# QuArk

*Conecte capital e demanda em mercados especializados*

## 1. Contexto

No Brasil, o acesso ao crédito para pessoas físicas e PMEs é frequentemente marcado por altas taxas de juros e burocracia nos canais tradicionais. Em paralelo, investidores buscam alternativas de rentabilidade superior aos investimentos conservadores, criando uma demanda clara para modelos que conectem as duas pontas de forma mais eficiente, onde a alta demanda por crédito acessível por parte de pessoas físicas e PMEs converge com o interesse de investidores por maior rentabilidade, fora dos canais tradicionais. Este modelo de negócio é diretamente viabilizado pelo marco regulatório do Banco Central, que instituiu a Sociedade de Empréstimo entre Pessoas (SEP) e forneceu a segurança jurídica necessária.

## 2. Objetivo

O Quark soluciona o problema crônico de acesso a capital entre pessoas físicas e a gestão de risco e facilita o investimento e empréstimo. A missão é criar uma plataforma segura e transparente que conecta diretamente credores e mutuários, de forma dinâmica e facilitada, definindo parâmetros detalhados para suas ofertas de crédito (juros, prazo e risco máximo). A plataforma se destina a Investidores Individuais (pessoa física) e Institucionais (empresas) que buscam melhores retornos em crédito e, do outro lado, pessoas físicas e empresas que necessitam de financiamento rápido e adaptável. A oportunidade reside em descentralizar o processo de crédito, oferecendo aos usuários controle total sobre o capital e o risco.

Assim como o mercado de ações transformou o investimento em empresas ao criar papéis (ações) negociáveis. Em nosso ecossistema, cada pedido de empréstimo é convertido em um "card” ou melhor dizendo um QUARK de investimento padronizado. Isso permite que qualquer pessoa ou empresa invista diretamente em necessidades de crédito a um custo muito baixo, enquanto mutuários recebem acesso a financiamento de forma ágil e justa. Essencialmente, a Quark cria um marketplace onde ofertas de empréstimo são negociadas com a mesma simplicidade e eficiência

## 3. Estrutura de Banco de Dados

Para suportar a solução, foi projetada uma estrutura de banco de dados relacional utilizando PostgreSQL para garantir a integridade transacional (ACID), essencial para operações financeiras. As entidades críticas para a operação inicial (MVP) são:

**Descrição das Entidades**

1.**Tabela USUARIO (USERS)**

- **user_id**: Identificador único do usuário/entidade.
- **tipo_entidade**: Tipo: 'PF' ou 'PJ'. Essencial para regras de negócio e regulamentação.
- **nome_completo**: Nome Civil (PF) ou Razão Social (PJ).
- **nome_fantasia**: Nome fantasia (PJ) ou NULL (PF).
- **email**: Contato único e login.

- **cpf_cnpj_hash**: Hash seguro do CPF ou CNPJ.
- **data_fundacao_nasc**: Data de Nascimento (PF) ou Data de Fundação (PJ).
- **data_cadastro**: Data e hora de criação do usuário.
- **score_credito**: Pontuação atual do risco (Motor de Score, 0-1000).
- **setor_atuacao**: Setor de atuação (Ex: 'Agronegócio', 'Tech'). Usado para elegibilidade de ofertas.
- **regiao**: Localização geográfica. Usado para elegibilidade de ofertas.
- **kyc_status**: Status de validação de identidade ('PENDING', 'VERIFIED', 'FAILED').
- **carteira_id_fk**: Referência à carteira financeira do usuário.

2. **Tabela CARTEIRA (ACCOUNTS)**
   - **carteira_id**: Identificador único da carteira.
   - **user_id_fk**: Referência ao proprietário da carteira.
   - **saldo_disponivel**: Saldo atual que deve ser gerenciado de forma transacional.
   - **status**: Status: 'ativa', 'bloqueada'.

