

Superando Barreiras de Usabilidade em Aplicações Blockchain

Uma Análise de Interação Humano-Computador em DeFi sobre Redes Compatíveis com EVMs

Emanuele L. M. Martins¹, Bryan K. Ferreira¹

¹Instituto de Tecnologia e Liderança (INTELI)
Av. Prof. Almeida Prado, 520 - Butantã, São Paulo - SP, 05508-070 - Brazil

emanuele.martins@sou.inteli.edu.br, bryan@ime.usp.br

Resumo. O estudo investiga barreiras de usabilidade em aplicações de Finanças Descentralizadas (DeFi) executadas em redes compatíveis com a Ethereum Virtual Machine (EVM), mostrando que problemas de fluxo, terminologia e feedback comprometem a adoção, especialmente entre iniciantes. Para enfrentar essas limitações, o trabalho propõe uma interface aprimorada e a compara a uma versão não otimizada usando métricas de desempenho, número de cliques e o questionário NASA-TLX. Os resultados indicam que a interface melhorada aumenta a taxa de conclusão de tarefas, reduz cliques redundantes e diminui carga mental, frustração e pressão temporal, inclusive entre usuários experientes, que relatam maior fluidez e previsibilidade. O artigo conclui que refinamentos de usabilidade voltados para aplicações financeiras descentralizadas são determinantes para elevar confiança e adoção, recomendando a padronização de processos, mensagens menos técnicas e a redução de etapas críticas para mitigar a fadiga de operações e ampliar o alcance da Web3.

Abstract. This study examines usability barriers in Decentralized Finance (DeFi) applications operating on Ethereum Virtual Machine (EVM)-compatible networks, showing that issues related to user flow, terminology, and feedback hinder adoption, particularly among novice users. To address these challenges, the work proposes an improved interface and compares it with a non-optimized version using performance metrics, click counts, and the NASA-TLX questionnaire. The results indicate that the enhanced interface increases task-completion rates, reduces redundant interactions, and decreases mental demand, frustration, and time pressure, including among experienced users, who report greater fluency and predictability. The study concludes that targeted usability refinements are essential to strengthening user trust and adoption of decentralized financial applications, recommending process standardization, less technical messaging, and the reduction of critical steps to mitigate operational fatigue and broaden Web3 accessibility.

1. Introdução

Nos últimos anos, o avanço das tecnologias de blockchain tem possibilitado o surgimento de um novo paradigma financeiro, no qual a intermediação é substituída por contratos

inteligentes que operam de maneira autônoma e transparente. Dentro desse contexto, as aplicações de finanças descentralizadas (DeFi) se consolidaram como um dos pilares mais promissores da Web3, oferecendo instrumentos financeiros acessíveis, auditáveis e globais. Contudo, a promessa de democratização trazida por tais sistemas tem sido limitada por um fator frequentemente subestimado: a usabilidade. Apesar do rigor técnico e da robustez criptográfica dessas soluções, o processo de interação com carteiras digitais, a assinatura de transações e o gerenciamento de chaves privadas ainda impõem barreiras consideráveis, especialmente para usuários iniciantes. Essa complexidade técnica, aliada à falta de padronização de fluxos e de terminologias consistentes, contribui para uma experiência fragmentada que restringe a adoção em larga escala, evidenciando a necessidade de investigar a interação entre humanos e sistemas descentralizados sob a ótica da Interação Humano-Computador (IHC).

Pesquisas recentes indicam que a experiência do usuário é um fator crítico para a consolidação das tecnologias descentralizadas. Estudos sobre o comportamento de usuários em plataformas DeFi apontam que dificuldades em tarefas aparentemente simples, como conectar uma carteira, compreender taxas de transação ou interpretar mensagens de erro, resultam em altos índices de abandono logo nas primeiras interações. Além disso, o excesso de confirmações e de etapas técnicas durante operações rotineiras gera o fenômeno conhecido como fadiga de operações, no qual o esforço cognitivo supera os benefícios percebidos do sistema, reduzindo o engajamento e a confiança do usuário. Tais evidências revelam que a segurança técnica, isoladamente, não é suficiente para promover a adoção, a segurança percebida e a fluidez da interação são igualmente determinantes para o sucesso de uma aplicação descentralizada. Nesse sentido, compreender como os usuários se comportam, interpretam e reagem a esses sistemas é essencial para alinhar o design de interfaces às suas expectativas e limitações cognitivas.

Diante desse cenário, este trabalho tem como objetivo analisar e propor formas de superar barreiras de usabilidade em aplicações DeFi, com foco na redução da fadiga operacional e na ampliação da compreensão do usuário sobre os processos envolvidos. Para isso, se desenvolveu uma metodologia experimental baseada em testes A/B com duas versões de uma mesma plataforma, uma construída segundo princípios heurísticos de usabilidade e outra propositalmente desprovida desses preceitos, mas ambas sustentadas pelo mesmo contrato inteligente. Essa abordagem permite isolar o impacto do design da interface sobre a experiência do usuário, mensurando diferenças em eficiência, eficácia e satisfação. O estudo busca, assim, contribuir para o avanço das práticas de design e avaliação de usabilidade em ambientes descentralizados, oferecendo evidências empíricas sobre como decisões de interface influenciam a confiança, o esforço cognitivo e o engajamento dos usuários em ecossistemas de finanças descentralizadas.

2. Fundamentação Teórica

2.1. Fundamentos de Blockchain

Uma rede *blockchain* pode ser definida como um sistema de registros compartilhados, distribuídos e imutáveis, em que as transações são gravadas de forma permanente e só podem ser alteradas mediante consenso de todas as partes envolvidas [Bashir 2018]. Além disso, esse sistema elimina a necessidade de intermediários, permitindo que as transações ocorram diretamente entre os participantes da rede [Bashir 2018].

2.1.1. Histórico e Evolução da Tecnologia

A ideia da criação do que hoje conhecemos como blockchain surgiu a partir do conceito de dinheiro eletrônico e da busca por formas de representar valores monetários na internet [Nakamoto 2008]. Desde os anos 1980, já existiam protocolos de dinheiro eletrônico baseados no modelo proposto por David Chaum. Entretanto, a primeira implementação prática de uma moeda digital descentralizada aconteceu através do Bitcoin [Bashir 2018].

O Bitcoin surgiu em 2008, quando um pseudônimo chamado Satoshi Nakamoto publicou o documento técnico “Bitcoin: A Peer-to-Peer Electronic Cash System” [Nakamoto 2008]. Satoshi combinou ideias anteriores, como b-money [Dai 1998] e HashCash [Back 2002], para criar um sistema de dinheiro eletrônico totalmente descentralizado, sem necessidade de uma autoridade central para emissão ou validação de transações [Antonopoulos 2014]. O grande avanço dessa rede foi usar um sistema de computação distribuída chamado Proof-of-Work para resolver o problema do gasto duplo, permitindo que a rede chegasse a um consenso global sobre o estado das transações sem confiar em uma entidade central [Antonopoulos 2014]. O Bitcoin não é controlado por nenhum governo ou empresa. Em vez disso, milhares de computadores espalhados pelo mundo, formando a rede Bitcoin, como ilustrado na Figura 1, mantêm o sistema funcionando continuamente, 24 horas por dia. Para utilizar o Bitcoin, não é necessário se cadastrar ou criar uma conta em nenhum lugar, basta ter acesso à internet e usar um programa, como um aplicativo no celular.

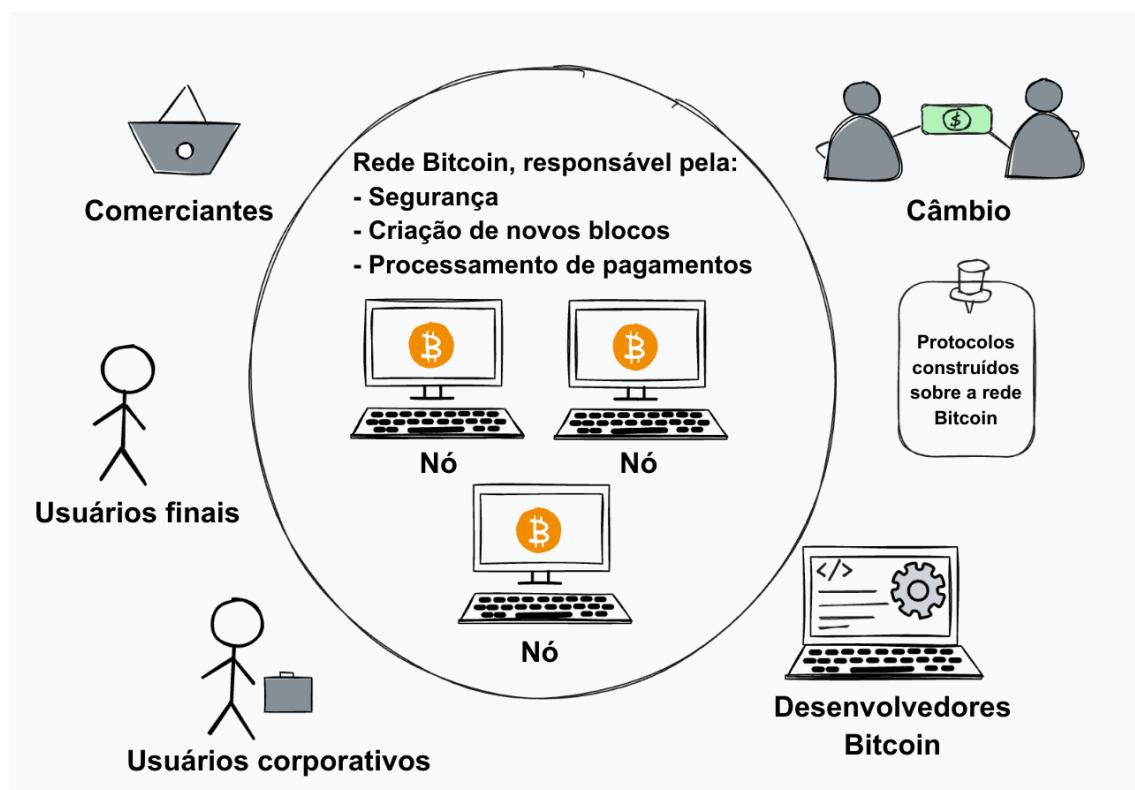


Figura 1. Participantes da rede. Fonte: Adaptado de [Rosenbaum 2019]

Ao longo do tempo, o *Bitcoin* demonstrou ser possível criar um sistema monetário

digital descentralizado e resistente à censura, sem a necessidade de intermediários. Entretanto, sua arquitetura foi projetada especificamente para transações financeiras ponto a ponto, limitando sua capacidade de suportar aplicações além do registro e transferência de valor. Essa limitação motivou o surgimento de novas plataformas de blockchain, como o *Ethereum*, cuja proposta foi ampliar o escopo da tecnologia para uma infraestrutura de propósito geral, capaz de executar contratos inteligentes e aplicações descentralizadas (dApps) sob consenso distribuído [Buterin 2014][Antonopoulos and Wood 2018].

Vitalik Buterin propôs a criação de uma máquina mundial de estados globais, permitindo que qualquer pessoa implementasse códigos que detêm valor e regras de negócio [Buterin 2014]. Esse projeto evoluiu e se tornou uma rede ativa em 2015, dando origem ao Ethereum, que abriu caminho para uma plataforma programável além da transferência de moeda [Antonopoulos and Wood 2018]. O coração dessa inovação é a *Ethereum Virtual Machine (EVM)*, uma máquina virtual descentralizada capaz de executar códigos de contratos inteligentes de forma segura e determinística em todos os nós da rede. A EVM garante que as regras programadas sejam seguidas fielmente e que as transições de estado ocorram de maneira idêntica em toda a *blockchain*, independentemente do participante [Antonopoulos and Wood 2018].

Na prática, o desenvolvimento desse novo conceito de rede mudou o foco do aspecto exclusivamente monetário para a infraestrutura em si. Dessa forma, o *Ethereum* permite a sincronização de transições de estado de qualquer tipo de dado, possibilitando a criação de *tokens*, aplicações financeiras, *stablecoins*, organizações autônomas descentralizadas (*Decentralized Autonomous Organizations — DAOs*) e diversas outras inovações.

2.2. Arquitetura e Funcionamento

2.2.1. Estrutura do bloco

Um bloco na *blockchain* é uma estrutura responsável por registrar e organizar transações de forma segura e ordenada, e é composto por duas partes principais, o cabeçalho e o corpo. O corpo do bloco reúne todas as transações incluídas, enquanto o cabeçalho, conhecido como Block Header, contém informações essenciais para identificar e validar o bloco, como o horário de criação, o nonce, o *hash* do bloco anterior (*parentHash*) e o *hash* das transações, chamado de Merkle Root [Bashir 2018].

A Merkle Root é gerada por meio da árvore de Merkle, uma estrutura que combina todos os *hashes* das transações até chegar a um único hash final, funcionando como um resumo do bloco [Bashir 2018]. Esse mecanismo permite verificar rapidamente se uma transação faz parte do bloco e garante que qualquer alteração em uma transação modifique a Merkle Root, preservando a integridade do conjunto. O *nonce*, por sua vez, é um número ajustado durante o processo de mineração para que o *hash* do bloco atenda aos requisitos definidos pela rede. Já o *hash* do bloco, gerado a partir de seu conteúdo, utilizando funções como Keccak-256 [Wood 2014] no caso do Ethereum, funciona como uma impressão digital única do bloco.

Além disso, o *parentHash* presente no cabeçalho conecta cada bloco ao anterior, formando uma cadeia contínua. Essa ligação por *hashes* garante a integridade e a imutabilidade da *blockchain*, pois qualquer alteração em um bloco modifica seu *hash*, comprometendo a sequência e invalidando todos os blocos seguintes. A figura 2 abaixo ilustra

como cada bloco depende criptograficamente do anterior, criando uma linha do tempo segura e verificável de todas as transações registradas na rede.

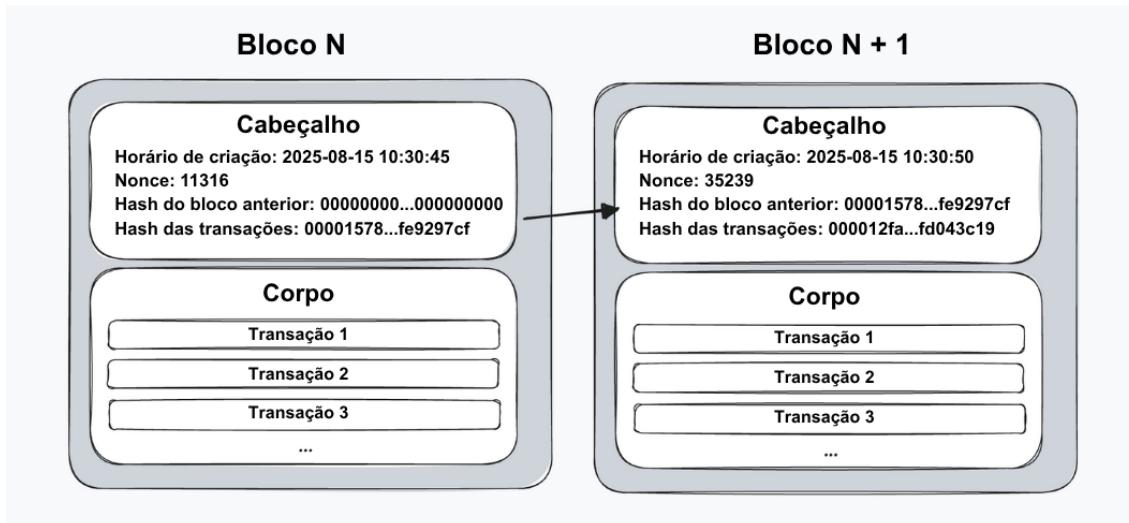


Figura 2. Exemplificação da estrutura de blocos na rede. Fonte: Autoria própria

2.2.2. Nós e Papéis na Rede

Para que uma rede descentralizada funcione corretamente, é necessário que diversos computadores, chamados de nós, executem o protocolo e se comuniquem entre si, formando uma infraestrutura robusta e distribuída. Esses nós desempenham diferentes funções, que, em conjunto, garantem a segurança, auditabilidade e acessibilidade da rede.

Os nós completos (*full nodes*) mantêm uma cópia local de todo o blockchain e do estado da rede, validando integralmente blocos e transações conforme as regras de consenso. Eles são fundamentais para a resiliência e verificabilidade do sistema, mas exigem considerável capacidade de armazenamento, largura de banda e tempo para sincronização [Bashir 2018].

Os clientes leves (*light clients*), por outro lado, não armazenam a cadeia completa nem reexecutam todas as transações. Eles verificam blocos utilizando apenas os cabeçalhos e provas de Merkle, confiando em nós completos para obter dados sob demanda. Por consumirem menos recursos, são ideais para carteiras digitais e dispositivos móveis, permitindo assinar e enviar transações de maneira eficiente [Antonopoulos and Wood 2018].

Além desses, existem os nós validadores, responsáveis por produzir e atestar blocos dentro do mecanismo de consenso da rede. Em sistemas Proof-of-Work (PoW), essa função é desempenhada por mineradores, que competem para encontrar o próximo bloco por meio de cálculos computacionais intensivos. Já em sistemas Proof-of-Stake (PoS), os validadores depositam (*stake*) o *token* nativo para participar da proposta e validação de blocos, recebendo recompensas pelo bom comportamento e podendo sofrer penalidades (*slashing*) em caso de conduta desonesta ou inatividade. Normalmente, os validadores operam nós completos com alta disponibilidade, boa conectividade e chaves privadas bem protegidas [Bashir 2018]. A atuação coordenada desses diferentes tipos de nós permite

que a rede *blockchain* seja descentralizada, equilibrando segurança, auditabilidade e acessibilidade para todos os participantes.

2.2.3. Mecanismos de Consenso

Entre os mecanismos de consenso empregados em redes blockchain, destacam-se o *Proof-of-Work (PoW)* e o *Proof-of-Stake (PoS)*. Ambos exercem a função essencial de assegurar que apenas blocos válidos sejam incorporados à cadeia, preservando a integridade, a segurança e o caráter descentralizado do sistema.

No *Proof-of-Work (PoW)*, os mineradores competem para resolver um problema criptográfico, cujo objetivo é encontrar um *nonce* que produza um *hash* válido para o bloco. Esse processo exige elevado poder computacional e significativo consumo de energia, o que torna eventuais ataques à rede economicamente inviáveis e incentiva o comportamento honesto dos participantes. O primeiro nó a determinar o *nonce* correto e gerar um *hash* conforme os critérios de dificuldade é recompensado com uma fração da moeda nativa, além das taxas de transação incluídas no bloco. Dessa forma, o PoW atua não apenas como um mecanismo de emissão monetária, mas sobretudo como um sistema de segurança que protege a rede contra manipulações e assegura a integridade da *blockchain* [Bashir 2018]. A Figura 3 ilustra esse processo.

