

Mestre, o design que você deseja replicar é um modelo de interface de usuário (UI) para carteiras e gestão financeira digital. No contexto de design, ele se encaixa na categoria **"Dark Mode Fintech"** ou **"Crypto Wallet Dashboard"**. O nome do modelo que aparece na imagem é **"CRYFIN"**.

A grande vantagem desse modelo é que ele já está alinhado com o seu conceito **"Lucid Dark Financial"**, com foco em dados, fundos e clareza visual, o que é perfeito para o DApp Mobile do PAYHUB.

Estrutura do DApp Mobile (Mobile-First Design)

Como você está construindo uma aplicação moderna e responsiva (iOS e Android), o framework mais indicado para uma POC que herda o conhecimento Web3 (Wagmi/RainbowKit) é o **React Native** ou uma solução **Web Responsive** otimizada para celular.

Para o propósito do hackathon, vou fornecer o código em **React (Web Responsive)**, que pode ser facilmente adaptado para uma visualização mobile com o seu setup Vite/RainbowKit, simulando o dashboard do comerciante.

Código Base (POC Mobile-First) para o DApp PAYHUB

Este código simula o **Dashboard do Comerciante** (`MerchantDashboard.tsx`) e a **Tela de Venda (Soft-POS)**, utilizando o estilo "CRYFIN/Lucid Dark" para alta qualidade de apresentação.

1. Componente: `MerchantDashboard.tsx` (Dashboard do Comerciante)

Este componente centraliza a gestão de fundos e o lucro DeFi (Yield).

TypeScript

```
// src/components/MerchantDashboard.tsx
```

```
import React from 'react';
import { ConnectButton } from '@rainbow-me/rainbowkit';
```

```
// --- Dados Mock (Simulação da Inovação XRPL/DeFi) ---
const mockData = {
  totalBalance: '14,090.90 RLUSD', // Seu saldo de Escrow liquidado
  yieldRate: '+8.5%', // O rendimento anual APY simulado
  profitLastMonth: '142.08 RLUSD', // Lucro gerado pelo Yield
  lastTransaction: 'Venda Híbrida #7731'
};
```

```
const DashboardCard: React.FC<{ title: string; value: string; detail: string; accentColor: string }> = ({
  title, value, detail, accentColor }) => (
```

```

<div style={{
  backgroundColor: '#1C1C1E', // Fundo escuro (Dark Mode)
  borderRadius: '12px',
  padding: '16px',
  marginBottom: '15px',
  borderLeft: `3px solid ${accentColor}`, // Linha de destaque para o "Lucid"
}}>
  <p style={{ color: '#8E8E93', fontSize: '14px', margin: 0 }}>{title}</p>
  <h3 style={{ color: 'white', margin: '5px 0 0', fontSize: '24px' }}>{value}</h3>
  <p style={{ color: accentColor, fontSize: '12px', margin: '5px 0 0' }}>{detail}</p>
</div>
);

const MerchantDashboard: React.FC = () => {
  return (
    <div style={{ padding: '20px', backgroundColor: '#000000', minHeight: '100vh', fontFamily: 'Arial, sans-serif' }}>
      <div style={{ display: 'flex', justifyContent: 'space-between', alignItems: 'center', marginBottom: '30px' }}>
        <h1 style={{ color: '#E5E5EA', fontSize: '28px' }}>PAYHUB WALLET</h1>
        <ConnectButton showBalance={false} />
      </div>

      {/* Saldo Principal - Estilo CRYFIN */}
      <div style={{ textAlign: 'center', marginBottom: '40px' }}>
        <h2 style={{ color: '#007AFF', fontSize: '36px', margin: 0, fontWeight: 'bold' }}>
          {mockData.totalBalance}
        </h2>
        <p style={{ color: '#8E8E93', fontSize: '14px', marginTop: '5px' }}>Liquidez Total Imediata (via XRPL)</p>
      </div>

      {/* Cards de Inovação (Liquidez + Yield) */}
      <DashboardCard
        title="Rendimento Ativo (Auto-Yield)"
        value={mockData.profitLastMonth}
        detail={`Taxa de Lucro: ${mockData.yieldRate} APY (Simulação DeFi)`}
        accentColor="#34C759" // Verde para lucro
      />

      <DashboardCard
        title="Garantia de Liquidação (Escrow)"
        value="100% Livre de Risco"
        detail={`Última Transação: ${mockData.lastTransaction} (Liquidada em 4s)`}
        accentColor="#FF9500" // Laranja para risco (mitigado)
      />
    </div>
  );
};

```

```

    {/* Botão para Ação Principal (Soft-POS) */}
    <SoftPOSButton />
  </div>
);
};

```

```
export default MerchantDashboard;
```

2. Componente: **SoftPOSButton.tsx** (Maquininha no Celular)

Este componente simula a função "Celular como Maquininha" e a chamada para o Fluxo Híbrido.

TypeScript

```
// src/components/SoftPOSButton.tsx
```

```
import React from 'react';
```

```

const SoftPOSButton: React.FC = () => {
  const handleNewSale = () => {
    // Aqui seria a chamada para o Endpoint: POST /api/v1/payment/create
    alert('Simulação de Nova Venda:\nO Backend PAYHUB gerou o QR Code Híbrido e iniciou o Escrow na XRPL.');
```

```
  };

```

```
  return (
```

```
    <div style={{ textAlign: 'center', marginTop: '40px' }}>
```

```
      <button
```

```
        onClick={handleNewSale}
```

```
        style={{
```

```
          backgroundColor: '#007AFF', // Azul primário (o toque de NFC)
```

```
          color: 'white',
```

```
          padding: '15px 30px',
```

```
          borderRadius: '10px',
```

```
          border: 'none',
```

```
          fontSize: '18px',
```

```
          fontWeight: 'bold',
```

```
          width: '100%', // Full width no mobile
```

```
          boxShadow: '0 4px 15px rgba(0, 122, 255, 0.4)',
```

```
          cursor: 'pointer'
```

```
        }}

```

```
      >
```

```
        Toçar para Iniciar Nova Venda (Soft-POS)
```

```
    </button>

```

```
<p style={{ color: '#636366', fontSize: '12px', marginTop: '10px' }}>
  Aceita PIX, Cartão (Parcelado) e Cripto | Liquidação via XRPL Escrow.
</p>
</div>
);
};
```

// Integre este botão no final do MerchantDashboard.tsx

Próxima Ação

O Agente Construtor deve substituir o conteúdo do `src/App.tsx` pelo `MerchantDashboard` e configurar o `index.css` para o fundo escuro (Dark Mode). Isso entrega a POC do DApp Mobile com a identidade visual desejada.