
  - garantia\_id (PK): Identificador único da garantia (UUID).
  - proposta\_id (FK): Proposta vinculada (UUID).
  - produtor\_id (FK): Dono da garantia (UUID).
  - token\_uri: URI ou hash do token (IPFS) (VARCHAR).
  - tipo token: Tipo do token (ERC721 / ERC1155 / RECEIPT) (VARCHAR).
  - quantidade / unidade: Ex.: 1000 sacas, toneladas (DECIMAL / VARCHAR).
  - informacao\_armazenagem: Local de armazenagem / depósito (VARCHAR).
  - criado\_em / atualizado\_em: Timestamps (TIMESTAMP).

#### REEMBOLSOS

- Registros de pagamentos realizados pelos produtores.
  - reembolso\_id (PK): Identificador do reembolso (UUID).
  - emprestimo\_id (FK): Empréstimo associado (UUID).
  - valor: Quantia paga (DECIMAL).
  - pago\_em: Data do pagamento (TIMESTAMP).
  - metodo: Meio de pagamento (on-chain / off-chain: Pix, boleto) (VARCHAR).
  - tx\_hash: Hash da transação on-chain (se houver) (VARCHAR).

### **PONTUACOES**

- Histórico de pontuações (score) dos produtores.
  - pontuacao\_id (PK): Identificador do registro de pontuação (UUID).
  - produtor id (FK): Produtor avaliado (UUID).
  - valor\_pontuacao: Score numérico (0–1000) (INT).
  - pd: Probabilidade de inadimplência (0–1) (DECIMAL).
  - versao\_modelo: Versão do modelo de cálculo (VARCHAR).
  - snapshot features ipfs: Snapshot das features usado no cálculo (IPFS) (TEXT).
  - calculado\_em: Data do cálculo (TIMESTAMP).

### **TRANSACOES**

• Movimentações financeiras registradas (on-chain ou reconciliadas off-chain).

- transacao\_id (PK): Identificador da transação (UUID).
- emprestimo id (FK): Empréstimo relacionado (opcional) (UUID).
- from\_wallet / to\_wallet: Carteiras de origem e destino (VARCHAR).
- valor: Quantia movimentada (DECIMAL).
- moeda: Tipo de moeda (BRL / USDT / ETH) (VARCHAR).
- status: PENDENTE / CONFIRMADA / FALHA (VARCHAR).
- tipo: LOAN / REEMBOLSO / SEIZURE (VARCHAR).
- criado\_em: Timestamp de criação (TIMESTAMP).

## Considerações de Arquitetura

## Segurança

- Criptografia de dados sensíveis (PII) em repouso e em trânsito.
- Campos de KYC encriptados.
- Uso de vault para gerenciamento seguro de chaves.

#### Consistência

- Transações ACID para operações financeiras críticas.
- Uso de filas (RabbitMQ/Kafka) para conciliação eventual entre eventos on-chain e off-chain.

#### **Escalabilidade**

- Sharding por região ou commodity, se necessário.
- Read-replicas para consultas analíticas.

#### **Auditoria**

• Triggers e tabelas de <u>audit\_log</u> para alterações críticas (mudança de status, drawdown, seize de garantia).

#### Gestão de Risco

• Tabela risk\_profiles consolidando LTV, score, PD e recomendações por empréstimo.

## 4. Estrutura de Front-end

A solução de pagamentos via WhatsApp requer uma interface web para administração e configuração, além de uma integração direta com a API do WhatsApp Business. A estrutura de front-end proposta utiliza tecnologias modernas para garantir uma experiência de usuário otimizada e responsiva.

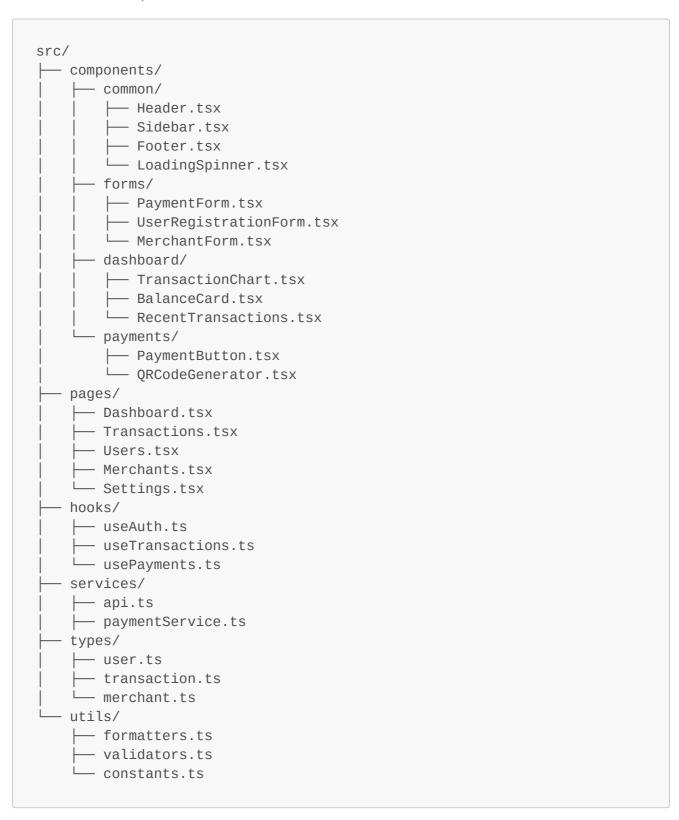
## Arquitetura do Front-end

A aplicação front-end será desenvolvida utilizando React.js com TypeScript, proporcionando uma base sólida para desenvolvimento escalável e manutenível. A arquitetura seguirá o padrão de componentes reutilizáveis e gerenciamento de estado centralizado.