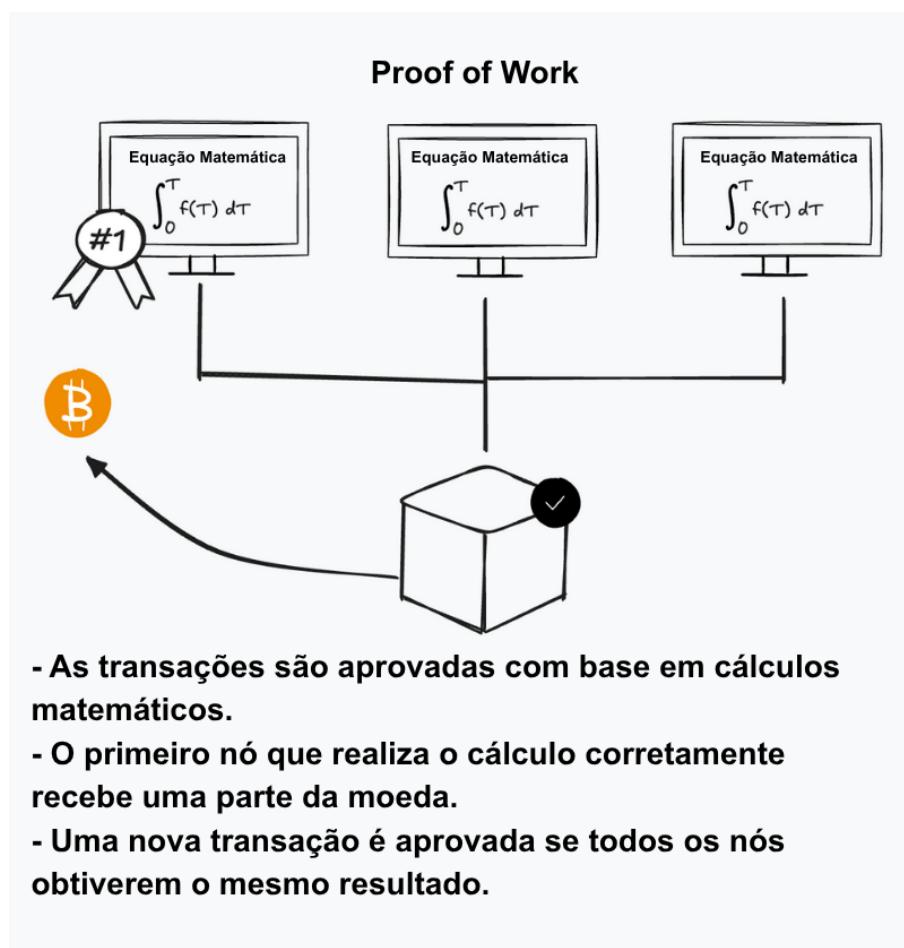


Figura 3. Exemplificação do Proof-Of-Work. Fonte: Adaptado de [Daniels 2025]

No *Proof-of-Stake* (PoS), os validadores são selecionados para propor e validar blocos de acordo com a quantidade de tokens que depositam como garantia (stake). A escolha dos validadores ocorre de forma pseudoaleatória, ponderada pelo valor em stake, de modo a equilibrar oportunidade e segurança. Caso algum participante adote comportamentos desonestos ou tente fraudar o processo de consenso, parte ou a totalidade do valor em stake pode ser confiscada, em um mecanismo denominado *slashing*. As recompensas, por sua vez, são distribuídas de maneira proporcional à quantia bloqueada e ao comportamento honesto dos validadores. Entre as principais vantagens do PoS destacam-se a elevada eficiência energética, a redução das barreiras de entrada e o estímulo à descentralização, contribuindo para um sistema mais inclusivo e sustentável [Antonopoulos and Wood 2018]. A Figura 4 ilustra esse processo.

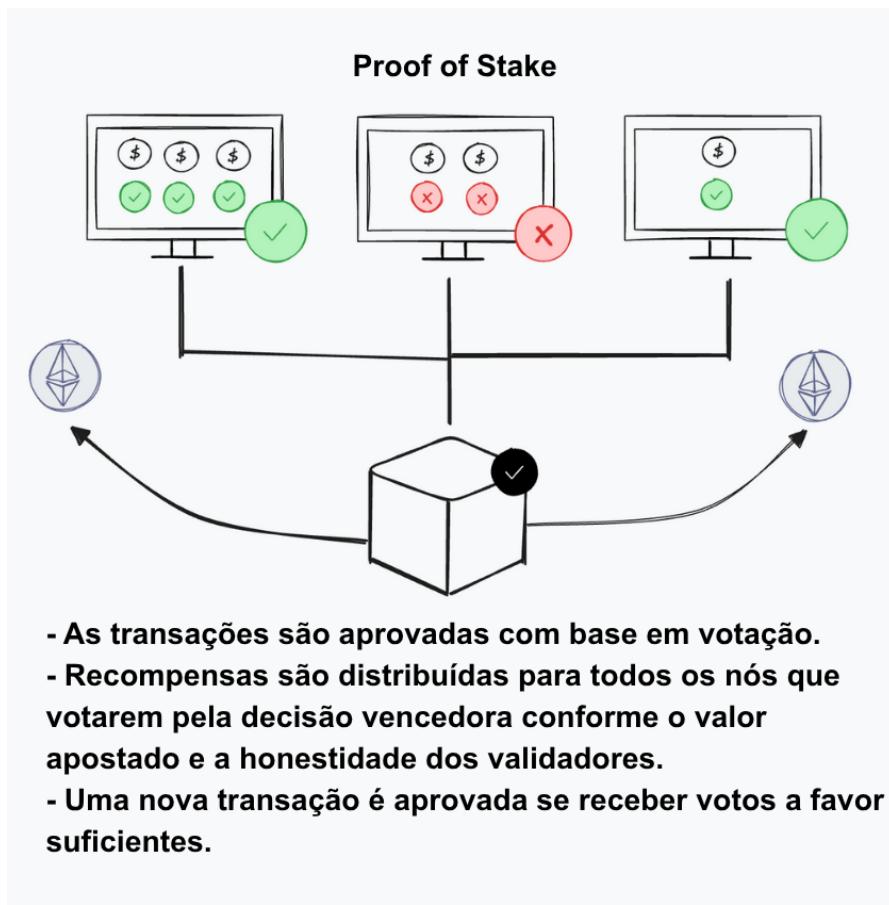


Figura 4. Exemplificação do Proof-Of-Stake. Fonte: Adaptado de [Daniels 2025]

O Ethereum, por exemplo, adotava inicialmente o *Proof-of-Work* (PoW) como mecanismo de consenso, porém migrou para o *Proof-of-Stake* (PoS) com o objetivo de aprimorar a sustentabilidade e a segurança da rede. Essa transição também buscou ampliar o grau de descentralização e reduzir a vulnerabilidade a ataques, consolidando um modelo de consenso mais eficiente e resiliente.

2.2.4. Criptografia e segurança.

Em redes blockchain, a segurança não depende de logins e senhas, mas sim de criptografia. O processo começa com funções de hash, que convertem qualquer dado em um resumo de tamanho fixo e aparentemente aleatório. No ecossistema Ethereum, o algoritmo utilizado para gerar esse hash é o Keccak-256, como mencionado anteriormente [Bashir 2018].

O resultado de uma função de hash é pseudoaleatório, embora seja totalmente determinístico, ou seja, a mesma entrada sempre gera o mesmo resultado, qualquer alteração mínima na entrada produz uma saída completamente diferente e imprevisível. A identidade dos usuários na rede é definida por um par de chaves geradas a partir da curva elíptica secp256k1, uma chave privada, que é basicamente um número aleatório de 256 bits, e sua respectiva chave pública, obtida por multiplicação escalar na curva [Antonopoulos and Wood 2018]. Nesse sistema, quem possui a chave privada controla os fundos associados ao endereço, portanto, perder essa chave significa perder acesso aos ativos.

Para autorizar operações na rede, utilizam-se assinaturas digitais. A transação é serializada, processada por uma função de hash e assinada com a chave privada usando o algoritmo ECDSA. Isso garante que apenas o detentor da chave privada pode autorizar a transação e que os dados não foram alterados. Qualquer nó da rede pode verificar a assinatura utilizando a chave pública, confirmado a autenticidade da transação [Antonopoulos and Wood 2018].

Para simplificar o backup, carteiras modernas adotam a seed phrase (BIP-39), uma sequência mnemônica de 12 ou 24 palavras que codifica entropia de alta qualidade. Essa seed é convertida em uma chave-mestra, da qual podem ser derivadas milhares de chaves privadas e endereços diferentes, seguindo os padrões BIP-32/BIP-44 [Antonopoulos and Wood 2018].

2.3. Princípios de Interação Humano-Computador (IHC)

2.3.1. Definição de Interação Humano-Computador (IHC)

A Interação Humano-Computador (IHC) constitui uma área multidisciplinar que integra conhecimentos da ciência da computação, da psicologia cognitiva e do design, com o objetivo de aprimorar a comunicação e a cooperação entre pessoas e sistemas computacionais. Seu foco central reside no desenvolvimento de interfaces e sistemas que considerem as necessidades, limitações e expectativas dos usuários, promovendo experiências tecnológicas mais intuitivas, eficientes e acessíveis. A IHC ultrapassa o escopo do design de interfaces gráficas, abrangendo tanto a funcionalidade dos sistemas quanto o papel desempenhado por esses artefatos durante a interação humana. Entre seus princípios fundamentais destacam-se a prevenção de erros, a possibilidade de desfazer ações com facilidade, a redução da carga cognitiva de curto prazo e o fornecimento de feedback informativo e contextualizado [Nielsen 1994].

Uma definição amplamente aceita da disciplina é apresentada em documento oficial da ACM SIGCHI, no qual a IHC é descrita como a área dedicada ao design, avaliação e implementação de sistemas computacionais interativos para uso humano, assim como ao

estudo dos fenômenos relacionados a esses sistemas [Hewett et al. 1992]. Essa definição reforça o caráter multidisciplinar da área e destaca que o foco principal recai sobre a natureza da interação, e não apenas sobre sua manifestação visual.

A experiência do usuário (UX) constitui um elemento determinante para a aceitação e a adoção de sistemas interativos. Aspectos afetivos, como expressividade, estética, frustração, presença de agentes e antropomorfismo, influenciam diretamente a percepção de confiança, o engajamento e a fidelização dos usuários. Em contrapartida, uma UX deficiente, marcada por mensagens ambíguas, excesso de etapas ou falta de clareza, tende a gerar insatisfação e abandono do sistema [Preece et al. 2002]. Esses fatores demonstram que atributos emocionais e subjetivos desempenham papel central na forma como as pessoas avaliam e escolhem tecnologias.

De acordo com a norma ISO 9241-11 [for Standardization 2018], a usabilidade corresponde ao grau em que um sistema permite que usuários específicos alcancem objetivos particulares com eficácia, eficiência e satisfação em um contexto de uso definido. A experiência do usuário, por outro lado, envolve percepções, emoções, expectativas e respostas que emergem antes, durante e após a interação. Assim, enquanto a usabilidade se concentra no desempenho e na qualidade da execução das tarefas, a UX engloba dimensões subjetivas mais amplas, como confiança, valor percebido e impacto emocional.

2.3.2. Aspectos Cognitivos e Perceptuais da Intereração

Alguns dos conceitos fundamentais na área de Interação Humano-Computador (IHC) incluem affordance, visibilidade e carga cognitiva. Esses princípios constituem referenciais teóricos amplamente reconhecidos e aplicados para o desenvolvimento de interfaces mais intuitivas, eficientes e alinhadas ao comportamento humano.

O conceito de *affordance* se refere às pistas perceptíveis que indicam ao usuário a possibilidade de realizar uma determinada ação. Exemplos clássicos incluem um botão que sugere a possibilidade de clique ou uma alça que indica ser puxada. Em interfaces digitais, contudo, a simples imitação de objetos físicos não é suficiente, torna-se essencial empregar convenções consolidadas, fornecer feedback claro e estabelecer restrições lógicas ou culturais que orientem o usuário de maneira eficaz [Preece et al. 2002]. A literatura distingue entre *affordance* reais (características físicas que permitem a ação), *affordance* percebidas (interpretações do usuário sobre o que é possível fazer), *affordance* falsas (que sugerem ações que não se concretizam) e *affordance* ocultas (ações possíveis, mas não percebidas) [Norman 2013]. Em ambientes digitais, o papel das convenções e metáforas visuais torna-se ainda mais central para comunicar as *affordance* percebidas de forma clara e consistente.

A visibilidade, por sua vez, diz respeito à clareza com que as funções e o estado da interface são apresentados ao usuário. Uma interface bem projetada deve responder à pergunta “o que posso fazer agora?”, tornando explícitas as ações disponíveis e seus possíveis resultados. Isso requer mapeamentos comprehensíveis entre controles e efeitos, bem como feedback imediato que permita ao usuário reconhecer o impacto de suas escolhas [Preece et al. 2002]. Princípios correlatos incluem o mapeamento natural, entendido como a relação intuitiva entre controles e efeitos do sistema, e o *feedback*, definido como

o retorno imediato e informativo após uma ação [Norman 2013]. Interfaces blockchain que apresentam operações complexas sem feedback adequado violam diretamente esses princípios, prejudicando a compreensão e aumentando o risco de erros.

Por fim, o princípio da carga cognitiva enfatiza a importância de reduzir o esforço mental exigido durante a interação. A psicologia cognitiva demonstra que decisões se tornam mais lentas conforme aumenta a quantidade de opções disponíveis, como aponta a *Lei de Hick* [Hick 1952], enquanto a Lei de Fitts evidencia que tarefas motoras ficam mais lentas à medida que requerem maior precisão [Fitts 1954]. Interfaces que dependem da memorização de etapas, códigos ou comandos tendem a aumentar a probabilidade de erros e a comprometer a usabilidade. Para mitigar esse problema, recomenda-se manter a consistência, de modo que o próprio contexto auxilie a memória, e promover o uso da chamada cognição externa. Sempre que possível, o sistema deve automatizar tarefas mecânicas, como cálculos, verificações ou o rastreamento de progresso, permitindo que o usuário concentre sua atenção nas decisões cognitivamente mais relevantes [Preece et al. 2002].

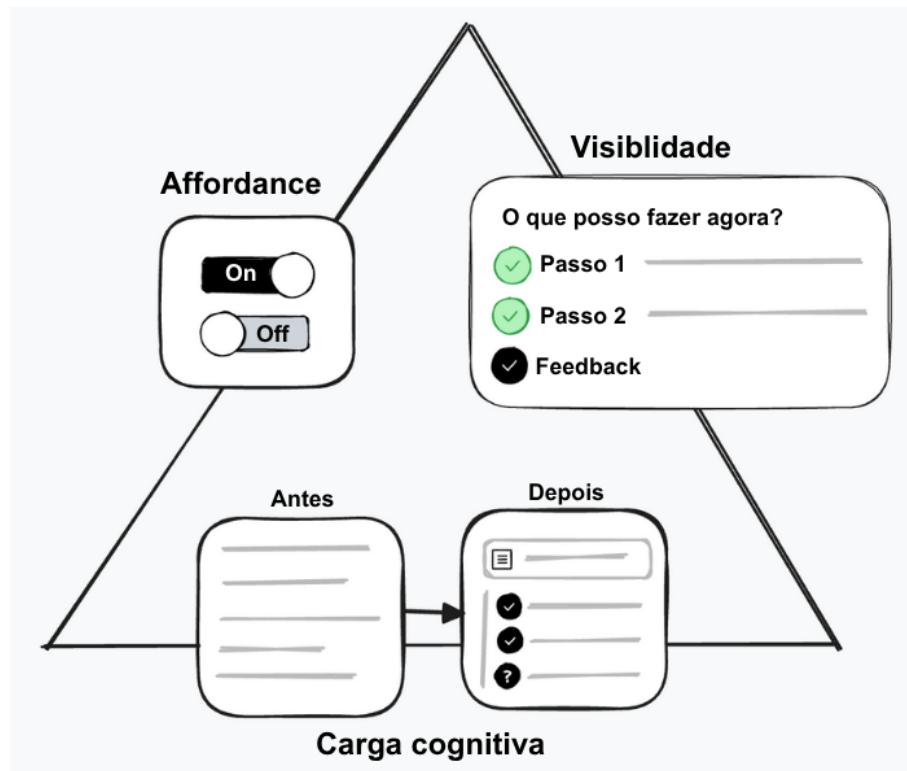


Figura 5. Exemplificação de Affordance, Visibilidade e Carga Cognitiva. Fonte: Autoria Própria.

2.3.3. Segurança Técnica vs. Segurança Percebida

No campo da Interação Humano-Computador (IHC), um dos conceitos centrais é o de segurança usável, que busca integrar a robustez técnica dos sistemas com a clareza e a confiança transmitidas pela interface ao usuário [Preece et al. 2002]. Essa integração é essencial, pois a segurança percebida pelo usuário frequentemente se iguala, em im-

portância, à segurança técnica garantida por mecanismos como criptografia e verificações matemáticas.

A segurança técnica diz respeito à implementação de algoritmos avançados que asseguram a confidencialidade e a integridade dos dados. Contudo, quando a interface não comunica de forma transparente o que ocorre no sistema, um ambiente tecnicamente robusto pode ser interpretado como inseguro. Os usuários tendem a confiar mais no que veem do que nos processos invisíveis executados no back-end [Preece et al. 2002]. Essa discrepância torna-se especialmente crítica em sistemas baseados em blockchain, nos quais interfaces que ocultam informações relevantes ou apresentam mensagens ambíguas podem levar a decisões equivocadas, como autorizar transações maliciosas ou enviar valores para endereços incorretos.

A literatura clássica em segurança computacional indica que mecanismos de proteção devem obedecer ao princípio da aceitabilidade psicológica, segundo o qual controles de segurança devem ser compreensíveis e não impor esforço excessivo ao usuário [Saltzer and Schroeder 1975]. No contexto de sistemas descentralizados e, em particular, de plataformas DeFi, essa distinção entre segurança técnica e segurança percebida torna-se ainda mais relevante, uma vez que erros podem resultar em perdas financeiras irreversíveis. Interfaces que não traduzem operações complexas em linguagem acessível podem induzir usuários a conceder permissões amplas sem compreender plenamente as consequências [Nielsen 1994]. Entre os problemas recorrentes estão endereços truncados, pop-ups confusos, tokens falsos e propostas de governança ambíguas, todos explorando falhas de design de experiência do usuário (UX).

