
RFP: Sistema de Consolidação de Ativos Financeiros (AssetTrack)

Data: 18 de Dezembro de 2025

Responsável: André Campos / Group Decode Solution

1. Persona do Usuário

Nome: Eduardo, o "Investidor em Ascensão"

Perfil: Investidor com portfólio diversificado (BRL/USD/Cripto) que busca centralização.

Desejo: Automatizar o cálculo de preço médio e ver a alocação de ativos em tempo real sem usar planilhas complexas.

Dor: Falta de tempo para atualizar cotações e dificuldade em consolidar investimentos internacionais.

2. Requisitos do Sistema

2.1 Requisitos Funcionais (RF)

- **RF01 - Fluxo de Onboarding:** Cadastro -> Verificação de E-mail -> Redirecionamento Automático para Checkout (Stripe).
- **RF02 - Sistema de Indicação:** Código único de 6 dígitos gerado no cadastro. Suporte a rastreamento de quem indicou quem.
- **RF03 - Status de Adimplência:** O sistema deve validar no banco de dados se o `subscription_status` é `active` antes de liberar qualquer rota de dados.
- **RF04 - Cadastro em Lote (Assets):** Rota para importar múltiplos ativos de uma vez via JSON.
- **RF05 - Lançamento em Lote (Transactions):** Rota para importar histórico de compras/vendas via JSON.
- **RF06 - Sincronização de Mercado:** Rota administrativa para atualizar preços globais de ativos 2x ao dia.

2.2 Requisitos Não Funcionais (RNF)

- **RNF01 - Segurança de Acesso:** Implementação de Middleware de autenticação JWT e verificação de status de pagamento.
 - **RNF02 - Integridade Financeira:** Uso de Hashing para senhas e transações atômicas no banco para garantir que o preço médio não seja corrompido em imports massivos.
-

3. Arquitetura do Sistema (Architecture.md)

Modelo: Monolito Modular

A solução será construída como um **Monolito Modular**. Diferente de um monolito tradicional bagunçado, o código é separado em módulos independentes (User, Assets, Finance, Payments).

Vantagens:

1. **Custo Operacional Mínimo:** Roda em uma única instância, economizando recursos que seriam gastos em múltiplos microserviços.
2. **Facilidade de Deployment:** Pipeline única de CI/CD.
3. **Consistência de Dados:** Facilita transações ACID no PostgreSQL (Supabase).

4. Esquema do Banco de Dados

SQL

-- Perfis e Controle de Acesso

```
CREATE TABLE profiles (  
  id UUID REFERENCES auth.users ON DELETE CASCADE PRIMARY KEY,  
  full_name TEXT,  
  phone TEXT,  
  birth_date DATE,  
  email_verified BOOLEAN DEFAULT FALSE,  
  referral_code VARCHAR(6) UNIQUE,  
  referred_by UUID REFERENCES profiles(id),  
  subscription_status VARCHAR(20) DEFAULT 'inactive', -- active, inactive, past_due  
  stripe_customer_id TEXT,  
  created_at TIMESTAMP WITH TIME ZONE DEFAULT NOW()  
);
```

-- Ativos do Usuário

```
CREATE TABLE assets (  
  id UUID DEFAULT uuid_generate_v4() PRIMARY KEY,  
  user_id UUID REFERENCES profiles(id),  
  ticker VARCHAR(20) NOT NULL,  
  name TEXT,  
  type VARCHAR(50), -- Ação, Fundo, Renda Fixa, etc.  
  currency VARCHAR(5), -- R$, USD, £, EUR  
  quantity DECIMAL DEFAULT 0,  
  avg_price DECIMAL DEFAULT 0,  
  current_price DECIMAL DEFAULT 0,  
  updated_at TIMESTAMP WITH TIME ZONE DEFAULT NOW()  
);
```