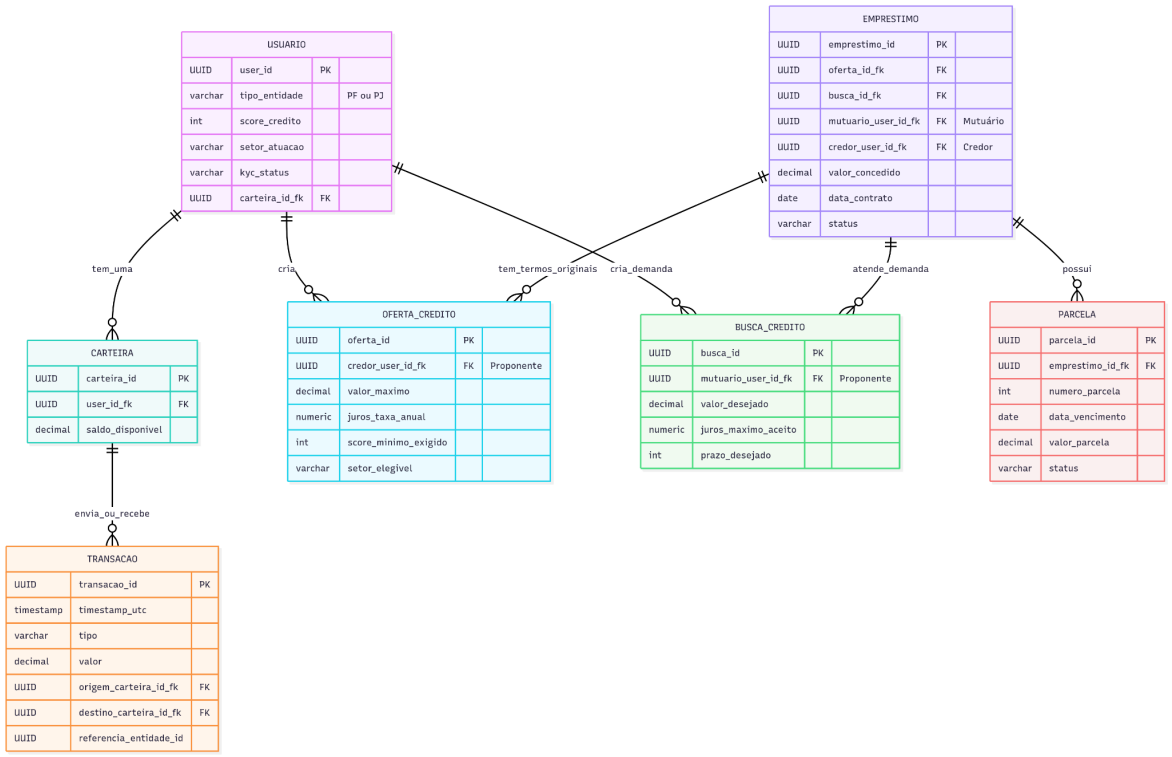
3. **Tabela TRANSACAO (TRANSACTIONS - Ledger Imutável)**
   - **transacao_id**: Identificador único da transação.
   - **timestamp_utc**: Hora exata do evento (auditoria/antifraude).
   - **tipo**: Tipo de movimento: 'P2P_DEBITO', 'EMPRESTIMO_CONCEDIDO', 'PAGAMENTO_PARCELA'.
   - **valor**: Valor movimentado.
   - **origem_carteira_id_fk**: Carteira que enviou os fundos.
   - **destino_carteira_id_fk**: Carteira que recebeu os fundos.
   - **referencia_entidade_id**: Referência a outra entidade (Ex: emprestimo_id ou parcela_id).

4. **Tabela OFERTA_CREDITO (CREDIT_OFFERS)**
   - **oferta_id**: Identificador único da oferta.
   - **credor_user_id_fk**: Usuário Credor que define os termos.
   - **valor_maximo**: Quantia máxima que o Credor está disposto a emprestar.
   - **juros_taxa_anual**: Taxa de juros anual cobrada pelo Credor.
   - **prazo_meses**: Prazo máximo em meses.
   - **score_minimo_exigido**: Score mínimo que o Mutuário deve ter para ver a oferta.
   - **setor_elegivel**: Filtro de nicho do Mutuário (Ex: 'Agronegócio').
   - **data_expiracao**: Data e hora em que a oferta é retirada do mercado.
   - **status**: Status: 'ATIVA', 'PAUSADA', 'COMPROMETIDA'.