Além disso, a cultura de autocustódia característica do ecossistema blockchain exige que interfaces forneçam orientações claras sobre práticas de backup, uso de *seed phrases* e assinaturas fora da cadeia. Qualquer ambiguidade nesses processos amplia a superfície de risco, favorecendo ataques como *phishing*. Paradoxalmente, quanto mais robusta e *permissionless* a infraestrutura, maior a probabilidade de erros humanos mediados por uma experiência de uso inadequada.

Estudos empíricos pioneiros demonstraram que usuários não conseguem operar sistemas seguros quando as interfaces são mal projetadas, inaugurando o campo da segurança usável [Whitten and Tygar 1999]. A conclusão central, de que a segurança depende diretamente da qualidade da interação, permanece válida e é particularmente crítica no ecossistema DeFi, no qual decisões equivocadas são irreversíveis.

A confiança do usuário pode ser analisada com base no modelo tripartite proposto por Mayer, Davis e Schoorman [Mayer et al. 1995], composto pelos elementos de competência percebida, benevolência e integridade. Em interfaces blockchain, fatores como textos vagos, endereços truncados e permissões pouco claras comprometem diretamente esses três componentes, reduzindo a disposição do usuário em confiar no sistema.

2.3.4. Modelos de Avaliação de Interface

A avaliação de interfaces desempenha um papel fundamental para assegurar que sistemas interativos sejam eficazes, eficientes e satisfatórios para seus usuários [Nielsen 1994]. Diferentes métodos podem ser empregados ao longo do ciclo de desenvolvimento, com-

binando análises baseadas em princípios, modelos cognitivos e experimentação com usuários reais. Essa combinação permite identificar tanto problemas estruturais da interface quanto dificuldades perceptivas ou cognitivas que emergem durante o uso.

Um dos modelos mais conhecidos é o conjunto das dez heurísticas de usabilidade propostas por Jakob Nielsen [Nielsen 1994], que servem como diretrizes amplamente reconhecidas para orientar a análise de problemas recorrentes. Entre os princípios avaliados estão a visibilidade do status do sistema, a correspondência entre sistema e mundo real, o controle e liberdade do usuário, a consistência, a prevenção de erros, a redução da carga de memória, a flexibilidade de uso, o design minimalista e a clareza das mensagens de ajuda. Essas heurísticas são especialmente úteis para identificar falhas conceituais em interfaces críticas, como carteiras digitais ou plataformas DeFi, nas quais decisões equivocadas podem resultar em perdas irreversíveis.

Complementarmente, Ben Shneiderman apresentou as *Eight Golden Rules of Interface Design* [Shneiderman et al. 2009], que enfatizam a importância da consistência, do fornecimento de feedback informativo, da oferta de ações de desfazer (*undo*) e refazer (*redo*), da prevenção de erros e da redução da carga cognitiva. Esses princípios reforçam a necessidade de interfaces transparentes e orientadas ao usuário, especialmente relevantes em ecossistemas blockchain, onde operações são frequentemente apresentadas em termos técnicos e exigem alto grau de confiança.

Os testes de usabilidade com usuários reais complementam esses modelos ao fornecer evidências empíricas sobre como pessoas interagem com a interface. Tais testes permitem observar diretamente dificuldades, estratégias, dúvidas e erros, além de coletar métricas objetivas como tempo de execução, taxa de sucesso e número de erros [Preece et al. 2002]. Contudo, para compreender de forma mais profunda o esforço cognitivo envolvido durante a interação, instrumentos padronizados também podem ser empregados.

Entre eles, destaca-se o NASA Task Load Index (NASA-TLX) [Hart and Staveland 1988], uma das ferramentas mais consolidadas para mensurar a carga de trabalho percebida pelo usuário durante a realização de uma tarefa. O método avalia seis dimensões: demanda mental, demanda física, demanda temporal, desempenho percebido, esforço e frustração. Ao aplicar o NASA-TLX, torna-se possível identificar pontos do fluxo que geram sobrecarga cognitiva, insegurança ou frustração, fornecendo insights que heurísticas tradicionais muitas vezes não capturam.

Dessa forma, a combinação entre avaliações heurísticas, modelos cognitivos e instrumentos empíricos constitui um arcabouço robusto para analisar e aprimorar interfaces. Essa abordagem integrada é particularmente valiosa no contexto das aplicações blockchain, nas quais clareza, confiança e redução da carga mental são determinantes para evitar erros críticos e fomentar a adoção em larga escala.

3. Revisão da literatura

A adoção de aplicações de finanças descentralizadas (*DeFi*) esbarra em dois obstáculos recorrentes, a barreiras de usabilidade, que dificultam a formação de modelos mentais estáveis por usuários não especialistas [Jang et al. 2020], e a fadiga de operações, causada por sequências longas e pouco transparentes de passos, aprovações e confirmações

[Saldivar et al. 2023]. A literatura recente em HCI para sistemas distribuídos argumenta que reduzir fricções instrumentais é fundamental para converter a curiosidade inicial em retenção e uso recorrente. Tecnologias emergentes dificilmente alcançam seu potencial sem um foco consistente em experiência do usuário, o que exige equilibrar inovação tecnológica com estratégias de design centrado no usuário [Glomann et al. 2019].

3.1. Barreiras de Usabilidade em DeFi

Ainda antes da realização de qualquer transação, o processo de onboarding já se configura como uma barreira significativa à adoção, afastando uma parcela relevante dos potenciais usuários. A configuração inicial frequentemente se revela complexa e carente de orientações claras, desde a criação de carteiras até a execução de operações básicas. Essa dificuldade se intensifica diante da necessidade de troca de dispositivo ou da perda da *seed phrase*, uma vez que muitos usuários não compreendem sua importância nem conhecem os procedimentos adequados para recuperação [Voskobojnikov et al. 2021]. A exigência de anotar, armazenar e posteriormente localizar uma sequência de palavras representa um ponto de fricção recorrente entre usuários e aplicações descentralizadas [Moniruzzaman et al. 2020].

Outro fator relevante é o desalinhamento entre o modelo mental herdado do sistema bancário tradicional e o funcionamento da blockchain. Usuários frequentemente apresentam dificuldades para compreender taxas, endereços e a irreversibilidade das operações. Muitos acreditam que as taxas são definidas pelo aplicativo, quando na realidade são determinadas pela rede, e esperam a possibilidade de cancelar ou reverter transações, expectativa que não se aplica ao contexto descentralizado [Voskobojnikov et al. 2021]. Esse descompasso é intensificado por interfaces pouco intuitivas, nas quais ações essenciais permanecem ocultas, mensagens de erro não contribuem para a resolução de problemas e, para usuários sem domínio do inglês, traduções incompletas ou inadequadas comprometem o entendimento em momentos críticos [Froehlich et al. 2021].

Paralelamente, o receio de perda de fundos é amplificado pela comunicação pouco clara durante transações, backups ou restaurações. A gestão de chaves, em especial, permanece pouco intuitiva e sujeita a erros, uma vez que a distinção entre chaves públicas e privadas é abstrata para a maioria dos usuários [Moniruzzaman et al. 2020].

Em síntese, é observada uma cadeia de fricções que se retroalimentam. As instabilidades técnicas minam a confiança inicial, dificuldade de onboarding e de recuperação de conta dificultam o retorno após contratemplos e compreensão insuficiente de conceitos fundamentais torna operações rotineiras arriscadas. Esse conjunto de fatores não apenas eleva o custo mental de cada ação, mas também transformam pequenos obstáculos em desistências definitivas, o que restringe o funil de adoção e impede que os usuários desenvolvam a fluência operacional necessária para sua permanência no ecossistema.

3.2. Fadiga de Operações

Pesquisas recentes apontam que usuários relatam frustração e fadiga ao lidar com sistemas que exigem muitos passos para realizar operações simples. A necessidade de múltiplas confirmações e verificações de segurança pode tornar o processo cansativo, especialmente para iniciantes [Albayati et al. 2021]. Além disso, sistemas que exigem navegação por

múltiplas plataformas e possuem etapas pouco claras ou repetitivas geram ansiedade e preocupação nos usuários [Si et al. 2024]. A falta entendimento sobre o significado de cada etapa contribui para a sensação de insegurança e sobrecarga cognitiva. Para lidar com essa complexidade, alguns usuários inclusive adotam estratégias, como testar operações com valores pequenos antes de realizar transações maiores, para assim garantir que não irão se perder durante o processo e realizar transferências erradas [Si et al. 2024].

Outro ponto de preocupação comum entre usuários novatos é o tempo de espera entre a realização e o registro de uma transação no blockchain. Esse intervalo costuma ser maior do que em sistemas centralizados e, frequentemente, os usuários não recebem informações claras sobre quanto tempo devem aguardar. Como resultado, surgem insegurança, sensação de erro e abandono da tarefa [Jang et al. 2020]. Problemas de usabilidade também são relatados em relação à ausência de explicações sobre o que está acontecendo, o que cada botão faz ou por que certas ações são necessárias. Isso obriga o usuário a repetir etapas e buscar informações por tentativa e erro, aumentando o risco de desistência [Jang et al. 2020].

Esses desafios se confirmam em testes de usabilidade realizados com aplicações que utilizam a rede *EVM*. Mais de 70% do tempo total dos participantes foi consumido por etapas relacionadas ao uso do Metamask e da rede Ethereum, especialmente durante a instalação da carteira e criação da conta, momentos marcados por bloqueios e confusão [Saldivar et al. 2023]. Além disso, os termos técnicos apresentados nas janelas de transação, como *gas fee*, *gas price* e *gas limit*, foram considerados excessivamente complexos para pessoas com pouca experiência em blockchain, gerando ainda mais insegurança e dificultando o processo [Saldivar et al. 2023].

3.3. Padrões de Design para Reduzir Fricção

Estudos indicam que a redução da fricção na utilização de aplicações descentralizadas depende do alinhamento dos fluxos operacionais com modelos mentais já consolidados pelos usuários em bancos e corretoras, por meio da reutilização de termos e ícones familiares. Por exemplo, ao tratar a carteira como uma *conta*, com ações de *ver saldo* e *transferir*, evita-se a necessidade de reaprendizagem de operações básicas, promovendo uma navegação mais previsível [Jang et al. 2020]. Esta coesão com práticas tradicionais deve ser refletida em funcionalidades críticas, como mecanismos para lidar com transações pendentes, aproximando a experiência do que o usuário espera ao “alterar” uma operação bancária. Da mesma forma, opções de backup em nuvem, com a seed-phrase cifrada e armazenada em serviço escolhido pelo usuário, reduzem o risco percebido na recuperação de acesso e replicam padrões já aceitos em aplicativos convencionais [Voskobojnikov et al. 2021].

Tal alinhamento é potencializado quando o onboarding inicial é transparente, guiando o usuário passo a passo na instalação, criação de conta e execução da primeira operação. Além de reduzir a incerteza das etapas iniciais, esse percurso pode ser simplificado com a integração de carteiras em um modelo *custodial-like*, no qual a aplicação abstrai a gestão de chaves e os detalhes de transação para iniciantes, sem impedir que usuários experientes conectem suas próprias carteiras e acessem configurações avançadas [Saldivar et al. 2023]. A interface pode ser adaptada ao nível de proficiência do usuário em que são feitos perfis simplificados para iniciantes, focados em tarefas essenciais, e

visões avançadas para usuários experientes, incluindo ajuste fino de taxas e importação de chaves [Voskobojnikov et al. 2021]. Essa personalização deve ser acompanhada de orientação adequada, como tutoriais curtos e ambientes de teste, os quais auxiliam na explicação de conceitos como irreversibilidade, taxas e importância do backup. Por fim, a comunicação clara sobre sincronização, por exemplo, quando o saldo ainda não refletiu o último bloco, e sobre limitações do protocolo, como a impossibilidade de “cancelar” uma transação confirmada, previnem expectativas irrealistas e fortalecem a confiança [Voskobojnikov et al. 2021].

Paralelamente, é fundamental a diminuição do cansaço operacional imposto por confirmações repetitivas. Parte das autorizações pode ser agregada ou mediada por resumos de intenção, reduzindo cliques e diálogos redundantes sem comprometer a verificabilidade e a segurança [Saldivar et al. 2023]. No nível micro da interação, a simplificação dos fluxos atua de maneira decisiva. Operações comuns devem exigir o mínimo de etapas possível com confirmações apenas quando agregarem valor real à segurança ou à compreensão do risco [Si et al. 2024]. Cada etapa precisa fornecer feedback imediato e mensagens de erro devem sugerir ações corretivas, evitando a apresentação exclusiva de códigos técnicos [Si et al. 2024]. A transparência sobre o propósito de cada etapa e o que está sendo aprovado pode ser promovida por meio de resumos curtos e *tooltip* nos pontos críticos, como permissões de gasto e assinaturas de contratos, reduzindo ambiguidades sem sobrecarregar a interface [Si et al. 2024].

Por fim, a automação e o preenchimento inteligente desempenham papel relevante na aceleração de tarefas repetitivas e na redução do esforço exigido dos usuários, como a sugestão automática de endereços frequentemente utilizados e o resgate das últimas preferências configuradas. Simultaneamente, mecanismos de autenticação de baixo atrito, como biometria e PIN, contribuem para a preservação da segurança sem transformar cada ação em um obstáculo adicional. O princípio orientador nessas práticas é garantir que a proteção seja devidamente explicada e percebida pelo usuário, em vez de simplesmente imposta [Si et al. 2024].

4. Metodologia

Neste estudo, buscamos investigar formas de eliminar barreiras de usabilidade e reduzir a fadiga operacional em aplicações *DeFi* por meio de um teste A/B conduzido com duas versões de uma mesma plataforma. Ambas as versões oferecem funcionalidades idênticas, diferenciando-se apenas nos fundamentos de usabilidade aplicados. Em revisões recentes, foi apontado que, dentre as principais operações e categorias de serviços no âmbito de *DeFi*, se destacam stablecoins, troca de ativos e transferências. Segundo o relatório da *Wharton School* (2021), essas operações são centrais para a experiência do usuário e representam as principais interações típicas em ambientes *DeFi* [Gogel et al. 2021]. Por esse motivo, o desenvolvimento de *features* da plataforma para o cenário experimental se justifica por reunir características representativas dessas interações. Na plataforma desenvolvida, os usuários podem realizar o depósito e retirada de ativos, trocar ativos, enviar a outros usuários, se alinhando às práticas observadas nos principais protocolos DeFi descritos na literatura.

Com base nesse desenho experimental, foram estabelecidas as seguintes hipóteses de pesquisa:

1. A versão com princípios de usabilidade aumenta a satisfação e a carga cognitiva percebida em comparação à versão sem tais princípios;
2. O ganho proporcionado pela versão com princípios heurísticos é mais pronunciado entre usuários novatos do que entre usuários experientes;
3. A versão baseada em princípios heurísticos reduz a taxa de abandono de tarefas em comparação à versão sem esses fundamentos;

Para operacionalizar a comparação, as duas versões da aplicação foram implementadas com fundamentos distintos de usabilidade. A primeira seguiu princípios heurísticos consolidados, priorizando *feedback* imediato, terminologia clara e consistente, minimização da carga cognitiva e prevenção de erros [Nielsen and Molich 1990]. O fluxo de navegação foi projetado para ser intuitivo, com rótulos descritivos, mensagens de confirmação legíveis e indicadores visuais de progresso em cada etapa. Em contraste, a segunda versão foi construída sem a aplicação desses princípios, os rótulos utilizam linguagem técnica, há escassez de *feedback* visual, pouca consistência entre elementos da interface e mensagens genéricas de erro ou confirmação. Essa configuração representa um cenário menos amigável, próximo das barreiras de usabilidade ainda comuns em aplicações *DeFi* emergentes, sem comprometer a segurança ou a funcionalidade essencial do sistema.

4.1. Métricas do estudo

Para a construção das métricas deste estudo, serão utilizadas medidas objetivas e subjetivas para obtenção de dados quantitativos. Em pesquisas de usabilidade, métricas objetivas como tempo de execução, número de cliques e taxa de sucesso permitem quantificar a eficiência e a eficácia do sistema. No entanto, essas métricas não capturam aspectos subjetivos da experiência do usuário, como satisfação, carga cognitiva ou reações emocionais [Foster et al. 2009][ASSILA et al. 2016]. Nesse contexto, estudos defendem que, além das métricas objetivas de desempenho, é fundamental compreender as percepções dos usuários, especialmente considerando o caráter inovador, tecnicamente complexo e cognitivamente exigente desses sistemas [Saldivar et al. 2023].

4.1.1. Métricas objetivas de desempenho

Os dados estatísticos selecionados para este estudo abrangem diferentes dimensões da experiência do usuário. Inicialmente, o tempo de sessão foi adotado por ser uma métrica clássica de eficiência de normas internacionais de usabilidade, como a *ISO 9241-11*, a eficiência é expressa pelo tempo necessário para completar tarefas [?], e *Nielsen* destaca o tempo como indicador fundamental da intuitividade da interface [Nielsen 1994]. Complementando essa perspectiva, a taxa de sucesso foi incluída como métrica de eficácia, pois representa diretamente a capacidade dos usuários de alcançar seus objetivos no sistema, sendo considerada uma das formas mais objetivas de verificar se a aplicação cumpre sua função essencial [Nielsen 1994][Rubin and Chisnell 2008]. Já o número de cliques reflete a carga de trabalho operacional exigida para concluir uma ação. Segundo *Shneiderman*, a redução de interações desnecessárias é fundamental para tornar sistemas mais eficientes e menos cansativos [Shneiderman and Plaisant 2004], de modo que interfaces que demandam menos cliques tendem a exigir menor esforço cognitivo e físico do usuário.

Além dessas métricas, a taxa de abandono foi utilizada como um importante indicador de frustração e de falhas no fluxo de interação, permitindo identificar pontos críticos em que a interface pode gerar desistência, um fenômeno frequentemente associado a barreiras cognitivas [Sauro and Lewis 2012]. Por fim, o mapa de calor do cursor foi adotado como complemento às métricas tradicionais, possibilitando a análise visual de padrões de atenção e áreas de confusão, permitindo compreender não apenas o resultado final das interações, mas também o percurso percorrido pelos usuários ao longo das tarefas.

