

MVP - Controle Financeiro Compartilhado

Objetivo do MVP

Permitir que usuários criem grupos, registrem despesas compartilhadas e visualizem quem deve o quê para quem de forma simples e clara.

Funcionalidades Essenciais

1. Autenticação Básica

- **Login/Cadastro simples** (email + senha)
- **Perfil básico** (nome, foto opcional)
- **Logout**

Justificativa: Base necessária para identificar usuários e associar dados.

2. Gerenciamento de Grupos

- **Criar grupo** (nome + descrição)
- **Convidar membros** (por email ou código)
- **Aceitar convites**
- **Visualizar membros do grupo**

Justificativa: Core da aplicação - sem grupos não há compartilhamento.

3. Registro de Despesas

- **Adicionar despesa** (valor, descrição, data, categoria básica)
- **Escolher quem pagou**
- **Definir divisão:** igual entre todos ou valores específicos
- **Listar despesas do grupo**
- **Editar/Excluir despesas próprias**

Justificativa: Funcionalidade principal que entrega valor imediato.

4. Cálculo de Saldos

- **Saldo individual** (quanto cada um deve/tem a receber)
- **Resumo de quem deve para quem**
- **Total gasto por pessoa**

Justificativa: O que realmente resolve o problema dos usuários.

5. Dashboard Simples

- **Resumo do grupo** (total gasto, saldo pessoal)
- **Últimas despesas**
- **Status de débitos/créditos**

Justificativa: Visão geral rápida e intuitiva.

O que NÃO entra no MVP

Funcionalidades para Versões Futuras

- **IA/Alertas inteligentes** - Complexo demais para MVP
- **Relatórios avançados** - Dashboards simples já entregam valor
- **Múltiplas moedas** - Adiciona complexidade desnecessária
- **Fotos de recibos** - Feature nice-to-have
- **Notificações push** - Pode ser email simples inicialmente
- **Pagamentos integrados** - Foco no controle, não no pagamento
- **Categorias avançadas** - 3-5 categorias básicas são suficientes
- **Histórico/Auditoria detalhada** - Adiciona complexidade
- **Temas/Personalização** - Não é prioridade

Estrutura Técnica Mínima

Backend (Spring Boot)

Controllers:

- AuthController (login/register)
- UserController (perfil)
- GroupController (CRUD grupos)
- ExpenseController (CRUD despesas)
- BalanceController (cálculos)

Models:

- User
- Group
- Expense
- UserGroup (relacionamento)

Services:

- AuthService
- GroupService
- ExpenseService
- BalanceService

Frontend (React)

Pages:

- Login/Register
- Dashboard
- Group Detail
- Add Expense
- Profile

Components:

- ExpenseList
- BalanceCard
- MemberList
- ExpenseForm

Database (PostgreSQL)

Tables:

- users
- groups
- expenses
- user_groups
- expense_splits

Valor

- Usuário consegue dividir conta do restaurante em 5 minutos
- Sabe exatamente quanto deve/tem a receber
- Evita discussões sobre "quem pagou o quê"

Viabilidade

- Stack conhecida (React + Spring Boot + PostgreSQL)
- Sem integrações externas complexas
- CRUD simples + cálculos matemáticos básicos

Tempo

- **Estimativa: 6-8 semanas** para equipe de 3-4 pessoas
- Backend: 3-4 semanas
- Frontend: 3-4 semanas
- Integração e testes: 1-2 semanas

Roadmap de Desenvolvimento

Semana 1-2: Setup + Backend Base

- Configuração do projeto
- Autenticação
- Models e migrations
- APIs básicas de usuário e grupo

Semana 3-4: Core Backend

- CRUD de despesas
- Lógica de cálculo de saldos
- Testes unitários básicos

Semana 5-6: Frontend Base

- Setup React + routing
- Telas de login e dashboard
- Integração com APIs

Semana 7-8: Finalização

- Tela de despesas e saldos

- Polimento da UI
- Testes de integração
- Deploy

Dicas de Implementação

1. **Comece pelos cálculos:** A lógica de divisão de despesas é o coração da aplicação
2. **UI Mobile-first:** Maioria dos usuários usará no smartphone
3. **Estados de loading:** Feedback visual para todas as operações
4. **Validações simples:** Campos obrigatórios e valores positivos
5. **Dados de teste:** Crie grupos e despesas de exemplo para demonstração

UX Mínima mas Essencial

- **Responsivo:** Funciona bem no mobile
- **Cores claras:** Verde para crédito, vermelho para débito
- **Loading states:** Não deixar usuário no escuro
- **Mensagens de erro:** Claras e acionáveis
- **Navegação intuitiva:** Máximo 3 cliques para qualquer funcionalidade

Lembre-se: MVP não é sobre fazer tudo, mas sobre fazer o essencial muito bem! 🎯