-- Histórico de Compras e Vendas

```
CREATE TABLE transactions (  
  id UUID DEFAULT uuid_generate_v4() PRIMARY KEY,  
  user_id UUID REFERENCES profiles(id),  
  asset_id UUID REFERENCES assets(id),  
  type VARCHAR(20) NOT NULL, -- Buy, Sell, Dividend, etc.  
  quantity DECIMAL NOT NULL,  
  price DECIMAL NOT NULL,  
  total_value DECIMAL NOT NULL,  
  created_at TIMESTAMP WITH TIME ZONE DEFAULT NOW()  
);
```

```
id UUID DEFAULT uuid_generate_v4() PRIMARY KEY,  
asset_id UUID REFERENCES assets(id),  
type VARCHAR(10), -- BUY ou SELL  
value DECIMAL NOT NULL,  
quantity DECIMAL NOT NULL,  
transaction_date TIMESTAMP WITH TIME ZONE  
);
```

5. Detalhamento das Rotas da API (Node.js)

Módulo: Auth & Onboarding

- **POST /auth/register**: Recebe dados do usuário e código de indicação. Cria o registro e retorna o link do Stripe para pagamento.
- **GET /auth/verify-email**: Valida o token de e-mail e atualiza **email_verified** para **true**.
- **GET /auth/session**: Verifica se o usuário está logado e se o **subscription_status** no banco é **active**.

Módulo: Pagamentos (Stripe Integration)

- **POST /payments/create-checkout**: Gera uma sessão do Stripe para o plano de R\$ 15,00.
- **POST /payments/webhook**: Rota de escuta para o Stripe. Quando o pagamento é confirmado, atualiza o **subscription_status** para **active**.

Módulo: Assets (Gestão de Carteira)

- **GET /assets**: Lista todos os ativos e o valor consolidado. (Requer status **active**).
- **POST /assets**: Cadastra novo ativo relacionando à um usuário.
- **POST /assets/bulk**: Recebe um Array de JSON: [{"ticker": "PETR4", "quantity": 100, "avg_price": 30.50}, ...]. O backend processa e cria todos de uma vez para o usuário logado.
- **DELETE /assets/:id**: Remove um ativo da carteira.

Módulo: Transactions (Movimentações)

- **POST /transactions/bulk**: Cadastra nova transação relacionando um ativo e um usuário.
- **POST /transactions/bulk**: Recebe um Array de JSON: [{"ticker": "AAPL", "type": "BUY", "value": 150.00, "quantity": 10, "date": "2025-12-01"}, ...]. O sistema registra as transações e recalcula o preço médio automaticamente.

Módulo: Admin / Automação

- **PATCH /admin/sync-market-prices**: Rota protegida que recebe o JSON massivo com cotações atuais e atualiza a coluna **current_price** em todos os registros da tabela **assets** que possuem aquele ticker.

6. Projeção Financeira e Análise de Lucro

Considerando o custo fixo do Stripe de aproximadamente **3.99% + R\$ 0,39** por transação.

Número de Clientes	Faturamento Bruto (R\$)	Taxas Stripe (Est. 7%*)	Imposto (7%)	Lucro Líquido (R\$)	Margem de Lucro
10 Clientes	R\$ 150,00	R\$ 10,50	R\$ 10,50	R\$ 129,00	86%
50 Clientes	R\$ 750,00	R\$ 52,50	R\$ 52,50	R\$ 645,00	86%
100 Clientes	R\$ 1.500,00	R\$ 105,00	R\$ 105,00	R\$ 1.290,00	86%

**Nota: Estimamos 7% para cobrir a taxa percentual + o custo fixo por transação de R\$ 0,39 e possíveis estornos.*

7. Fluxo de Negócio para aprovação do PM

1. **Usuário** preenche o formulário de cadastro.
2. **Sistema** envia e-mail de confirmação e exibe tela de "Aguardando Pagamento".
3. **Usuário** é redirecionado para o Stripe e paga R\$ 15,00.
4. **Stripe** avisa nosso Backend via Webhook.
5. **Sistema** libera o acesso alterando `subscription_status` para `active`.
6. **Usuário** faz o upload em lote (Bulk) de sua carteira antiga e começa a acompanhar os gráficos.