Tabela 1. Métricas objetivas utilizadas no estudo

Métrica	Descrição	Forma de registro
Tempo de sessão	Intervalo total entre o início e a finalização das tarefas	Capturado automaticamente por logs de interação
Taxa de sucesso	Proporção de participantes que concluíram corretamente cada tarefa	Classificação binária (sucesso/falha)
Número de cliques	Total de interações necessárias até a conclusão da ação	Contagem automática de eventos de clique

4.1.2. Métricas subjetivas de experiência

Para complementar a análise objetiva do desempenho dos participantes, serão coletados também dados subjetivos referentes à carga cognitiva percebida durante a interação com as interfaces. Para isso, será adotado o NASA Task Load Index (NASA-TLX), um dos instrumentos mais difundidos e validados para mensuração de carga de trabalho mental em sistemas interativos, ambientes complexos e tarefas operacionais [Hart and Staveland 1988][Hart 2006]. O NASA-TLX se destaca por sua capacidade de captar múltiplas dimensões da carga cognitiva, tornando-o especialmente adequado para contextos como aplicações *DeFi*, nas quais os usuários frequentemente lidam com tarefas de tomada de decisão, análise de risco e execução de operações em sequência.

Neste estudo, será utilizada a variante RAW do NASA-TLX, um procedimento simplificado que dispensa a etapa de ponderação dos pesos entre dimensões [Byers et al. 1989]. Em vez de realizar comparações pareadas para definir pesos individuais, cada participante responderá a um único formulário por interação realizada. As seis dimensões avaliadas serão pontuadas diretamente em escalas de 0 a 100. Posteriormente, a pontuação final é obtida por meio da média aritmética das seis dimensões, resultando em um escore único representativo da carga cognitiva percebida. A literatura recente aponta que a versão RAW apresenta confiabilidade semelhante à versão ponderada, sendo amplamente recomendada por sua simplicidade e boa validade psicométrica [Grier 2015].

A Tabela 2 apresenta os seis itens avaliados no instrumento NASA-TLX (RAW), adaptados para o formato de questionário aplicado aos participantes.

Tabela 2. Dimensões avaliadas pelo questionário NASA-TLX (RAW)

Nº	Categoría	Pergunta
1	Exigência Mental (<i>Mental Demand</i>)	Quanto esforço mental e concentração foram necessários para usar a plataforma? (Ex.: pensar, decidir, lembrar, compreender o que estava acontecendo).
2	Exigência Física (<i>Physical Demand</i>)	Quanto esforço físico (cliques, digitação, movimentação) foi necessário para completar as ações?
3	Exigência Temporal (<i>Temporal Demand</i>)	Quão pressionado(a) pelo tempo você se sentiu durante o uso da plataforma?
4	Desempenho (<i>Performance</i>)	Quão satisfeito(a) você ficou com o seu desempenho geral na plataforma? (0 = Fracasso total / 100 = Sucesso total)
5	Esforço (<i>Effort</i>)	Quanto esforço total você precisou fazer para usar a plataforma com sucesso?
6	Nível de Frustração (<i>Frustration</i>)	Quão irritado(a), inseguro(a), estressado(a) ou frustrado(a) você se sentiu durante o uso da plataforma?

4.2. Estratificação dos participantes

Para a condução do experimento, os participantes serão distribuídos em três grupos distintos de acordo com seu nível de familiaridade com sistemas *DeFi*, iniciantes, intermediários e avançados. Essa estratificação permite observar com maior precisão como diferentes graus de experiência influenciam a percepção de usabilidade, especialmente em um domínio no qual o conhecimento prévio pode afetar diretamente a fluidez da interação e a carga cognitiva percebida [Nielsen 1994]. Os iniciantes serão caracterizados como usuários que não possuem ou possuem muito pouca experiência com aplicações *DeFi*. Os intermediários incluem participantes que já tiveram algum contato prévio com plataformas descentralizadas, ainda que de forma esporádica ou limitada. Por fim, o grupo avançado será composto por usuários que trabalham com esse tipo de tecnologia ou que fazem uso frequente e aprofundado de aplicações *DeFi*. A classificação dos participantes será realizada por meio de autodeclaração do nível de familiaridade com tecnologias blockchain, abordagem amplamente adotada em estudos de usabilidade e Interação Humano-Computador quando o grau de expertise é subjetivo, contextual e de difícil mensuração objetiva [Nielsen 1994, Preece et al. 2002]. A literatura da área reconhece que a experiência do usuário deve ser compreendida em relação às tarefas e ao domínio específico do sistema avaliado, sendo a autopercepção um indicador válido para a segmentação inicial de perfis [McGrenere and Ho 2000, Tullis and Albert 2008].

Além disso, a ordem de exposição às versões com maior usabilidade e com menor usabilidade foi alternada de forma aleatória entre os participantes, com o objetivo de mitigar o viés de aprendizado. Estudos indicam que, quando os participantes utilizam primeiro uma versão, podem adquirir familiaridade com as tarefas, o que tende a influenciar seu desempenho na segunda interação [Rubin and Chisnell 2008]. A alternância aleatória da versão inicial reduz esse efeito, aumentando a confiabilidade da comparação

entre condições. A Figura 5 ilustra o fluxo completo de aplicação dos testes, incluindo a etapa de categorização dos participantes, a execução das tarefas padronizadas em ambas as plataformas e a coleta das percepções subjetivas por meio do formulário NASA-TLX (RAW). Dessa forma, além de permitir comparações diretas entre os sistemas avaliados, a metodologia também possibilita investigar se o nível de experiência dos usuários impacta significativamente a percepção de carga cognitiva em ambientes financeiros descentralizados.

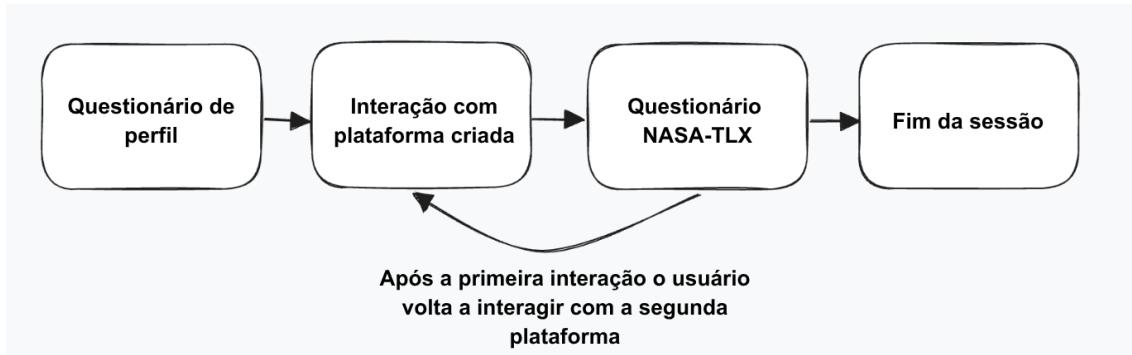


Figura 6. Fluxo de aplicação dos testes. Fonte: Autoria própria

4.3. Procedimento do teste de usabilidade

A literatura ressalta que a definição de tarefas específicas e consistentes é essencial para a validade dos testes de usabilidade [Group 2023], e que procedimentos padronizados são necessários para permitir comparações sistemáticas entre diferentes condições experimentais [Bastien 2010]. Dessa forma, para a execução dos testes, foi estabelecido um conjunto de tarefas que todos os participantes deveriam realizar em ambas as plataformas. Essas tarefas foram elaboradas de modo a contemplar as principais funcionalidades do sistema e a mimetizar o comportamento típico de usuários em plataformas *DeFi*, assegurando a padronização entre os testes e possibilitando a obtenção de resultados consistentes para a análise.

Além disso, a escolha das tarefas buscou refletir cenários de uso realistas, incorporando operações centrais identificadas na literatura como fundamentais para a experiência de interação em ambientes *DeFi*, tais como depósito e retirada de ativos, troca de tokens, transferências entre usuários e operações de empréstimo [Gogel et al. 2021]. A inclusão desses fluxos garante não apenas a avaliação da usabilidade de funcionalidades representativas, mas também a possibilidade de analisar como diferentes níveis de experiência dos participantes impactam sua interação com o sistema.

5. Desenvolvimento dos Protótipos

5.1. Smart Contracts

Para a representação das funcionalidades descritas na metodologia, foram desenvolvidos três contratos inteligentes principais. O primeiro deles, denominado *Vault*, é responsável pela gestão dos tokens e constitui o núcleo operacional do sistema. Os demais contratos, *TokenFactory* e *BaseToken*, foram criados para permitir a geração dinâmica de novos tokens e a definição de suas propriedades fundamentais. O desenvolvimento foi conduzido

Tabela 3. Roteiro de tarefas definidas para o teste de usabilidade

Nº	Tarefa	Explicação
1	Depositar ativos	Testar a facilidade com que o usuário consegue adicionar stablecoins à plataforma, verificando se entende os passos e confirmações necessárias.
2	Realizar uma troca de ativos (<i>swap</i>)	Avaliar a intuitividade da interface ao permitir converter uma <i>stablecoin</i> em outro <i>token</i> , incluindo clareza das taxas e resultados da operação.
3	Enviar ativos a outro usuário	Testar a clareza e segurança percebida no processo de transferência de <i>tokens</i> entre usuários, incluindo confirmação da transação.
4	Retirar ativos da plataforma	Avaliar a facilidade e confiança do usuário ao realizar a retirada de fundos, garantindo que o fluxo seja transparente e direto.

na linguagem *Solidity*, dentro do *framework Foundry*, que oferece um ambiente completo amplamente utilizada pelo mercado de desenvolvedores blockchain para a escrita, teste e implantação de contratos inteligentes. Essa ferramenta permite a execução de testes automatizados e simulações em ambiente local, garantindo maior controle sobre os experimentos e replicabilidade dos resultados.

5.1.1. Tokens do Sistema Simulado

Os tokens simulados para interação com o sistema seguiram o padrão *ERC20*, uma especificação formal adotada na rede Ethereum e em todas as blockchains compatíveis com a *EVM* para garantir interoperabilidade entre tokens fungíveis. Ele define um conjunto mínimo de funções e eventos que todo contrato deve implementar para ser reconhecido como um token compatível. Entre as principais funções estão *balanceOf*, que retorna o saldo de um endereço, *transfer*, que executa a transferência de tokens *approve* e *transferFrom*, que controlam permissões de movimentação de fundos por terceiros e *allowance* que consulta os limites de aprovação. Além disso, os eventos *Transfer* e *Approval* garantem rastreabilidade completa de todas as movimentações, facilitando a auditoria e o monitoramento por aplicações descentralizadas (*dApps*) e exploradores de blocos.

No sistema desenvolvido, foram criados dois tokens para simulação, o *USD*, representando uma *stablecoin* fictícia ancorada ao dólar americano, e o *WBTC*, uma versão tokenizada do Bitcoin. O token *USD* segue a convenção de possuir 18 casas decimais, equivalente ao padrão utilizado pelo Ether e pela maioria dos tokens *ERC20*. Já o *WBTC* adota 8 casas decimais, refletindo a granularidade original do Bitcoin, que opera com valores até um *satoshi* (1/100.000.000 de BTC). Essa diferença é importante porque implica diretamente as operações de conversão realizadas pelo contrato *Vault*, exigindo normalização das unidades numéricas antes de qualquer cálculo matemático entre diferentes ativos.

5.1.2. Contrato *Vault*

As operações fundamentais do contrato *Vault* consistem em permitir que os usuários realizem depósitos, saques, transferências internas e conversões entre diferentes tokens de maneira segura e eficiente. Essas funcionalidades constituem o núcleo da interação entre os agentes e o sistema, permitindo o gerenciamento de ativos digitais de forma descentralizada e auditável.

O processo de depósito é responsável por registrar a entrada de valores na carteira interna de cada usuário dentro do contrato. A função `deposit()` foi desenvolvida de modo a aceitar tokens ERC20, garantindo flexibilidade e interoperabilidade. Quando um depósito é realizado com o token nativo, o valor enviado é diretamente creditado no mapeamento de saldos do usuário, conforme ilustrado no trecho:

```
balanceOfToken[msg.sender] [NATIVE] += msg.value;  
emit Deposit(msg.sender, NATIVE, msg.value);
```

Após o registro do depósito, o contrato emite o evento `Deposit`, que permite o acompanhamento das transações por exploradores de blocos e ferramentas de monitoramento. Esse mecanismo de emissão de eventos é fundamental para assegurar a transparência e a rastreabilidade das operações dentro de um ambiente descentralizado. O processo de saque, por sua vez, realiza o procedimento inverso, debita o saldo registrado do usuário e transfere o valor correspondente para o endereço informado, emitindo o evento `Withdraw`. Essa simetria entre depósito e saque garante consistência na contabilidade interna do contrato e reforça a confiabilidade do sistema como custodiante digital.

Além dessas funcionalidades, o contrato também possibilita a movimentação de ativos entre usuários por meio da função `transferInternal()`, que realiza transferências internas sem a necessidade de gerar uma nova transação on-chain. O funcionamento dessa função é feita de forma que o saldo do remetente é decrementado e o do destinatário é incrementado no registro interno, conforme mostrado abaixo:

```
balanceOfToken[msg.sender] [token] -= amount;  
balanceOfToken[to] [token] += amount;
```

Essa forma de atualização de saldos mantém a integridade do sistema, evitando inconsistências e garantindo que todas as movimentações sejam refletidas de forma instantânea e rastreável. Além disso, o modelo de transferências internas reforça o caráter de eficiência do contrato, permitindo que operações lógicas múltiplas ocorram antes de serem efetivamente registradas na blockchain, o que é essencial para aplicações que requerem alta performance.

Entre as funcionalidades mais complexas do contrato *Vault* se encontra a operação de conversão entre tokens, implementada na função `swap()`. Essa função possibilita que os usuários convertam um ativo digital em outro com base em valores de mercado atualizados, tornando o contrato uma peça fundamental de um sistema de negociação automatizado. Para obter essas cotações de forma segura e confiável, o contrato integra-se com a rede *Chainlink*, que fornece *price feeds* descentralizados.

A *Chainlink* atua como um oráculo, isto é, um serviço intermediário que conecta o ambiente blockchain, determinístico e fechado, ao mundo externo, de onde extrai

informações de mercado verificáveis. No caso dos *price feeds*, a *Chainlink* agrupa dados de diversas corretoras e fontes de liquidez, garantindo que o preço utilizado pelo contrato represente um valor de consenso global e não dependa de uma única fonte centralizada. Essa arquitetura aumenta a confiabilidade do sistema e evita vulnerabilidades associadas a manipulações de preço, conhecidas como *oracle manipulation attacks* [?].

Dentro do contrato, a função `setTokenPriceFeed()` permite ao desenvolvedor associar cada token a um endereço específico de *price feed* fornecido pela *Chainlink*, o que inclui tanto o valor atual do ativo quanto a quantidade de casas decimais utilizada pelo oráculo. Durante a execução de uma conversão, a função `getTokenPrice()` é chamada para obter as informações de ambos os tokens envolvidos na operação, o token vendido (`sellToken`) e o token comprado (`buyToken`). A quantidade resultante da conversão é calculada pela expressão:

$$amountOut = \frac{quantity \times sellTokenPrice \times 10^{decimals_{buy}}}{buyTokenPrice \times 10^{decimals_{sell}}}$$

Essa fórmula tem como objetivo uniformizar as bases decimais dos tokens antes da conversão, garantindo que todas as operações matemáticas sejam realizadas sobre unidades compatíveis. Esse ajuste é necessário porque, conforme o padrão *ERC20*, cada token pode definir seu próprio número de casas decimais, refletindo diferentes granularidades de representação de valor.

Durante os testes realizados, foram utilizados dois tokens criados especificamente para o sistema, o USD, que representa uma *stablecoin* com 18 casas decimais, e o WBTC, um token que replica o Bitcoin e adota 8 casas decimais. Essa diferença numérica exige uma normalização de escala, multiplicando ou dividindo pelo fator $10^{decimals}$, de modo que todos os valores fiquem na mesma base antes da conversão. Por exemplo, para converter 1 unidade de WBTC em USD, o contrato precisa primeiro ajustar a granularidade dos valores, uma vez que 1 WBTC (8 decimais) equivale a 10^{10} unidades de USD (18 decimais). Esse mecanismo assegura precisão e coerência matemática no processo de conversão entre ativos com diferentes resoluções numéricas.

Após a padronização, o contrato realiza o cálculo da quantidade resultante, verifica a disponibilidade de liquidez no *pool* correspondente e, caso haja saldo suficiente, efetiva a operação debitando o token de origem e creditando o token de destino. O evento `Swap` é então emitido, registrando os detalhes da operação de forma imutável na blockchain. Esse fluxo é inteiramente determinístico, transparente e auditável, o que reforça a segurança e previsibilidade do sistema.

5.1.3. Contrato *TokenFactory* e *BaseToken*

O contrato *TokenFactory* foi desenvolvido com o propósito de permitir a criação dinâmica de novos tokens compatíveis com o padrão *ERC20*, eliminando a necessidade de replicar manualmente o código-fonte para cada ativo emitido. Sua concepção baseia-se no padrão de projeto *factory pattern*, no qual um contrato mestre é responsável por gerar instâncias de outros contratos a partir de parâmetros específicos. Essa abordagem confere ao sistema elevada flexibilidade e extensibilidade, permitindo a criação de novos

tokens de maneira eficiente, auditável e programaticamente controlada. Para instanciar um novo ativo, o contrato exige apenas quatro parâmetros fundamentais, sendo eles, o nome do token, o seu símbolo identificador, o número de casas decimais e a oferta inicial de unidades a serem emitidas.

Cada nova criação é registrada de forma permanente na blockchain por meio do evento `emit TokenCreated(string name, address tokenAddress, string symbol, uint8 decimals);`, o que possibilita rastrear a origem e as características de todos os tokens gerados. Esse registro também facilita a integração com ferramentas externas, como exploradores de blocos e sistemas de monitoramento, que podem acompanhar em tempo real o histórico de emissão. Além disso, a estrutura de armazenamento implementada no contrato mantém a relação direta entre o nome de cada token e o endereço do contrato correspondente, utilizando o mapeamento `mapping (string => address) public tokens;`. Esse mecanismo permite consultas rápidas e seguras, garantindo que os tokens criados possam ser referenciados e utilizados por outros módulos do ecossistema de forma padronizada.

Complementarmente, o contrato `BaseToken` atua como a implementação genérica do padrão ERC20 sobre o qual os tokens emitidos pela `TokenFactory` são baseados. Esse contrato define o conjunto de funções essenciais que regem o comportamento de um ativo fungível na blockchain, incluindo as operações de transferência de fundos (`transfer`), delegação de permissões a terceiros (`approve`), movimentação de fundos mediante autorização (`transferFrom`), bem como as funções de criação e destruição de tokens (`mint` e `burn`). Essa estrutura assegura que todos os tokens gerados mantenham compatibilidade com o ecossistema EVM, podendo ser integrados a carteiras digitais, protocolos de finanças descentralizadas e outros contratos inteligentes de forma interoperável.

