

Dados do Participante

Nome: Cezar de Sousa Tosta

Email: cezar.tosta@hotmail.com

Ferramentas utilizadas: Postman, Markdown, VS Code

Data de início: 30/09/2025

Data de conclusão: 02/10/2025

Relatório

1. Qual trilha você testou?

(x) Trilha A – Smart Wallet, KYC, Fiat, Portfolio, History

() Trilha B – Smart Wallet, Swaps, Transfer, Portfolio, History

() Trilha C – Smart Wallet, Liquidity Pools, Portfolio, History

Justificativa: escolhi a Trilha A por seu foco em compliance e integração com sistemas financeiros tradicionais, áreas estratégicas para adoção empresarial de Web3.

2. Quais endpoints você testou com mais profundidade?

Total de testes: 20 testes distribuídos em 5 sessões (Daily Boards)

Smart Wallet (13 testes):

- POST /wallets/register - Criação de wallets com validações
- GET /wallets/address - Consulta de wallet específica
- GET /wallets - Listagem de wallets
- GET /wallets/{address}/portfolio - Visualização de ativos
- GET /wallets/{address}/history - Histórico de transações
- POST /wallets/{address}/deposit - Criação de depósitos
- PATCH /wallets/{id}/metadata - Atualização de metadados
- PATCH /wallets/transactions/{id}/metadata - Metadados de transações

KYC (6 testes):

- POST /kyc/individual-verification-sessions/standard - Criação de sessão
- GET /kyc/individual-verification-sessions/standard/{sessionId} - Consulta de status
- POST /kyc/.../standard/{sessionId}/process - Processamento de verificação

Fiat (1 teste):

- POST /fiat/deposit/quote - Tentativa de criação de cotação (bloqueado por permissão)

Portfolio + History (2 testes):

- Reteste dos endpoints no contexto completo da Trilha A



3. Quais foram os principais bugs encontrados?

Bug #1: GET /wallets/address ignora parâmetro address ●

- **Severidade:** CRÍTICA
- **Comportamento:** Endpoint ignora completamente o parâmetro address fornecido e sempre retorna a wallet com salt "0"
- **Reprodutibilidade:** 100%
- **Impacto:** Impossível consultar wallets específicas por endereço. Usuários com múltiplas wallets não conseguem acessar wallets com salt diferente de "0"

Bug #2: GET /wallets/address exige parâmetros redundantes ●

- **Severidade:** MÉDIA
- **Comportamento:** Requer 3 parâmetros (address, externallyOwnedAccount, factory) quando apenas address deveria ser suficiente
- **Reprodutibilidade:** 100%
- **Impacto:** Aumenta complexidade desnecessariamente. Não segue padrão Web3

Bug #3: Mensagens genéricas para erros específicos ●

- **Severidade:** MÉDIA
- **Comportamento:** Erro de EOA inválido retorna mensagem genérica "Failed to prepare deployment"
- **Reprodutibilidade:** 100%
- **Impacto:** Dificulta debug

Bug #4: Campo salt com valor padrão não documentado ●

- **Severidade:** MÉDIA
- **Comportamento:** API aceita requisição sem salt e usa "0" silenciosamente
- **Reprodutibilidade:** 100%
- **Impacto:** Inconsistência entre documentação e implementação

Bug #5: Status code incorreto para autenticação ●

- **Severidade:** MÉDIA
- **Comportamento:** Retorna 403 ao invés de 401 quando API key está ausente
- **Reprodutibilidade:** 100%
- **Impacto:** Dificulta diagnóstico correto

Bug #6: Erro 500 intermitente no portfolio ●

- **Severidade:** MÉDIA
- **Comportamento:** Primeira chamada retorna 500, segunda chamada funciona

- **Reprodutibilidade:** Intermitente
- **Impacto:** Sugere instabilidade (cold start?)

Bug #7: Status code inadequado para saldo insuficiente ●

- **Severidade:** MÉDIA
- **Comportamento:** Retorna 500 ao invés de 400 para erro de saldo insuficiente
- **Reprodutibilidade:** 100%
- **Impacto:** Categorização incorreta como erro de servidor

Bug #8: Formato de metadata inconsistente ●

- **Severidade:** MÉDIA
- **Comportamento:** Wallet metadata aceita object, transaction metadata aceita apenas string
- **Reprodutibilidade:** 100%
- **Impacto:** Inconsistência entre endpoints similares



4. Quais comportamentos inesperados você identificou?

1. Documentação de KYC incorreta sobre campos obrigatórios

- Campos marcados como opcionais são obrigatórios na prática (livenessRequired, email, address, city, state, postalCode)

2. Processamento KYC retorna 204 mas falha silenciosamente

- POST /process retorna sucesso mas verificação falha de forma assíncrona
- Sem campo "failureReason" explicando o motivo

3. Funcionalidade Fiat bloqueada sem documentação prévia

- Todos endpoints retornam 403
- Documentação não menciona pré-requisitos ou processo de habilitação
- Requer aprovações que levam 5-10 dias úteis

4. Portfolio e History não integram com KYC/Fiat

- Endpoints são exclusivos para dados blockchain
- Separação arquitetural não documentada claramente

5. ETH nativo não suportado em deposit

- Não documentado quais tokens são suportados
- Descoberta por tentativa e erro



5. Como foi a experiência de usar a API?

A documentação foi suficiente?