5. **Tabela BUSCA_CREDITO (CREDIT_SEARCHES)**
    - **busca_id**: Identificador único da busca.
    - **mutuario_user_id_fk**: Usuário Mutuário que deseja o crédito.
    - **valor_desejado**: Valor que o Mutuário busca pegar emprestado.
    - **juros_maximo_aceito**: Máximo de juros que o Mutuário aceita pagar.
    - **prazo_desejado**: Prazo preferido em meses.
    - **data_expiracao**: Data e hora em que a busca é encerrada.
    - **status**: Status: 'ATIVA', 'NEGOCIANDO', 'CANCELADA'.

6. **Tabela EMPRESTIMO (LOANS)**
    - **emprestimo_id**: Identificador único do contrato de empréstimo.
    - **oferta_id_fk**: Referência à OFERTA_CREDITO que definiu os termos.
    - **busca_id_fk**: Referência à BUSCA_CREDITO que pode ter iniciado o match (opcional).
    - **mutuario_user_id_fk**: Usuário que recebeu o crédito.
    - **credor_user_id_fk**: Usuário que concedeu o crédito.
    - **valor_concedido**: Valor efetivamente transferido.
    - **juros_acordado**: Taxa de juros final.
    - **data_contrato**: Data de início do empréstimo.
    - **status**: 'ATIVO', 'PAGO', 'DEFAULT'.

7. **Tabela PARCELA (INSTALLMENTS)**
    - **parcela_id**: Identificador único da parcela.
    - **emprestimo_id_fk**: Referência ao contrato de empréstimo.
    - **numero_parcela**: Número sequencial da parcela (1, 2, 3...).
    - **data_vencimento**: Data de vencimento prevista.
    - **valor_parcela**: Valor total da parcela (principal + juros).
    - **valor_pago**: Valor efetivamente pago.
    - **data_pagamento**: Data e hora real do pagamento.
    - **status**: 'PENDENTE', 'PAGO', 'ATRASO', 'PARCIAL'.




**Considerações de Arquitetura**

- **Segurança:** Utilização de hash para documentos sensíveis e criptografia de dados em repouso (AES-256) e em trânsito (TLS 1.2+).
- **Consistência:** Forte integridade transacional (ACID) garantida pelo PostgreSQL para operações atômicas de débito e crédito.
- **Auditoria:** O design da tabela TRANSACAO como um ledger imutável atende ao requisito de rastreabilidade completa das operações.

## 4. Estrutura de Front-end

A interface do usuário será desenvolvida como uma Single Page Application (SPA) utilizando React, o que é ideal para garantir uma experiência de usuário rápida e fluida no Marketplace.

**Arquitetura e Tecnologias Principais:**

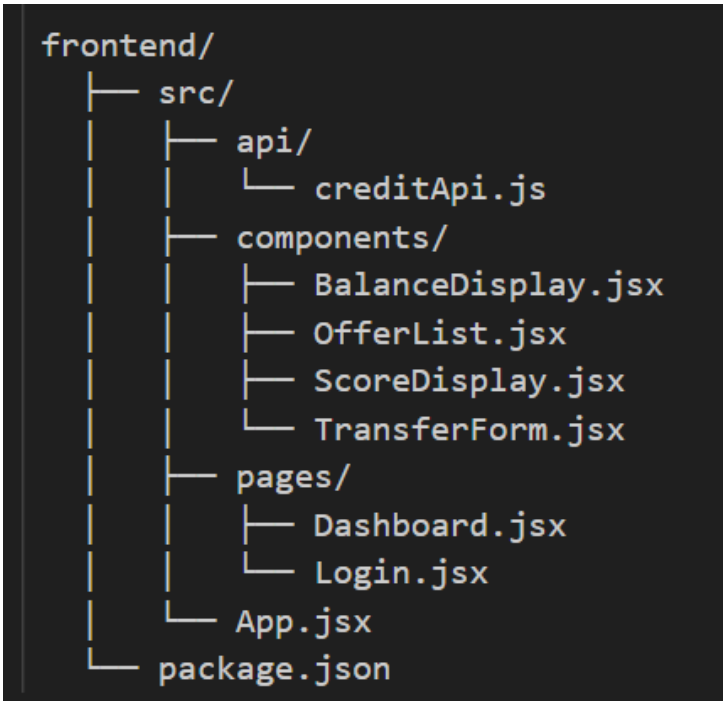
**React:** É a tecnologia principal para a construção da interface do usuário (UI).