A emissão de novos tokens é rigidamente controlada por meio do modificador `onlyOwner`, cuja implementação garante que apenas o proprietário do contrato possa expandir a oferta monetária. Esse controle é fundamental para evitar inflação não autorizada e preservar a previsibilidade econômica do sistema. De forma semelhante, a função `burn` possibilita a destruição de unidades em circulação, promovendo uma gestão monetária mais equilibrada e adaptável às necessidades do sistema simulado. Por fim, o contrato emite eventos como `Transfer` e `Approval`, que registram de forma imutável todas as transações e autorizações executadas, assegurando total transparência, auditabilidade e rastreabilidade das operações dentro da rede blockchain.

5.2. Frontend sem usabilidade

Na construção da versão do sistema projetada para apresentar menor usabilidade, a ferramenta `V0` foi utilizada para gerar rapidamente uma estrutura de interface sem otimizações intencionais de UX, garantindo que o resultado fosse funcional, mas neutro em termos de boas práticas. Por não aplicar heurísticas de usabilidade refinadas, o protótipo inicial produzido pelo `V0` naturalmente incorpora elementos que elevam a complexidade cognitiva da interação, como baixa visibilidade do estado do sistema, *affordance* menos evidente e menor orientação aos objetivos do usuário, fatores reconhecidos por reduzir a eficiência e a previsibilidade em sistemas interativos [Nielsen 1994].

Nessa versão projetada para apresentar menor usabilidade, a interação com a

aplicação ocorre exclusivamente por meio da extensão *MetaMask*, sem abstrações adicionais de interação ou camadas de apoio ao usuário. Essa escolha se alinha diretamente ao objetivo metodológico de construir uma interface que, embora funcional, incorpore características que ampliam a carga cognitiva e reduzem a fluidez da interação. Ao exigir que o participante comprehenda o processo completo de conexão, assinatura e gerenciamento de transações na própria carteira, a plataforma expõe de forma explícita detalhes técnicos que comprometem os princípios de redução da complexidade, visibilidade adequada do estado do sistema e prevenção de erros [Nielsen 1994].

A ausência de feedbacks imediatos após ações críticas, como o envio de transações, compromete o princípio da visibilidade, uma vez que o sistema deixa de fornecer sinais claros sobre o andamento das operações. Diante dessa falta de retorno, o usuário é obrigado a recorrer à própria inferência para deduzir o estado atual da interface ou consultar manualmente a carteira conectada, o que fragmenta a experiência e dificulta o onboarding na plataforma. Esse processo não apenas aumenta a incerteza durante etapas sensíveis, mas também eleva a carga cognitiva associada à execução das tarefas, especialmente entre usuários com menor familiaridade com aplicações descentralizadas. Da mesma forma, o uso de mensagens de erro com terminologia técnica, típica de interações diretas com carteiras e contratos inteligentes, reduz a legibilidade e a intuitividade da interface, enfraquecendo as heurísticas de consistência, correspondência com o mundo real e ajuda ao diagnóstico e recuperação de erros.

A aplicação foi construída sobre o framework *Next.js*, uma tecnologia amplamente utilizada no ecossistema *JavaScript* que permite o desenvolvimento de aplicações web. A comunicação com a *blockchain* foi implementada por meio das bibliotecas *Viem* e *Wagmi*, ferramentas essenciais para o desenvolvimento de aplicações descentralizadas que interagem com contratos inteligentes. O *Viem* é uma biblioteca moderna e fortemente tipada que fornece abstrações para chamadas de leitura e escrita em contratos, estimativas de gás e manipulação de dados em diferentes redes compatíveis com a *Ethereum Virtual Machine*. Já o *Wagmi*, construído sobre o *Viem*, oferece uma camada de abstração mais orientada a *React*, facilitando o gerenciamento de estados relacionados à conexão da carteira, leitura de dados e execução de transações.

A estrutura do *frontend* foi organizada em seis páginas principais, cada uma correspondendo a uma funcionalidade central do sistema. A primeira delas é a página de login, concebida de forma simples, contendo apenas o botão de conexão com a *MetaMask* ou outras carteiras compatíveis disponíveis no mercado. Esse fluxo inicial tem como objetivo estabelecer a autenticação por meio da carteira digital do usuário, etapa necessária para habilitar a interação com o contrato inteligente.

As Figura 7 e 8 ilustram esse fluxo de conexão, apresentando a interface minimalista que orienta o usuário no processo de vinculação da carteira ao sistema.

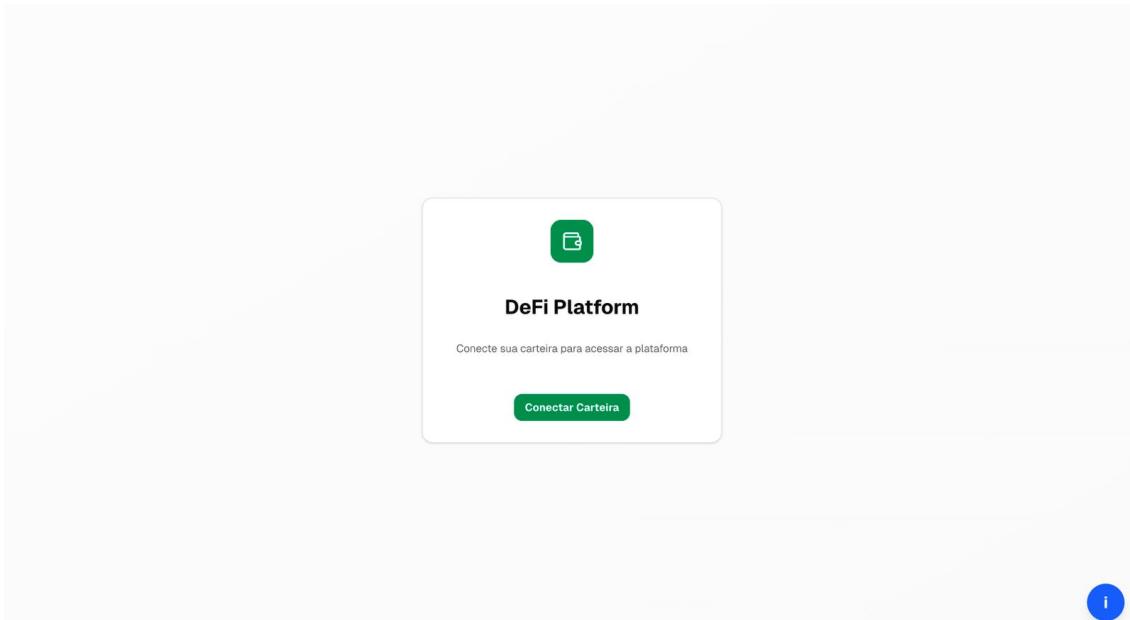


Figura 7. Página de entrada na plataforma, responsável pela conexão da carteira do usuário.

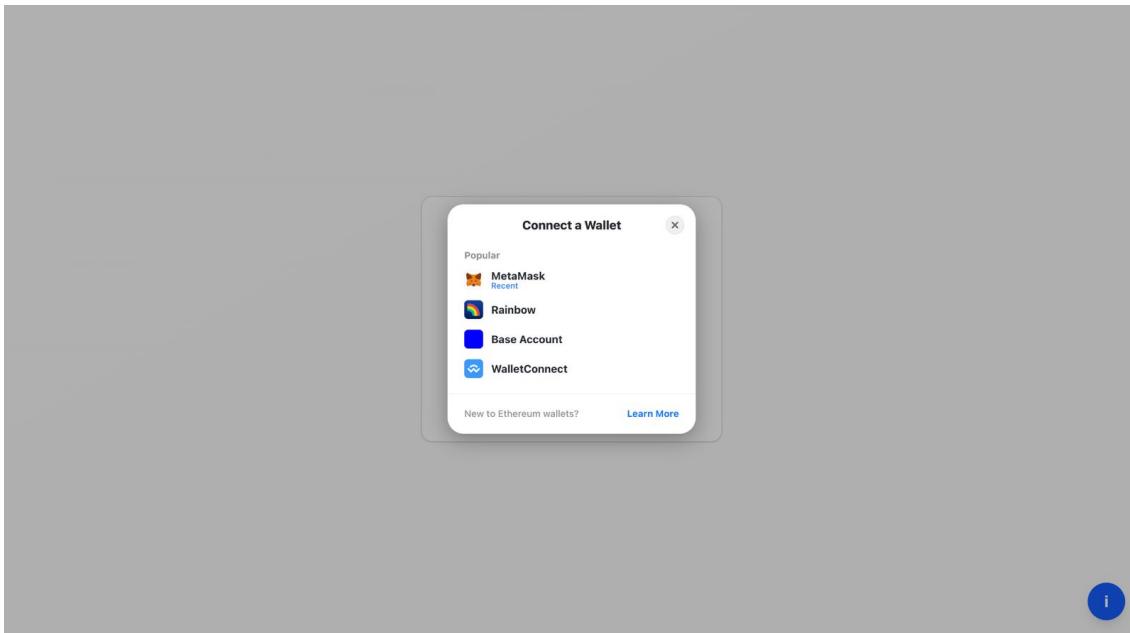


Figura 8. Página de conexão da carteira, responsável pela escolha do provedor de carteira a ser utilizado.

A primeira página após a entrada da plataforma é *Dashboard*, ilustrada na Figura 9, que atua como o ponto de entrada da aplicação. Nela, o usuário pode visualizar as informações consolidadas da carteira conectada, incluindo o saldo de cada token depositado no contrato Vault. Assim que entra na plataforma o usuário é capaz de solicitar tokens de teste, conforme ilustra a Figura 10

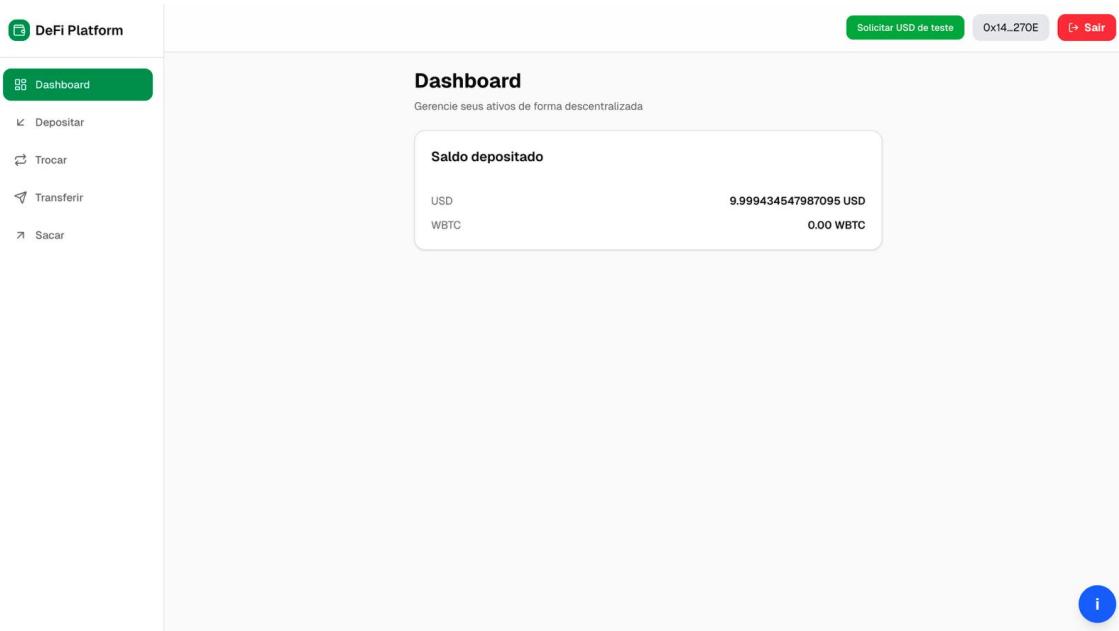


Figura 9. Página *Dashboard*, responsável por exibir o saldo atual da carteira conectada e os valores depositados no contrato vault.

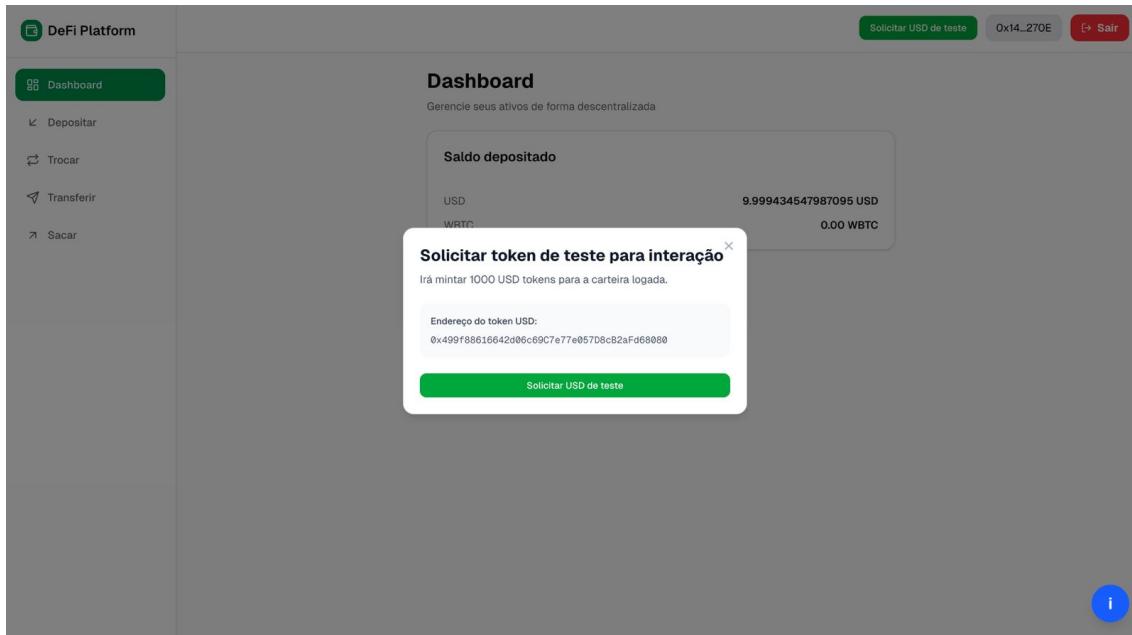


Figura 10. Página *Dashboard* com solicitação de tokens de teste aberta.

A página *Deposit*, exibida na Figura 11, é responsável por permitir que o usuário envie tokens de sua carteira pessoal para o contrato. Essa funcionalidade é essencial para o funcionamento das demais operações, uma vez que somente os tokens previamente depositados podem ser utilizados nas trocas, transferências internas e retiradas.

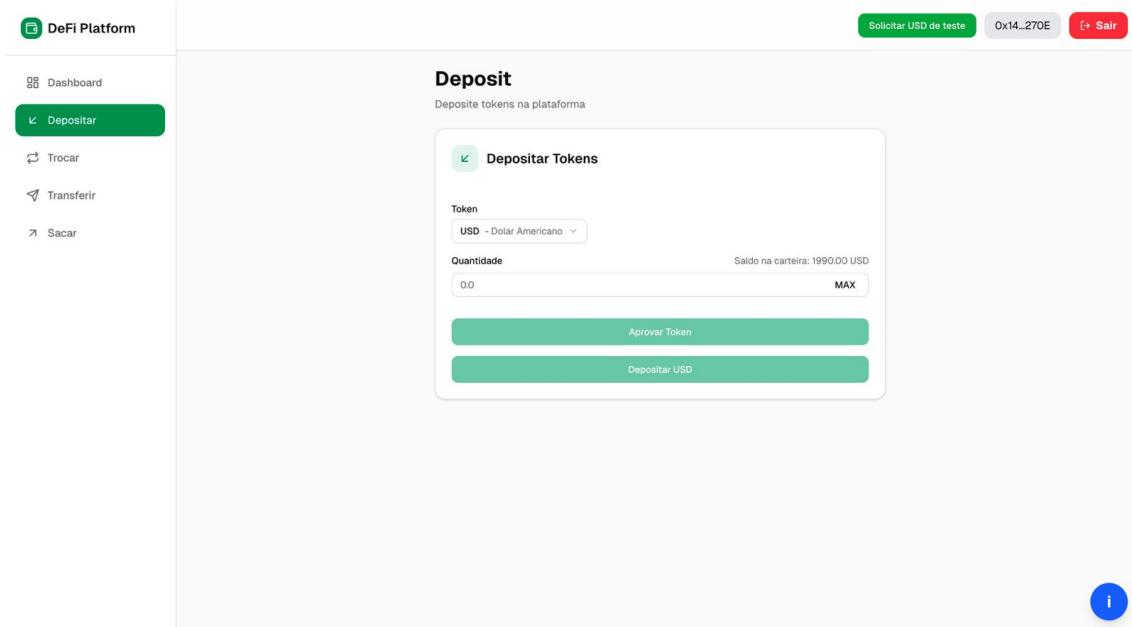


Figura 11. Página *Deposit*, utilizada para enviar tokens ao contrato *Vault* e atualizar os saldos internos do usuário.

Além disso, ao realizar a ação de envio de uma transação, o sistema aciona automaticamente o *handler* da carteira conectada, no caso, a *MetaMask*. Essa etapa é intermediada pela própria carteira, que exibe uma janela modal contendo os detalhes da operação que o usuário deve revisar e autorizar. A Figura 12 apresenta um exemplo dessa interface, ilustrando a tela exibida durante o processo de depósito. Essa mesma janela é aberta para todas as transações solicitadas, uma vez que cada operação requer confirmação explícita na carteira para garantir segurança e validade criptográfica.

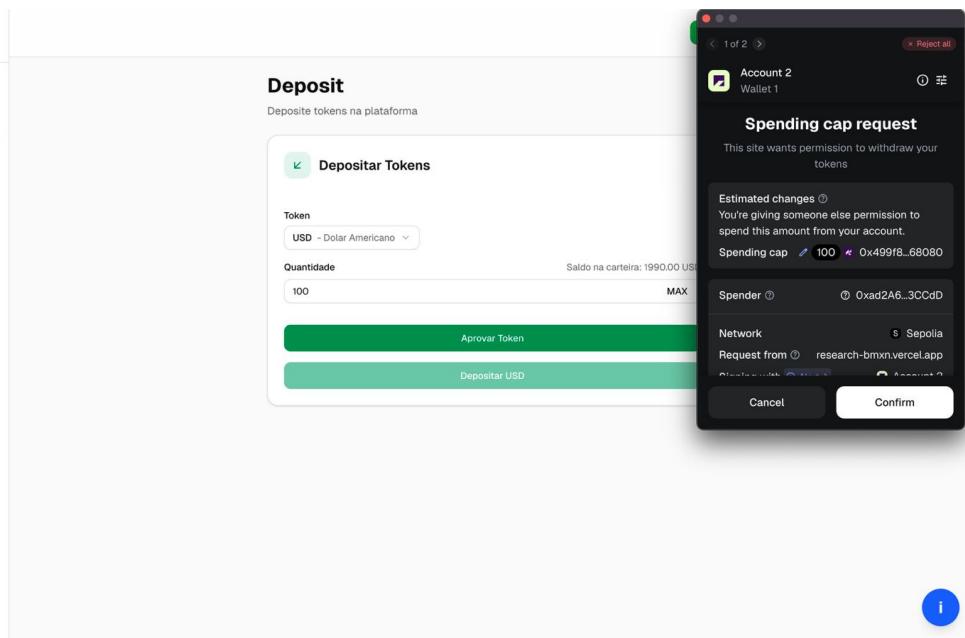


Figura 12. Handler *Metamask*, para confirmação da transação.