Nota: 3/5 ★★☆☆

Positivo: Estrutura organizada, tipos bem definidos, enums documentados

Negativo: Campos obrigatórios incorretos, pré-requisitos ausentes, comportamentos padrão não documentados, falta de guias end-to-end

As mensagens de erro ajudaram?

Nota: 4/5 ★★★★★

Positivo: Pattern consistente, validações listam todos problemas de uma vez, Traceld presente

Negativo: Algumas mensagens genéricas, status codes ocasionalmente incorretos, falta campo "failureReason" em alguns casos

O fluxo fez sentido?

Nota: 4/5 ★★★★★

Positivo: Fluxos principais são intuitivos, nomenclatura consistente, verbos HTTP corretos

Negativo: GET /wallets/address com 3 parâmetros confuso, falta endpoint de visão geral da conta, processamento assíncrono não é óbvio

O tempo de resposta era razoável?

Nota: 4/5 ★★★★★

Positivo: Maioria < 500ms, performance consistente após warmup

Negativo: Erro 500 intermitente no portfolio, alguns endpoints ocasionalmente > 1s



6. Alguma funcionalidade estava ausente ou incompleta?

1. GET /tokens/supported - Lista de tokens suportados por chain
2. GET /account/overview - Visão consolidada da conta
3. GET /wallets/{address} - Alternativa ao endpoint problemático atual
4. GET /account/timeline - Histórico unificado (on-chain + off-chain)
5. Campo "failureReason" - Em sessões KYC com status FAILED
6. Ambiente sandbox para Fiat - Testes sem aprovações reais



7. Quais melhorias você sugere?

Nomes de Campos:

- Renomear externallyOwnedAccount → ownerAddress ou eoa
- Padronizar formato de metadata entre endpoints

Design de Endpoints:

- Criar GET /wallets/{address} com path parameter
- Adicionar GET /account/overview para visão consolidada
- Criar GET /tokens/supported para listar tokens disponíveis

Lógica de Negócio:

- Validar presença de documentos antes de processar KYC

- Retornar 400 (não 500) para erros do cliente
- Adicionar campo "failureReason" em falhas de KYC

Fluxo Geral:

- Documentar pré-requisitos claramente (especialmente Fiat)
- Criar guias passo-a-passo de integração
- Adicionar Quick Start guides (< 5 minutos)

Retornos da API:

- Padronizar status codes consistentemente
- Enriquecer erros com campo "suggestions"
- Adicionar campo "nextSteps" em responses

Consistência:

- Padronizar formato de metadata (object em todos)
- Usar identificadores específicos consistentemente
- Implementar paginação cursor-based em todas listas



8. Como você avaliaria a estabilidade geral da API nesta trilha?

[x] Estável – poucos problemas, nada crítico

Justificativa:

Positivos:

- 100% de disponibilidade durante testes
- Performance geralmente boa (< 500ms)
- Validações robustas
- Segurança adequada

Problemas:

- 1 bug crítico (mas contornável)
- 7 bugs médios (não impedem uso)
- 0 bugs que causaram crashes ou perda de dados

Conclusão: A API está funcional e adequada para desenvolvimento/staging. Antes de produção em larga escala, recomenda-se corrigir o bug crítico e revisar documentação.



9. Há testes que você gostaria de ter feito, mas não conseguiu? Por quê?

1. Testes completos de Fiat

- **Motivo:** Funcionalidade bloqueada (403). Processo de habilitação requer 5-10 dias úteis. Participação iniciou na fase final do cronograma, sem tempo para aprovações.

2. Upload real de documentos KYC

- **Motivo:** Não tinha documentos de teste preparados. Tempo limitado. Formato exato do upload não estava claro.

3. Operações on-chain (Trilha B)

- **Motivo:** Escolha de trilha. Tempo limitado para testar duas trilhas completas.

4. Fluxo completo end-to-end

- **Motivo:** Múltiplos bloqueios (KYC FAILED, Fiat bloqueado, timing). Seria necessário 3-4 semanas para fluxo completo.

5. Testes de performance e carga

- **Motivo:** Fora do escopo de DX research. Ferramentas limitadas. Tempo limitado.



10. Comentários finais ou insights gerais?

Impressão Geral:

A API NotusLabs apresenta **base técnica sólida** com arquitetura bem pensada. A separação entre operações blockchain (on-chain) e compliance (off-chain) é correta e importante para segurança. O posicionamento estratégico - ponte entre Web2 e Web3 com compliance integrado - é valioso e necessário para adoção empresarial.

Principais Forças:

- Arquitetura e estrutura bem organizadas
- Segurança e controles de acesso adequados
- Performance aceitável para produção
- Proposta de valor clara e diferenciada

Principais Fraquezas:

- Documentação incompleta ou incorreta em pontos críticos
- Inconsistências entre endpoints similares
- Bug crítico impede funcionalidade core
- Falta de ambiente sandbox para Fiat

Recomendações Prioritárias:

1. **Crítico:** Corrigir bug GET /wallets/address
2. **Crítico:** Revisar documentação completa (especialmente KYC e Fiat)
3. **Importante:** Criar ambiente sandbox para Fiat
4. **Importante:** Padronizar status codes e formatos
5. **Desejável:** Adicionar endpoints de conveniência (overview, tokens supported)

Muito obrigado pela oportunidade de participar desta pesquisa!