**Tailwind CSS:** Framework CSS utilitário recomendado para a estilização rápida e responsiva dos componentes.

**React Scripts:** Gerencia o ambiente de desenvolvimento e o processo de build do projeto.

**Estrutura de Componentes:** A aplicação será organizada de forma modular para facilitar a manutenção e o reuso. A estrutura de arquivos incluirá uma pasta para

componentes reutilizáveis — como um componente para exibir o saldo, um formulário para transferências P2P, um visualizador para o score de crédito e uma lista para as ofertas de crédito. Haverá também uma pasta para as telas principais da aplicação, como o Dashboard e a tela de Login, e um módulo dedicado a centralizar a comunicação com a API.




Exemplo de estrutura a ser usada no MVP

**Principais Funcionalidades da Interface (MVP):**

**Dashboard e Login:** Serão criadas telas básicas para login e um dashboard central. O dashboard permitirá ao usuário visualizar seu saldo, score e as ofertas de crédito para as quais é elegível.

**Visualização de Saldo:** Um componente visual dedicado será responsável por exibir o saldo atualizado do usuário, atendendo ao requisito de visualização em tempo real.

**Transferência P2P:** Um formulário de transferência permitirá que os usuários enviem fundos para outros participantes da plataforma, cumprindo o requisito funcional de processar transações P2P.

**Visualização de Score e Ofertas:** A interface exibirá a pontuação de crédito do usuário e listará as ofertas de crédito compatíveis, atendendo aos requisitos de cálculo de score e filtragem do marketplace.

## 5. Estrutura de Back-end(Python)

A estrutura de back-end será servida por uma API RESTful, projetada para atender às funcionalidades essenciais da plataforma, com endpoints específicos para os serviços de Carteira, P2P, Score e Marketplace.

**Endpoints Essenciais (MVP)**

- **Serviços de Carteira e P2P:**
  - **Consulta de Saldo:** Um endpoint GET permitirá ao usuário autenticado visualizar seu saldo disponível em tempo real.
  - **Transferência P2P:** Um endpoint POST será responsável por processar as transferências de fundos entre usuários. Ele receberá o valor, o destinatário e a senha transacional, validando o saldo antes de iniciar a operação.
- **Serviços de Score e Marketplace:**
  - **Consulta de Score:** Um endpoint GET retornará o score de crédito atualizado do usuário.
  - **Listagem de Ofertas:** Um endpoint GET filtrará e exibirá as ofertas de crédito do marketplace para as quais o usuário é elegível, com base em seu score e setor.

**Endpoints Detalhados**

A API completa suporta funcionalidades adicionais, incluindo a gestão de todo o ciclo de vida de um empréstimo.