A página seguinte, denominada *Troca*, representada na Figura 13, tem como finalidade possibilitar a conversão entre diferentes tokens compatíveis com o sistema. Essa conversão é realizada com base nas taxas de câmbio fornecidas pelos oráculos da *Chainlink*, que garantem a integridade e atualidade das cotações utilizadas. Essa página reflete diretamente a função `swap()` implementada no contrato inteligente, reproduzindo na interface os cálculos e ajustes necessários para a troca entre ativos com diferentes casas decimais.

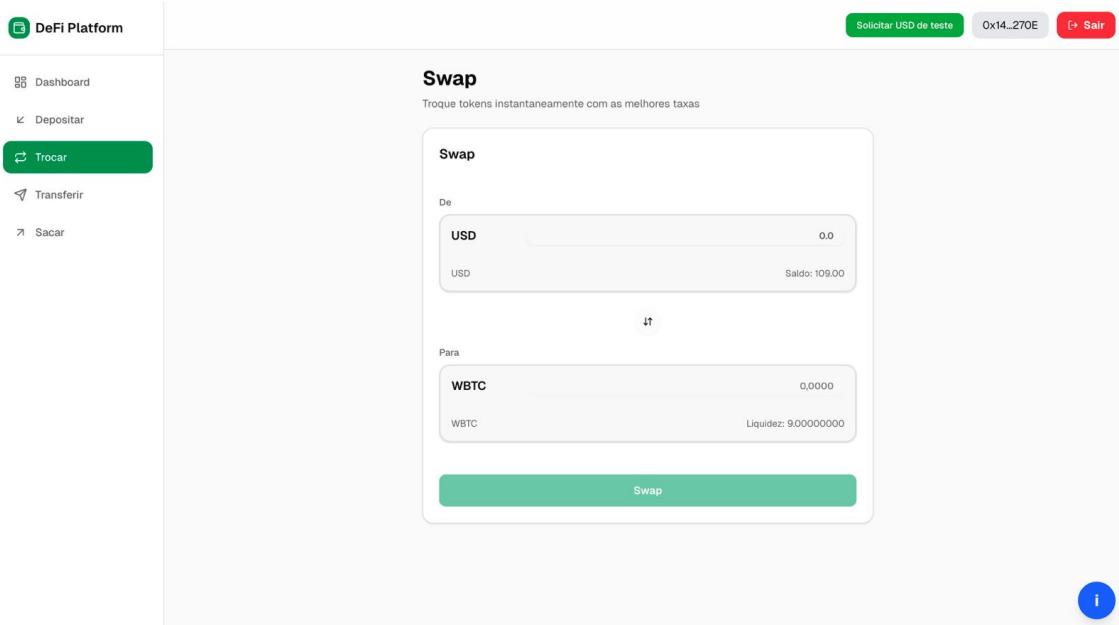


Figura 13. Página *Swap*, que permite a conversão entre tokens utilizando as cotações obtidas via oráculos da *Chainlink*.

A página *Transferir*, apresentada na Figura 14, tem como principal objetivo possibilitar a movimentação de tokens entre diferentes usuários dentro do próprio contrato, sem a necessidade de gerar uma nova transação na blockchain.

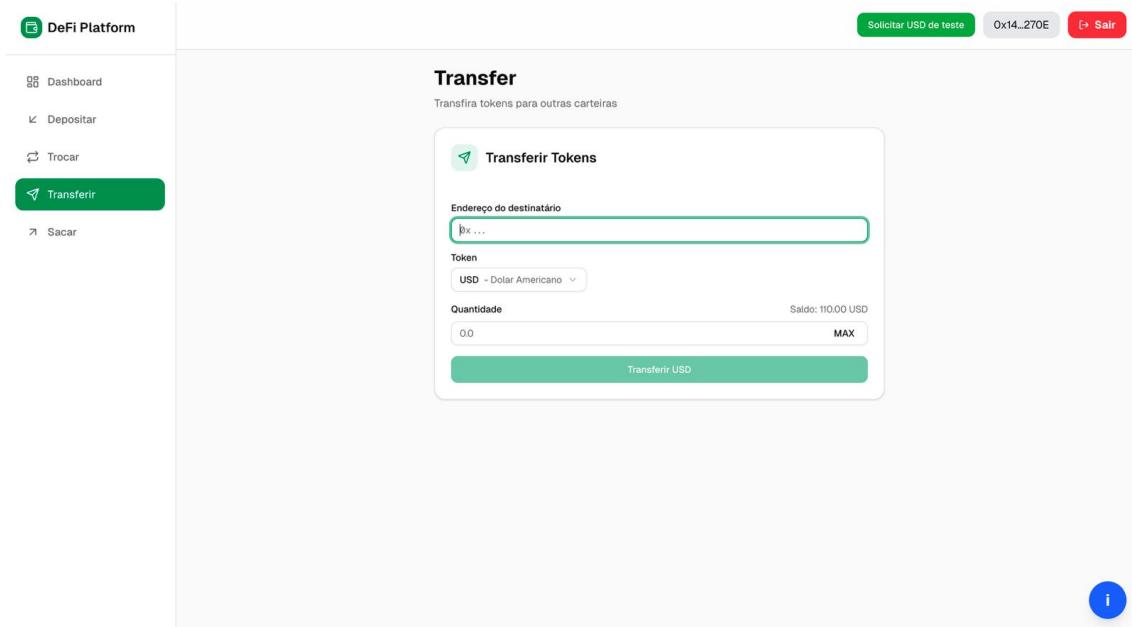


Figura 14. Página *Transfer*, que permite a movimentação interna de tokens entre diferentes usuários dentro do contrato.

Por fim, a página *Sacar*, ilustrada na Figura 15, completa o ciclo de operações disponíveis ao usuário. Nela, é possível efetuar o saque de tokens armazenados no contrato, transferindo-os novamente para a carteira conectada. Assim como nas demais operações, a confirmação é realizada por meio da *MetaMask*, que solicita a assinatura do usuário antes da execução da transação.

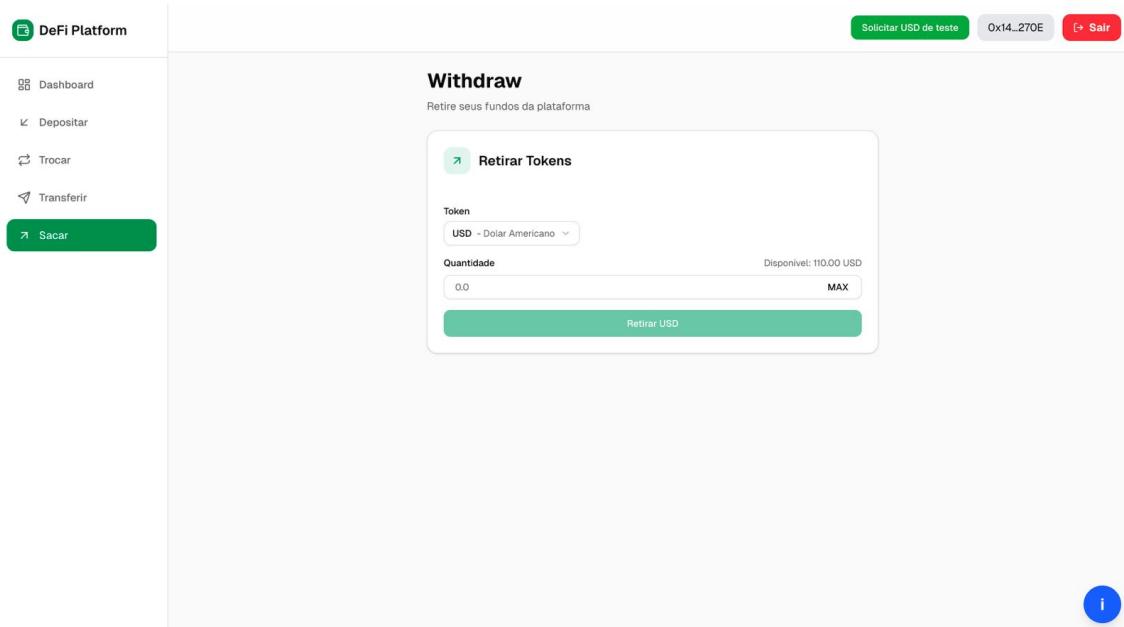


Figura 15. Página *Withdraw*, responsável pelo saque de tokens armazenados no contrato para a carteira conectada.

5.3. Frontend com usabilidade

Na construção da versão do sistema projetada para apresentar maior usabilidade, a interface foi desenvolvida manualmente, seguindo rigorosamente os princípios de UX discutidos ao longo deste trabalho. Em contraste com a estrutura neutra gerada pelo *V0* na versão de menor usabilidade, esta versão foi elaborada com foco explícito na redução da carga cognitiva, no aumento da visibilidade do estado do sistema e na oferta de affordances claras que orientam o usuário durante todo o fluxo de interação. As decisões de design adotadas foram fundamentadas nas heurísticas de Nielsen [Nielsen 1994], priorizando simplicidade, consistência, prevenção de erros e comunicação transparente das ações executadas pelo sistema.

Um dos principais aprimoramentos introduzidos nesta versão é a adoção de *Account Abstraction (AA)* e de um *paymaster* responsável por pagar as taxas de gás das transações, permitindo um fluxo de interação baseado em *smart accounts*. Essa abordagem, viabilizada pela integração com a plataforma *Privy*, oferece ao usuário funcionalidades como *login* social, autenticação por e-mail com código temporário e suporte a *passkeys*, reduzindo drasticamente barreiras de entrada e eliminando a necessidade de interagir diretamente com carteiras como a *MetaMask*. Consequentemente, o usuário não é exposto a janelas externas de assinatura, detalhes de gás ou terminologia técnica própria da blockchain, fatores reconhecidos na literatura como contribuintes para o aumento da carga cognitiva e da taxa de abandono. A remoção da carteira convencional também permitiu a implementação de um *handler* próprio, mais direto e centrado na tarefa, que comunica ao usuário exatamente o que está acontecendo sem expor jargões técnicos, valores de gás ou estruturas internas da *EVM*.

Outro aspecto central desta versão foi a introdução de *feedbacks* imediatos e contextualizados, garantindo que o usuário receba confirmações visuais e textuais a cada ação crítica. Essa escolha fortalece o princípio da visibilidade, reduz incertezas sobre o estado da aplicação e melhora a fluidez da interação, diminuindo a necessidade de inferência ou checagens externas. O uso de mensagens claras, alinhadas ao modelo mental do usuário, também reforça a heurística de correspondência com o mundo real e facilita a compreensão mesmo entre participantes com pouca experiência prévia em ambientes *DeFi*.

Dessa forma, a versão com usabilidade representa uma abordagem projetada intencionalmente para oferecer uma experiência mais intuitiva, consistente e cognitivamente eficiente, alinhada aos princípios teóricos discutidos no estudo. Ela contrasta de maneira controlada com a versão de menor usabilidade, permitindo avaliar de forma precisa o impacto das diferentes abordagens de design na percepção dos participantes e no desempenho durante as tarefas propostas.

A Figura 16 apresenta a página inicial da plataforma com usabilidade aprimorada. Diferentemente da versão sem usabilidade, onde o usuário depende exclusivamente da *MetaMask*, aqui o sistema oferece múltiplas formas simplificadas de autenticação: login por e-mail, autenticação via *Google* e login com *Passkey*.

Esse design reduz o atrito inicial, melhora a acessibilidade e elimina o primeiro grande obstáculo relatado em estudos sobre onboarding em aplicações Web3.

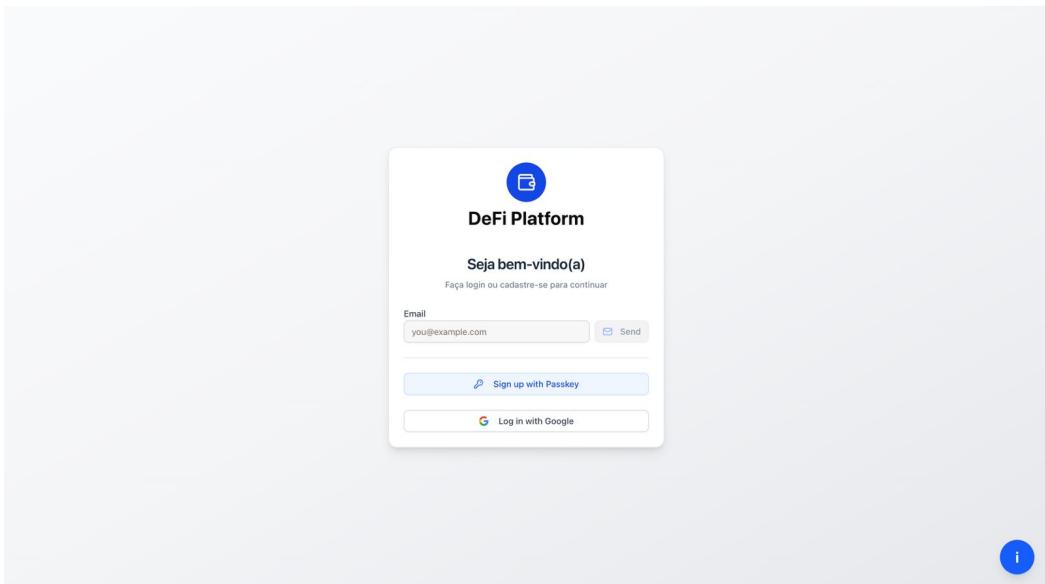


Figura 16. Página de login da plataforma com usabilidade aprimorada, oferecendo autenticação por e-mail, Google e Passkey.

Após a autenticação, o usuário é direcionado ao *Dashboard*, apresentado na Figura 17. Essa página fornece uma visão clara e organizada dos saldos disponíveis na carteira criada via Account Abstraction.

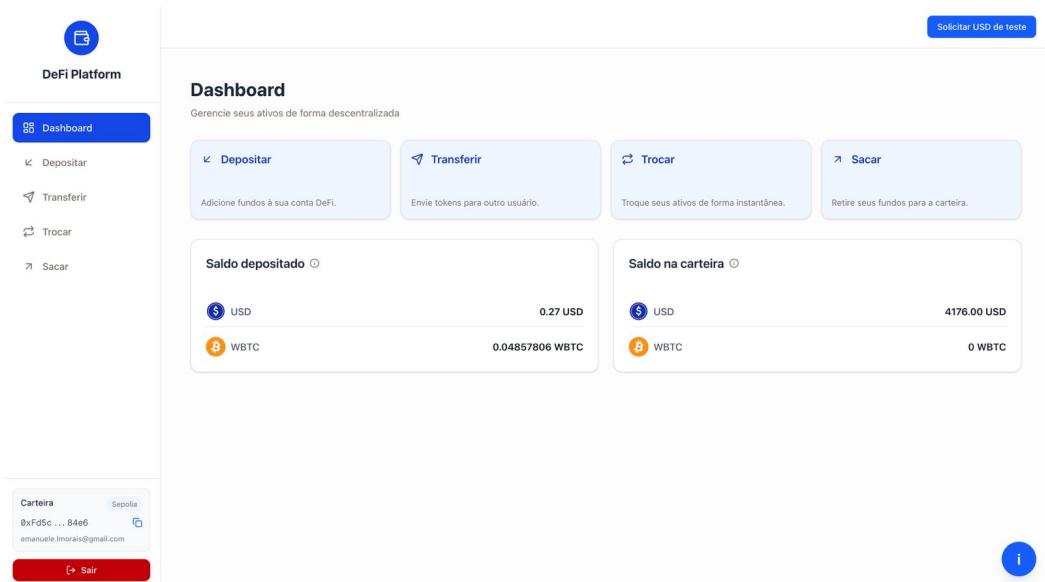


Figura 17. Dashboard da versão com usabilidade, apresentando saldos e navegação simplificada.

A Figura 18 mostra o modal de solicitação de tokens de teste, apresentado de forma clara, com instruções diretas e botão único que evita ambiguidades. Esse é um exemplo de affordance explícita, que orienta o usuário durante as tarefas iniciais.

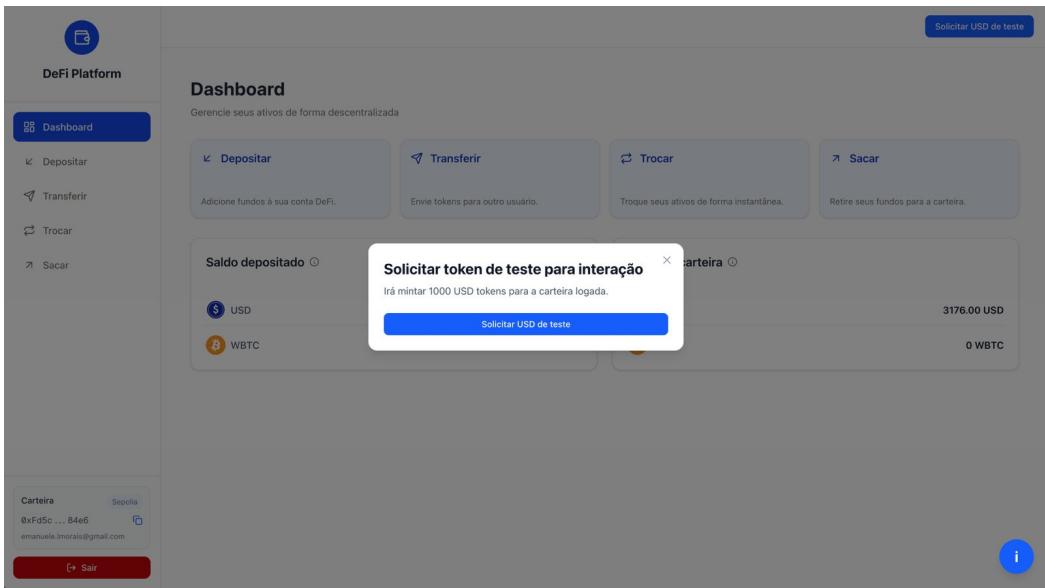


Figura 18. Modal de solicitação de tokens de teste, exibido imediatamente após o acesso inicial.

