Documentacao Financeira - Integracao Stripe com Custodia (Escrow)

1. Viso Geral

O fluxo financeiro do projeto Nobile ser baseado em pagamento com reteno, onde a plataforma recebe o valor total de uma venda e s repassa ao vendedor aps confirmao de entrega por parte do comprador.

2. Etapas do Fluxo de Venda

A. Compra e Pagamento

1. Usurio comprador escolhe um relgio e clica em Comprar agora.
2. Backend cria uma Stripe Checkout Session.
3. Stripe processa o pagamento com application_fee definida, mas o valor total retido pela plataforma.
4. A compra registrada no banco como status = "pago / aguardando envio".

B. Envio e Entrega

5. O vendedor envia o produto (informa cdigo de rastreio).
6. O comprador recebe o produto e confirma no sistema (ou o sistema confirma aps prazo X dias sem disputa).

C. Repasse ao Vendedor

7. Plataforma dispara um payout manual via API Stripe:
   - Deduz comisso da plataforma (j foi aplicada via application_fee).
   - Transfere o valor restante para a conta Stripe Connect do vendedor.

3. Configurao Tcnica com Stripe

Checkout Session (exemplo):

```
const session = await stripe.checkout.sessions.create({
  payment_method_types: ['card'],
  line_items: [...],
  mode: 'payment',
  success_url: '...',
  cancel_url: '...',
  payment_intent_data: {
    application_fee_amount: plataforma_fee_em_centavos,
    transfer_data: {
      destination: 'acct_xxx_vendedor'
    },
    capture_method: 'automatic'
  }
});
```

4. Exemplo Numrico

Venda de um relgio por R$ 10.000,00:

- Comisso da plataforma: 6,5% -> R$ 650,00
- Taxa Stripe: ~3,99% -> R$ 399,00
- Repasse ao vendedor: R$ 8.951,00

5. Banco de Dados (PostgreSQL)

Tabelas envolvidas:
- orders
  id, buyer_id, seller_id, product_id, status, payment_status, total_amount, net_amount, created_at

- transactions
  id, order_id, stripe_payment_id, type (checkout|payout), status, application_fee, transfer_amount, created_at

- vendors
  id, user_id, stripe_account_id, verified

6. Segurana e Protees

- Confirmao do recebimento obrigatria antes do repasse.
- Opo de disparo automtico do repasse aps X dias (sem contestao).
- Registro de eventos via Stripe Webhooks:
  - checkout.session.completed
  - payment_intent.succeeded
  - transfer.paid

7. Endpoints sugeridos

POST /checkout/initiate
-> Cria sesso Stripe, registra pedido "pendente"

POST /webhook/stripe
-> Atualiza status do pedido com base em eventos

POST /order/:id/confirm-delivery
-> Confirma entrega e libera repasse

POST /payout/:orderId
-> Realiza transferncia ao vendedor via Stripe

GET /admin/orders
-> Visualizao financeira completa

8. Benefcios do Modelo

- Evita fraudes e garante entrega antes do repasse.
- Protege o comprador (via reteno).

- Automatiza repasses com Stripe mantendo conformidade (KYC, antifraude).
- Permite controle completo sobre as taxas e comisses.

9. Parcelamento com Valor Dinamico

A plataforma pode calcular o valor total da compra com base no nmero de parcelas escolhidas pelo comprador,
repassando assim os juros ao cliente.

1. O sistema calcula o valor final com juros embutidos e exibe ao comprador (ex: 10x de R$ 1.160 Total R$ 11.600).
2. O backend cria a sesso Stripe com esse valor total ajustado.
3. O Stripe cobra o valor total vista, e o parcelamento real depende do banco emissor do carto.

Exemplo de clculo:

parcelas = 10
taxaMensal = 0.015 (1,5%)
valorProduto = 10000

valorFinal = valorProduto * (1 + taxaMensal) ** parcelas

O valorFinal ser enviado para o Stripe como total_amount. Dessa forma, o custo dos juros repassado ao comprador.