- **Serviços de Autenticação e Usuário**
  - `POST /api/v1/auth/register`: Realiza o cadastro inicial do usuário (PF ou PJ).
  - `POST /api/v1/auth/login`: Efetua o login e gera o token de autenticação (JWT).
  - `GET /api/v1/user/profile`: Recupera os dados do perfil do usuário logado.
  - `POST /api/v1/user/kyc/start`: Inicia o processo de validação de identidade (KYC/KYB).
- **Serviços de Carteira e Transações**
  - `GET /api/v1/wallet/balance`: Consulta o saldo disponível da carteira do usuário.
  - `GET /api/v1/transaction/history`: Lista o histórico detalhado de transações do usuário.
- **Serviços de Score e Risco**
  - `GET /api/v1/credit/score/factors`: Retorna os fatores que influenciaram o cálculo do score de crédito do usuário.
- **Serviços do Marketplace de Crédito**
  - `POST /api/v1/marketplace/offers`: Permite que um Credor crie uma nova oferta de crédito.
  - `POST /api/v1/marketplace/accept-offer/{offer_id}`: Permite que um Mutuário aceite uma oferta, disparando a criação do empréstimo.
  - `POST /api/v1/marketplace/searches`: Permite que um Mutuário crie uma busca ativa por crédito.
  - `GET /api/v1/marketplace/matches/{search_id}`: Permite que um Credor consulte as buscas ativas que correspondem aos seus critérios.

- **Serviços de Empréstimos e Parcelas**
    - `GET /api/v1/loan/my-loans`: Lista todos os contratos de empréstimo do usuário (como mutuário ou credor).
    - `GET /api/v1/loan/{loan_id}/installments`: Detalha o cronograma de parcelas de um empréstimo.
    - `POST /api/v1/loan/{loan_id}/pay-installment`: Processa o pagamento de uma parcela de empréstimo.