## Tecnologias Principais

- React.js 18+: Framework principal para construção da interface de usuário
- TypeScript: Para tipagem estática e melhor experiência de desenvolvimento
- React Router: Para navegação entre páginas
- Axios: Para comunicação com APIs
- React Hook Form: Para gerenciamento de formulários

## Estrutura de Componentes



## Principais Funcionalidades da Interface

- Dashboard Administrativo: visão geral das transações, usuários ativos e métricas de performance; gráficos interativos (volume, métodos, tendências).
- Gerenciamento de Usuários: visualizar, editar e gerenciar usuários, métodos de pagamento e histórico de transações.
- Painel de Transações: listagem detalhada com filtros por data, status, valor e tipo; busca e exportação.
- Configurações de Integração WhatsApp: gerenciamento de webhooks, tokens e parâmetros da API do WhatsApp Business.

## Integração com WhatsApp

A integração com o WhatsApp será realizada através da WhatsApp Business API, permitindo:

- Envio de mensagens automatizadas: confirmações de pagamento, notificações de saldo.
- Recebimento de comandos: processamento de comandos de pagamento via mensagem.
- Interface de chat: simulação de interface de chat para testes e operação.
- Geração de QR Codes: para pagamentos rápidos via Pix.

## Responsividade e Acessibilidade

A interface será totalmente responsiva, adaptando-se a diferentes tamanhos de tela (desktop, tablet, mobile). Seguirá as diretrizes de acessibilidade WCAG 2.1.

## Segurança no Front-end

- Autenticação JWT: Tokens seguros para autenticação de usuários.
- Validação de entrada: validação rigorosa de todos os dados inseridos no cliente.
- Sanitização: prevenção contra ataques XSS e injeção.
- HTTPS obrigatório: todas as comunicações criptografadas.

## 5. Fluxos e Integrações

- 1. Produtor cria proposal → anexa provas e tokeniza colateral (mint NFT/receipt).
- 2. Proposal fica OPEN e é indexada no marketplace (score, LTV recomendado).
- 3. Investidor analisa e financia total ou parcialmente (funding on-chain / off-chain).
- 4. Quando condições atendidas, drawdown libera fundos ao produtor.
- 5. Produtor realiza repayment parcial/total após venda da safra.
- 6. Em caso de default, procedure de seize é executada conforme termos.

**Integrações principais**: IPFS (documentos), oráculos (preço commodity), provedores de stablecoin / gateways, serviços KYC, INCRA/Receita para verificação.

# 6. Esqueleto de Backend

Stack recomendado: Node.js + TypeScript (Fastify/Express), PostgreSQL (Prisma/TypeORM), Redis, RabbitMQ/Kafka, ethers.js para Web3.

## Endpoints essenciais:

- POST /api/proposals
- GET /api/proposals
- POST /api/proposals/:id/fund
- POST /api/loans/:id/repay
- POST /api/scores/compute

Boas práticas: validação, idempotência, correlação on-chain/off-chain (tx\_hash), testes automatizados.

# 7. Considerações de Implementação

## Infra & serviços:

- API: Node.js 18+ (Fastify/Express)
- DB: PostgreSQL 13+
- Mensageria: RabbitMQ / Kafka
- Storage: IPFS/S3
- Blockchain: Rede EVM (Polygon/Gnosis)
- Oráculos: Chainlink ou custom
- CI/CD: GitHub Actions

## Segurança & compliance:

- Auditoria de smart contracts
- Vault para segredos
- LGPD: consentimento, eliminação, exportação de dados
- KYC e verificação documental (INCRA/Receita onde possível)

### Custos estimados (ex.)

- Infra: R\$ 3.000-6.000 / mês
- IPFS/Storage: R\$ 200-800 / mês
- Monitoramento: R\$ 500-1.500 / mês

## **KPIs sugeridos**

- Produtores cadastrados e investidores ativos
- Volume financiado por mês
- % loans em default
- Latência de funding/drawdown

# Próximos passos práticos

1. Gerar esqueleto do backend (Prisma + Fastify) e migrações do schema.

- 2. Criar smart contract MVP (Proposal/Loan/Collateral) e testes em Hardhat.
- 3. Desenvolver frontend minimal para criar proposals e investir (integração wallet).
- 4. Implementar pipeline inicial de score com dados sintéticos e endpoint de cálculo.

Se quiser, eu posso: gerar a migration SQL/Prisma a partir deste schema, escrever o smart contract MVP em Solidity com testes em Hardhat, ou criar o esqueleto do backend (pronto para rodar). Qual você prefere que eu faça primeiro?