Na Figura 26, observa-se a página *Depositar*. Aqui, o usuário realiza depósitos diretamente através da smart account, sem necessidade de abrir a *MetaMask*. Todo o processo ocorre na própria interface, com feedback imediato e mensagens contextuais.

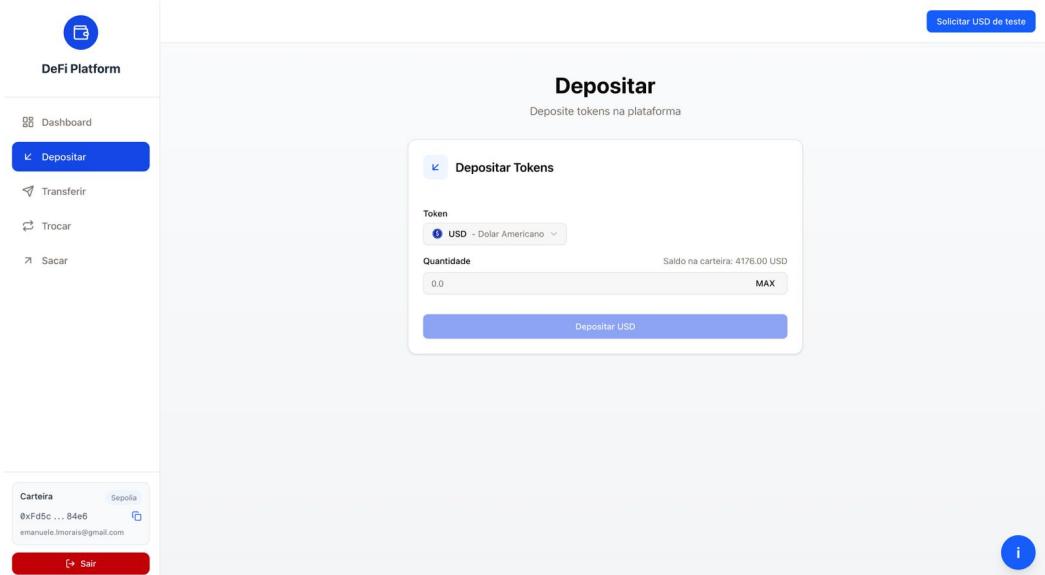


Figura 19. Página de depósito, com validações automáticas e execução interna via smart account.

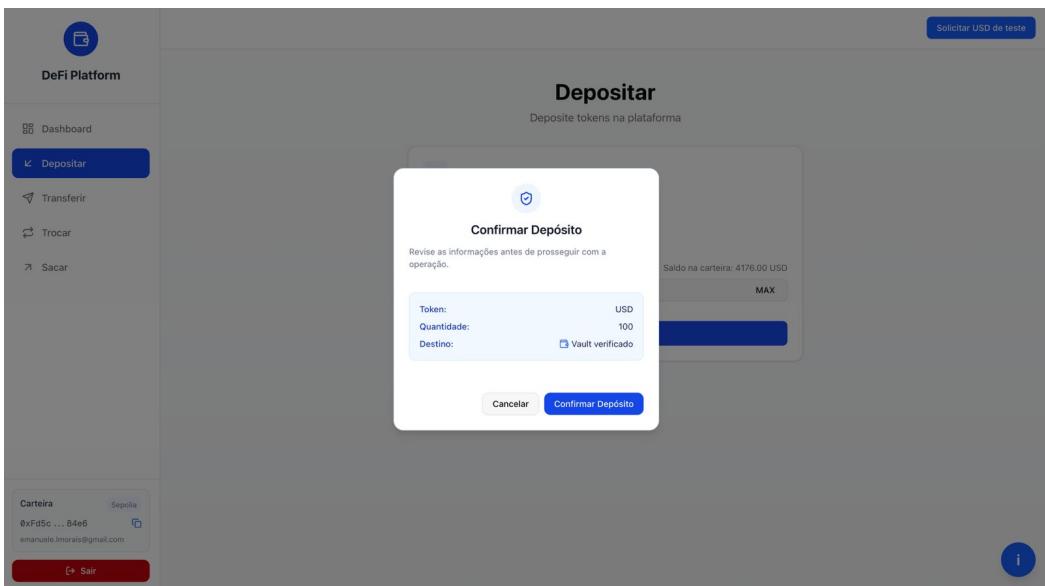


Figura 20. Handler de depósito personalizado

A Figura 21 apresenta a página de transferência de tokens. A principal diferença em relação à versão sem usabilidade é a possibilidade de realizar a transferência por e-mail, o que reduz etapas, evita a necessidade de copiar endereços extensos e se aproxima de modelos de interação já familiares aos usuários de plataformas Web2. Além disso, assim como nas demais páginas, o *handler* de confirmação de transação foi personalizado para apresentar informações de forma clara e contextualizada, diminuindo incertezas durante o processo e reforçando a previsibilidade do fluxo.

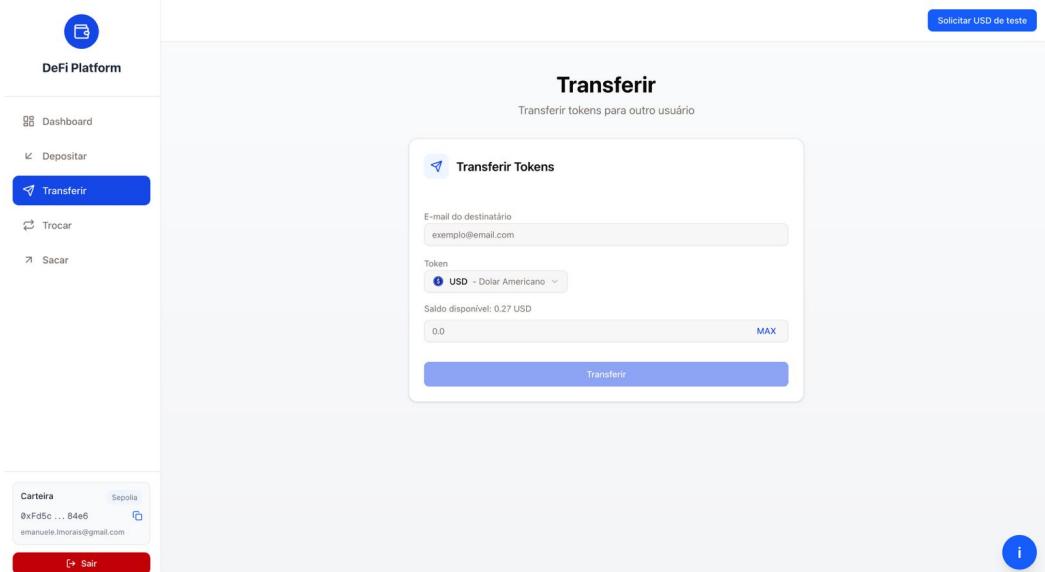


Figura 21. Página de transferência

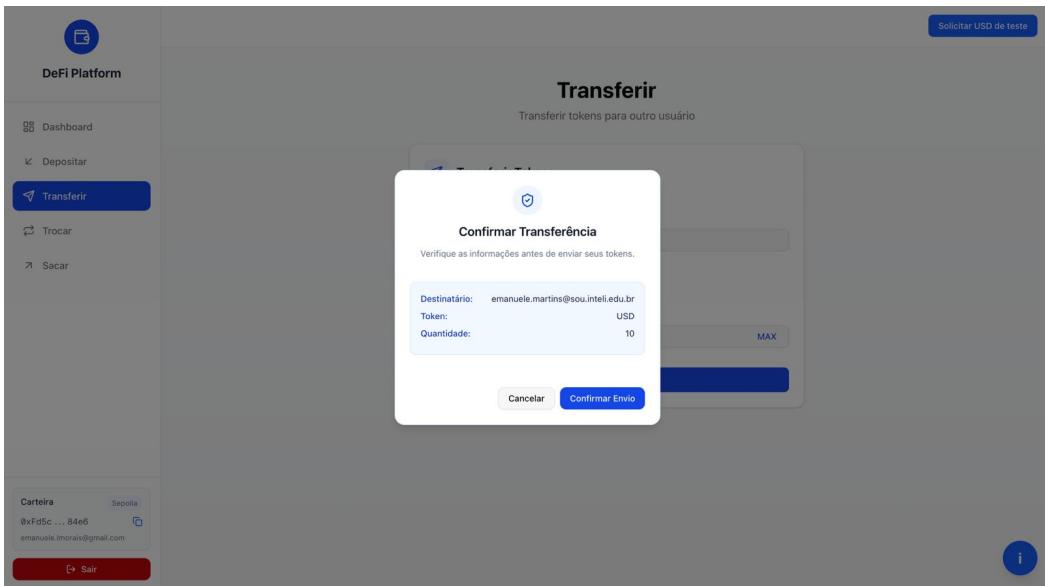


Figura 22. Handler de transferência personalizado

A página de troca, apresentada na Figura 23, incorpora melhorias importantes em relação à versão sem usabilidade. Entre elas, destaca-se o bloqueio automático de transações cujo valor ultrapasse o saldo disponível, tanto em termos de liquidez quanto de fundos do próprio usuário, prevenindo erros comuns e reduzindo tentativas frustradas. Além disso, foram incluídos ícones para facilitar a identificação visual dos tokens e tornar o fluxo mais intuitivo. Assim como nas demais páginas, essa interface também conta com um textithandler de confirmação personalizado, que apresenta informações de confirmação de maneira simples, contribuindo para um processo de tomada de decisão transparente.

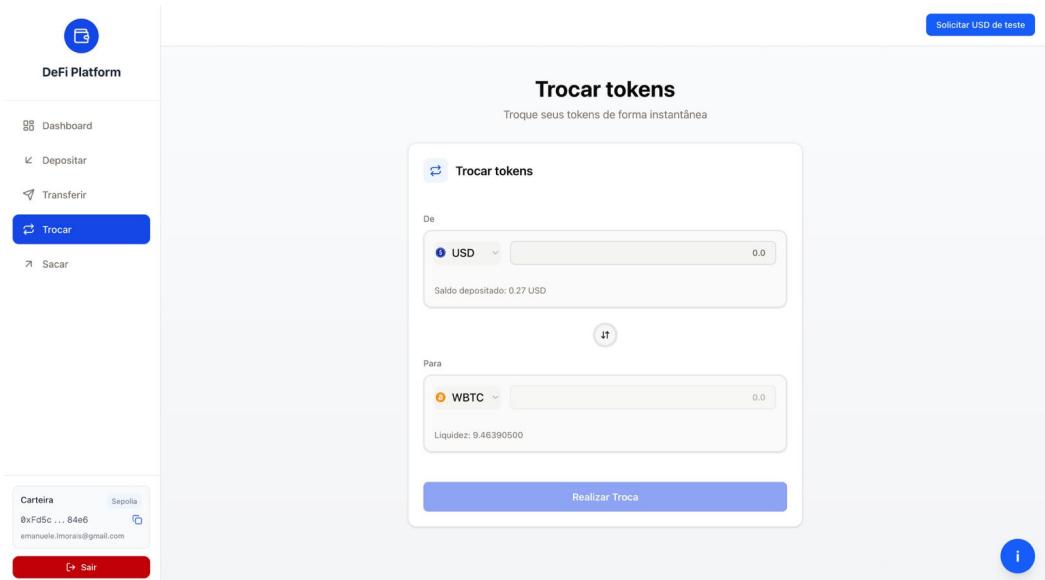


Figura 23. Página de troca de tokens

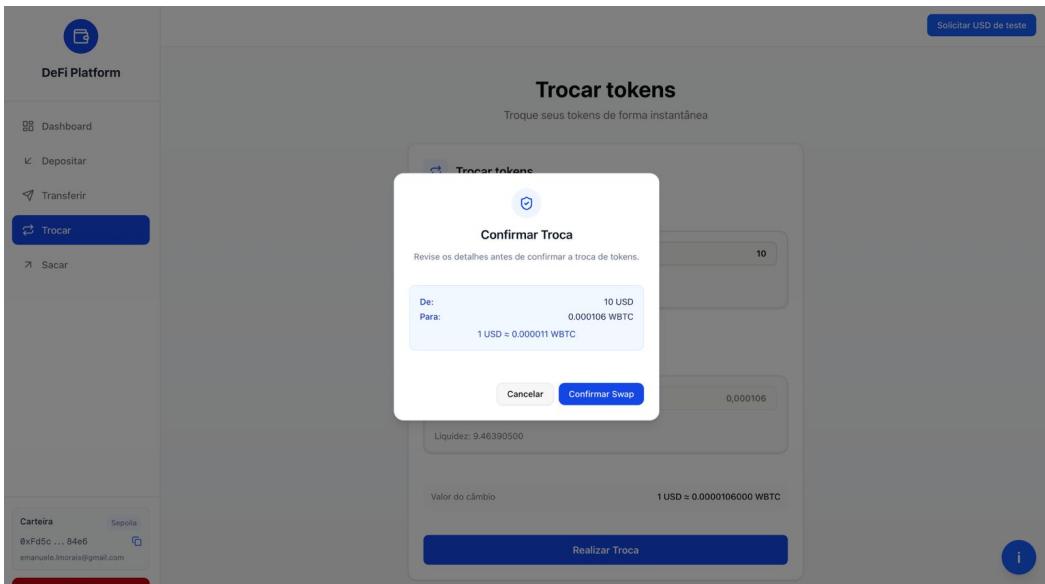


Figura 24. Handler de Troca personalizado

A Figura 25 apresenta a página dedicada ao saque de tokens da plataforma para a *smart account*. Assim como nas demais interfaces, essa página mantém a consistência visual por meio do uso de ícones padronizados para facilitar a identificação dos ativos envolvidos. Além disso, conta também com um *handler* de confirmação personalizado e simplificado.

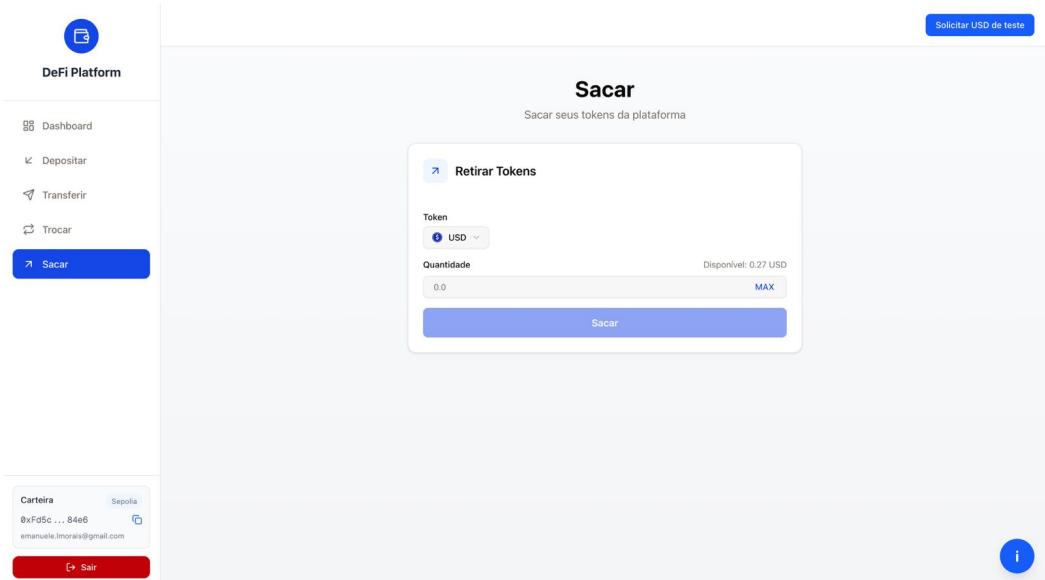


Figura 25. Página de saque

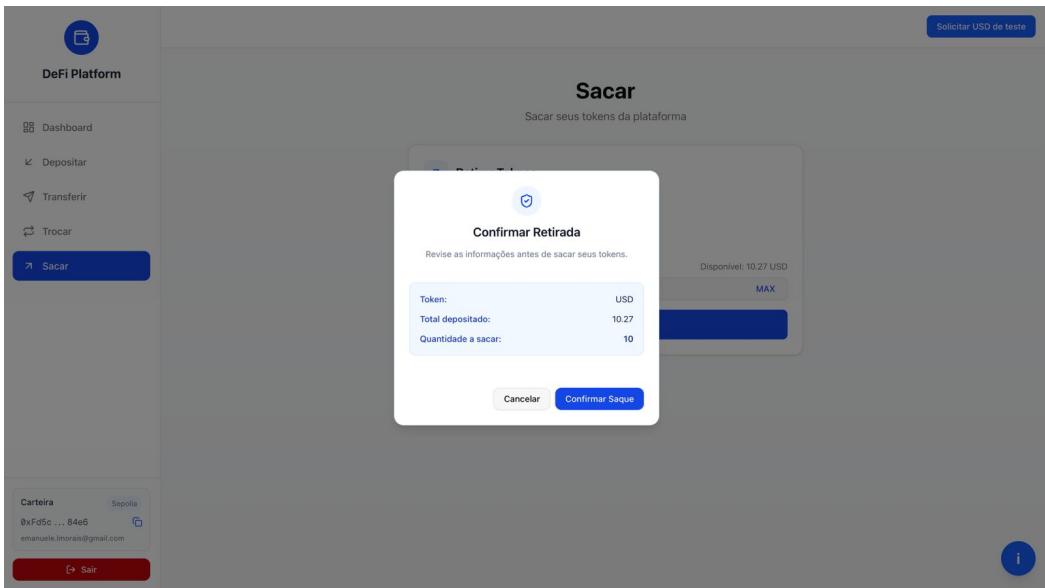


Figura 26. Handler de Saque personalizado

6. Análise dos Resultados

Durante o experimento, os participantes interagiram com ambas as versões do sistema, com e sem usabilidade, conforme o desenho experimental previamente estabelecido. Ao todo, 47 usuários concluíram integralmente o fluxo de interação proposto e respondendo aos instrumentos de avaliação subjetiva ao final da sessão. A média de idade dos participantes foi de 24,6 anos.

Em relação ao nível de familiaridade com tecnologias blockchain, a distribuição dos participantes foi heterogênea:

- 27 indivíduos (57,4 %) se identificaram como iniciantes, relatando pouco ou nenhum contato prévio com aplicações Web3;
- 13 indivíduos (27,7 %) se classificaram como intermediários, afirmando já ter utilizado alguma ferramenta ou plataforma descentralizada;
- 7 indivíduos (14,9 %) foram categorizados como avançados, com experiência contínua ou atuação profissional envolvendo blockchain.

Essa distribuição permitiu observar diferenças significativas entre os grupos, especialmente no que diz respeito à carga cognitiva percebida e à fluência na execução das tarefas.