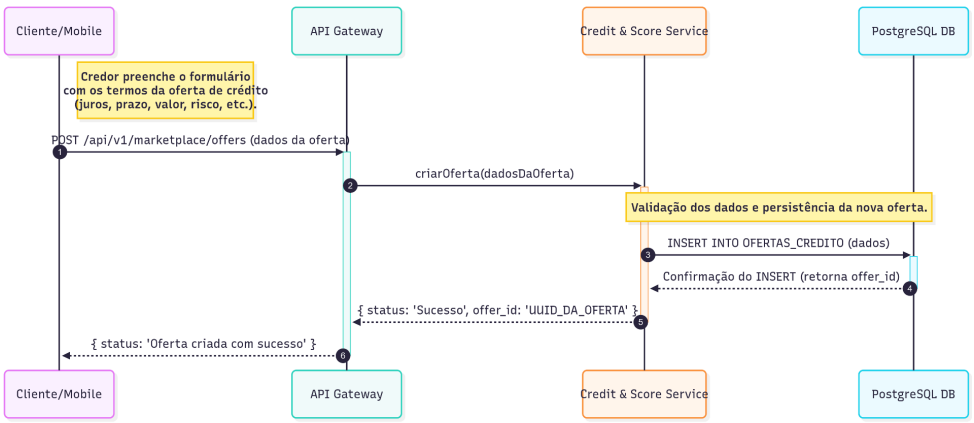



Diagrama de sequência Criar Oferta

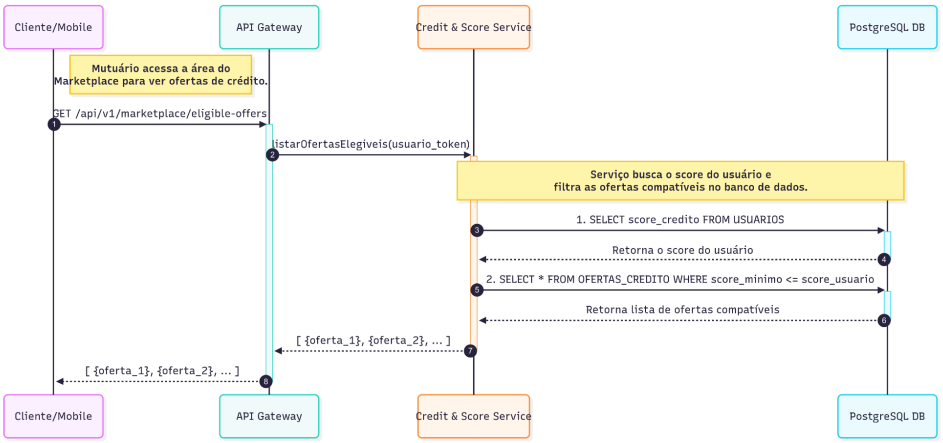



Diagrama de sequência Listar Ofertas

# 5. Imagens Ilustrativas para Fluxos e Integrações

## Fluxo de Transferência P2P

O fluxo se inicia com a autenticação do usuário. No dashboard, ele acessa a função de transferência, informa o ID do destinatário e o valor. O sistema valida se o saldo é suficiente. Se positivo, a transação é processada de forma atômica (débito na origem e crédito no destino) e um registro imutável é criado na tabela `TRANSACAO`. Finalmente, os saldos de ambos os usuários são atualizados na interface.

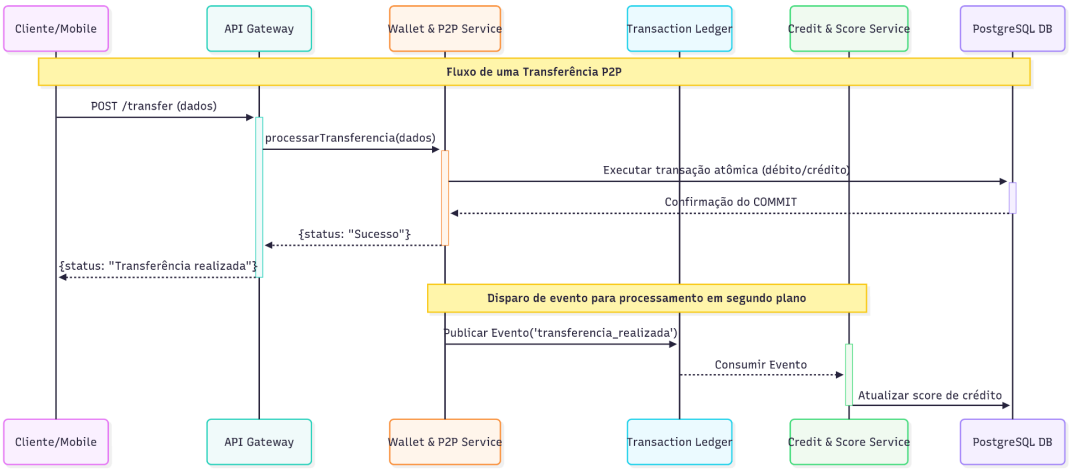



Diagrama sequência do Fluxo de uma Transferência P2P

## Arquitetura do Sistema

A arquitetura é baseada em microsserviços para garantir manutenibilidade e escalabilidade.

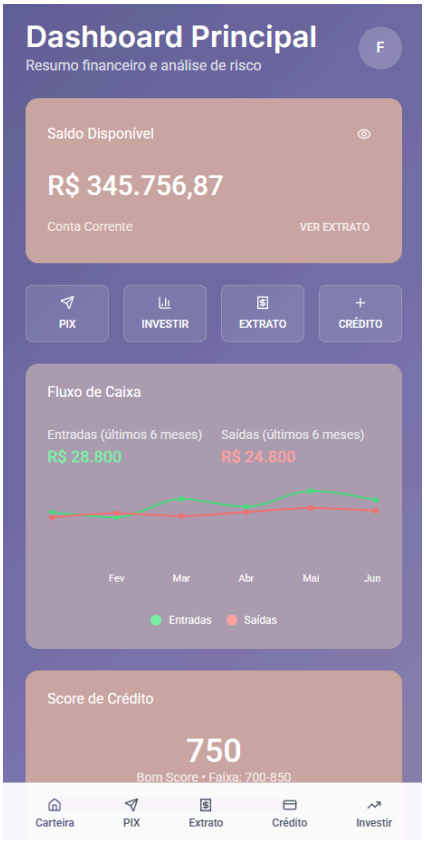
1. **Client (React):** A interface com o usuário que se comunica com o backend através de um API Gateway.
2. **API Gateway :** Centraliza as requisições, aplicando roteamento, autenticação e balanceamento de carga.
3. **Core Services (FastAPI):** Microsserviços responsáveis pela lógica de negócio:
    - **Wallet & P2P Service:** Gerencia saldos e executa as transferências P2P.
    - **Credit & Score Service:** Calcula o score de crédito e filtra as ofertas do marketplace.
4. **Transaction Ledger e Banco de Dados:** O Apache Kafka é recomendado para atuar como um log imutável de eventos, enquanto o PostgreSQL armazena o estado atual dos dados (saldos, usuários).

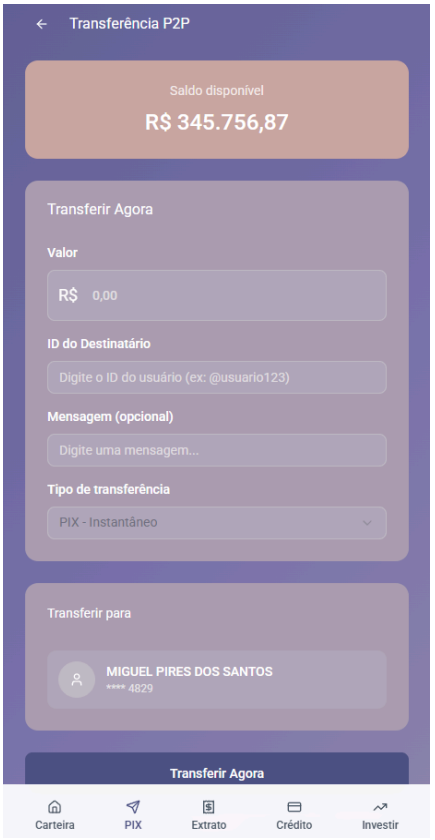
## 6. Protótipo Front

### Frontend Dashboard (React.js)

A interface, desenvolvida em React, utiliza componentes para exibir informações dinâmicas. O estado é gerenciado localmente ou com bibliotecas de estado.

### Protótipo de Frontend




À esquerda, o ponto de entrada do aplicativo. Exibe o saldo atual da carteira, um resumo gráfico do fluxo de caixa e score de crédito do usuário. Ao lado, a rota para transferências P2P.



As rotas para marketplace de ofertas e de ofertas ativas. Nelas, é possível visualizar as ofertas ativas e seus detalhes, além de criar novas ofertas de crédito a serem disponibilizadas no marketplace.

**Estrutura do Banco de Dados (PostgreSQL)**

O schema foi projetado para garantir a integridade e performance, utilizando UUIDs como chaves primárias para evitar enumeração e garantir unicidade global. A aplicação de transações ACID é mandatória para todas as operações que alteram saldos.

## 7. Considerações de Implementação

**Requisitos Técnicos**

- **Infraestrutura de Servidor:**
  - **Aplicação Backend:** Python 3.8+ com FastAPI.
  - **Banco de Dados:** PostgreSQL 15+ para produção.
  - **Containerização:** Docker e Docker Compose para orquestração dos serviços (`db`, `backend`).
  - **Event Streaming:** Apache Kafka para o ledger de transações imutáveis.

**Integrações Externas**

- **KYC/KYB:** Necessidade de integração com um serviço de verificação de identidade para cumprir regulações.
- **Certificados SSL:** Para garantir a comunicação segura via HTTPS (TLS 1.2+).

**Métricas de Sucesso:**

- ☐ **Criação de usuário**
- ☐ **Criação de oferta e busca**
- ☐ **Realização de empréstimo**

**Próximos Passos**

Após a aprovação do pré-projeto, será desenvolvido o MVP, permitindo que a ideia seja validada e novas funcionalidades sejam adicionadas à medida que se mostrem necessárias. A metodologia ágil será aplicada para que o projeto evolua constantemente em busca de aprimoramentos, evoluindo em direção a um produto completo que cumpra sua proposta de agregar valor por meio da facilitação de operações de crédito.