Para mitigar o viés de aprendizagem, isto é, a possibilidade de os participantes aprenderem o fluxo na primeira interação e terem melhor desempenho na segunda, o experimento utilizou ordem de exposição alternada entre as versões. Assim, 24 participantes iniciaram o teste pela versão sem usabilidade, enquanto 23 participantes começaram pela versão com usabilidade. Esse balanceamento garantiu maior confiabilidade à comparação entre as duas condições experimentais.

Durante a condução dos testes, observou-se que diversos usuários, especialmente os classificados como iniciantes, demonstraram dúvidas e insegurança ao lidar com o processo de criação e conexão de uma carteira própria utilizando a extensão MetaMask, etapa

obrigatória na versão sem usabilidade. Esse comportamento corrobora os achados da literatura de UX em Web3, que apontam que a necessidade de gerenciar chaves privadas, interpretar mensagens técnicas e compreender fluxos de assinatura constitui uma barreira significativa para novos usuários. Tais dificuldades impactaram diretamente a percepção de esforço cognitivo relatado ao final do experimento.

6.1. Análise dos Resultados do NASA-TLX

Para a avaliação da carga de trabalho percebida, foi utilizado o *NASA Raw TLX*, conforme descrito anteriormente, aplicando o instrumento imediatamente após cada interação realizada pelos participantes nas duas versões da plataforma. Dessa forma, cada usuário forneceu um conjunto de respostas para as seis dimensões avaliadas: demanda mental, demanda física, demanda temporal, desempenho, esforço e frustração, permitindo a comparação direta entre as condições com e sem usabilidade.

No questionário original, a métrica de *performance* segue a escala do NASA-TLX tradicional, em que **0 representa “fracasso” e 100 representa “sucesso total”**. Entretanto, para fins de visualização e comparação com as demais dimensões, todas orientadas para que valores mais altos indiquem maior carga de trabalho, foi necessária a **inversão da escala de performance**.

A fórmula utilizada para essa conversão foi:

$$\text{Performance invertida} = 100 - \text{Performance original}$$

Essa transformação tornou os valores de performance coerentes com a direção interpretativa das demais dimensões do TLX, permitindo análises comparativas mais intuitivas nos gráficos apresentados.

6.1.1. Médias gerais

Ao analisar as médias gerais do NASA-TLX considerando todos os participantes, independentemente do nível de conhecimento em blockchain, observa-se um padrão consistente, a versão com usabilidade apresentou valores substancialmente menores em todas as dimensões avaliadas. Esse resultado indica uma redução significativa da carga de trabalho percebida na interação, confirmando que as melhorias de design implementadas impactaram positivamente a experiência de uso de forma ampla e transversal a diferentes perfis de usuários.

A demanda mental apresentou a maior diferença absoluta entre as plataformas, refletindo diretamente o efeito das abstrações introduzidas, como o uso de contas inteligentes, feedbacks imediatos e a eliminação da necessidade de manipulação direta da carteira MetaMask. Dimensões como demanda física e temporal também apresentaram reduções expressivas, sugerindo menor esforço na compreensão das etapas e menor tempo necessário para completar as tarefas.

As dimensões mais associadas ao aspecto emocional de esforço e de frustração também revelaram discrepâncias marcantes. Usuários da versão sem usabilidade relataram níveis elevados de tensão, incerteza e necessidade de repetição de comandos,

enquanto a versão com usabilidade reduziu drasticamente esses efeitos, tornando a experiência mais fluida e previsível.

Por fim, a métrica de *performance invertida*, alinhada ao sentido interpretativo das demais dimensões, reforça esse padrão, quanto melhor a usabilidade percebida, menor a carga associada ao sucesso na realização das tarefas. Esses resultados globais consolidam a evidência de que o conjunto de heurísticas de design aplicadas não apenas beneficiou grupos específicos, mas aumentou de forma geral a qualidade da interação, reduzindo barreiras cognitivas e emocionais enfrentadas durante o uso da plataforma. A figura 27 reflete os resultados obtidos.

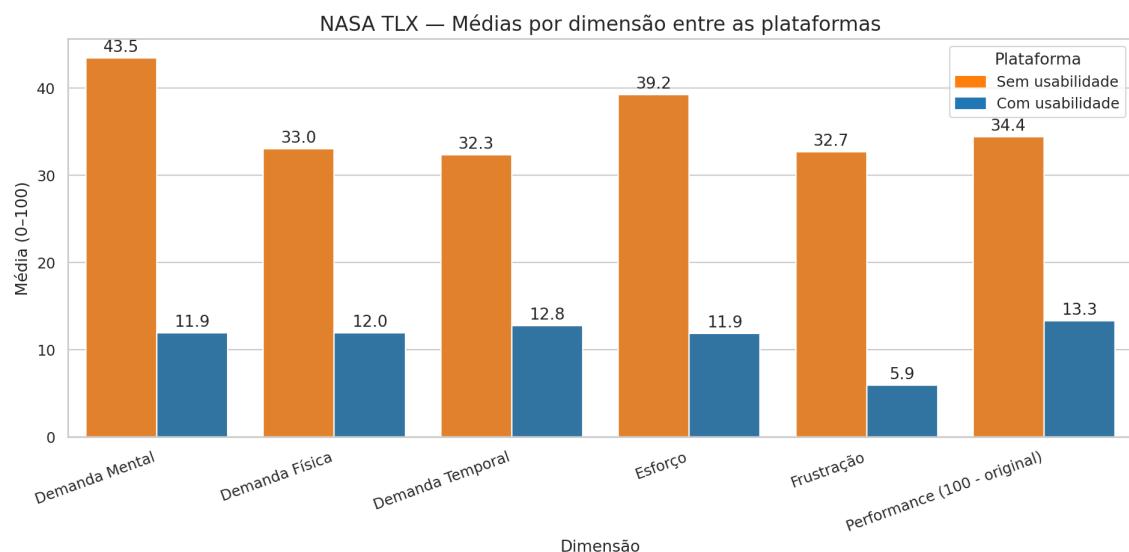


Figura 27. Média geral do NASA-TLX entre as plataformas

6.1.2. Por Nível de Conhecimento

A avaliação realizada por meio do questionário NASA-TLX permitiu identificar diferenças expressivas na carga cognitiva percebida pelos participantes ao interagir com as duas versões da plataforma, considerando separadamente cada nível de familiaridade com tecnologias blockchain. Os resultados demonstram que, independentemente do grau de experiência, a versão com usabilidade promoveu reduções substanciais nas demandas mentais, temporais e emocionais associadas à execução das tarefas propostas. No entanto, a magnitude dessa redução variou significativamente entre os grupos, revelando padrões importantes relativos à influência do conhecimento prévio sobre a experiência de uso.

Desempenho dos participantes iniciantes. Usuários com pouca ou nenhuma familiaridade com aplicações descentralizadas foram os que apresentaram os maiores índices de esforço, frustração e carga mental ao utilizar a versão sem usabilidade. Esse grupo demonstrou forte sensibilidade à ausência de feedbacks claros, terminologias técnicas e necessidade de operar diretamente com a MetaMask. Já na versão com usabilidade, observou-se uma redução drástica em todas as dimensões avaliadas, evidenciando que esse público depende fortemente de mecanismos de suporte, orientação contextual e abstrações de complexidade para realizar tarefas com segurança.

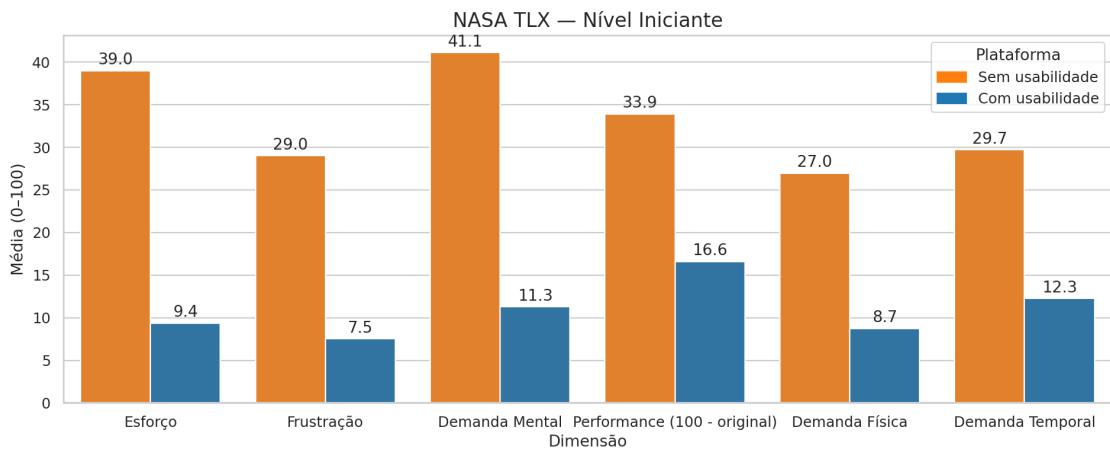


Figura 28. Média do NASA-TLX entre participantes iniciantes

Desempenho dos participantes intermediários. Usuários com algum conhecimento prévio demonstraram menor dificuldade que os iniciantes na versão sem usabilidade, mas ainda assim sofreram impacto significativo das limitações de design. As dimensões de demanda mental e esforço permaneceram elevadas, indicando que a simples familiaridade com o domínio não reduz plenamente a carga cognitiva de fluxos Web3 tradicionais. A versão com usabilidade, por outro lado, resultou em quedas acentuadas nessas cargas, demonstrando que esse grupo se beneficia de forma clara de interfaces mais orientadas à ação do usuário e menos dependentes de configurações técnicas.

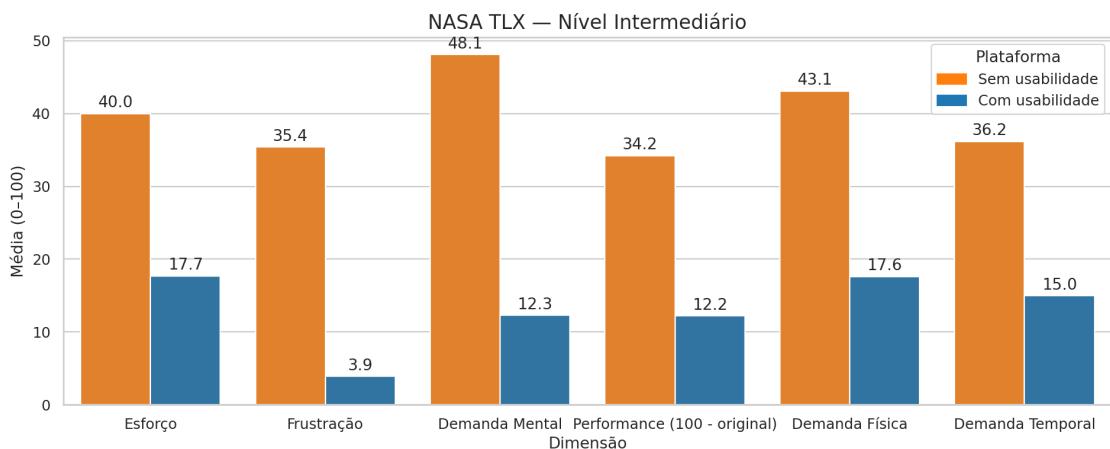


Figura 29. Média do NASA-TLX entre participantes intermediários

Desempenho dos participantes avançados. Participantes com alta familiaridade, incluindo profissionais que trabalham diariamente com blockchain, presentaram desempenho relativamente melhor na versão sem usabilidade quando comparados aos demais grupos. Ainda assim, as avaliações mostraram que mesmo usuários experientes enfrentam esforço adicional decorrente da ausência de abstrações, do feedback limitado e da necessidade de interpretar mensagens técnicas. Na versão com usabilidade, esses usuários relataram reduções substanciais na frustração, na carga mental e na pressão temporal, in-

dicando que melhorias de UX beneficiam até mesmo especialistas, aumentando a fluidez e previsibilidade da interação.

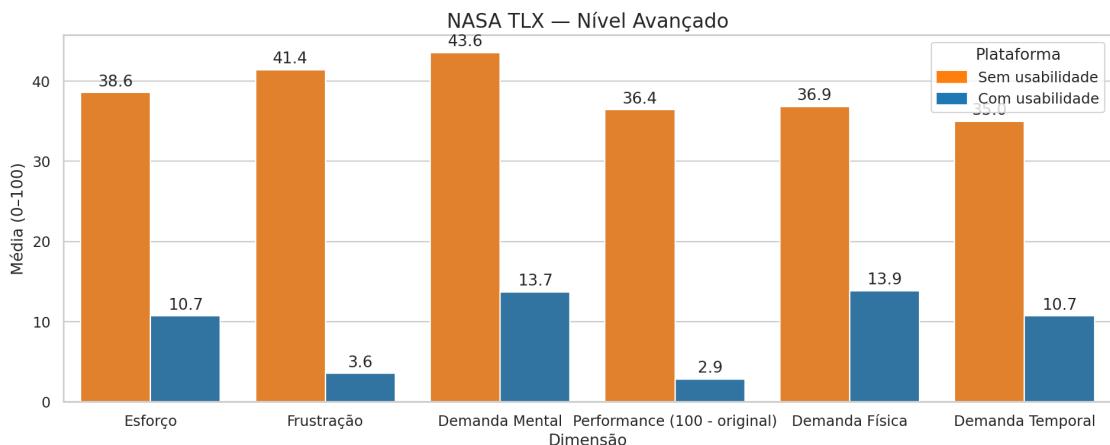


Figura 30. Média do NASA-TLX entre participantes avançados

6.2. Análise dos Resultados de Desempenho e Interação nas Tarefas

Além da avaliação subjetiva da carga de trabalho percebida por meio do NASA-TLX, realizou-se uma análise objetiva do comportamento dos participantes ao utilizar as duas versões da plataforma. Foram examinados o número total de tarefas concluídas, a distribuição dessas tarefas por tipo e a quantidade de cliques necessários para completar as operações. Esses indicadores permitem compreender, sob uma perspectiva quantitativa, como a usabilidade influencia diretamente a eficiência, fluidez e esforço operacional durante a interação.

6.2.1. Tarefas Completas por Plataforma

Foram analisados dados de tarefas completadas pelos 47 participantes que interagiram com ambas as versões da plataforma. Considerando que cada usuário deveria realizar quatro tarefas (Depósito, Saque, Transferência e Troca), o total máximo possível de tarefas a serem concluídas é de **188**. A Figura 31 apresenta o total efetivamente concluído em cada protótipo. Observa-se que a versão com usabilidade resultou em **168 tarefas concluídas**, enquanto a versão sem usabilidade atingiu **143 tarefas**. Assim, a plataforma com usabilidade alcançou **89 % do total possível**, ao passo que a versão sem usabilidade atingiu **76 %**.

Essa diferença substancial indica que a interface aprimorada permitiu que os participantes completassem um número significativamente maior de operações, refletindo maior eficiência tanto na interpretação das instruções quanto na execução das etapas. A proximidade entre o desempenho obtido e o limite teórico máximo sugere que a versão com usabilidade reduziu de forma consistente pontos de atrito, incertezas e erros durante a navegação. Em contraste, a versão sem usabilidade comprometeu a conclusão de parte relevante das atividades, evidenciando o impacto direto de problemas de design na capacidade dos usuários de finalizar tarefas essenciais.

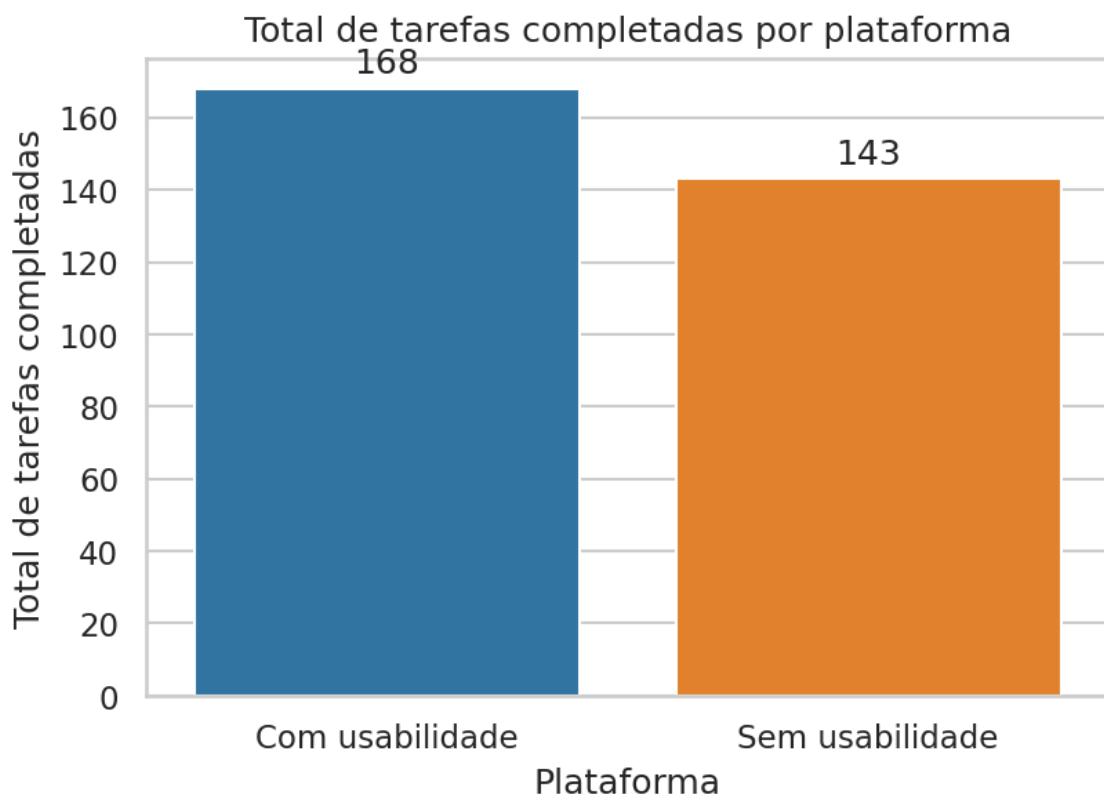


Figura 31. Total de tarefas completadas por plataforma

Já a Figura 32 detalha a quantidade de tarefas realizadas em cada categoria. Como havia 47 participantes no estudo, o total esperado por categoria seria de **47 tarefas**. Valores inferiores indicam usuários que não conseguiram concluir determinada ação. Os resultados demonstram um padrão consistente entre todas as categorias, a plataforma com usabilidade apresentou desempenho superior.

Essas tendências reforçam a hipótese de que a usabilidade impacta não apenas o total geral de tarefas concluídas, mas também a qualidade e fluidez da execução em cada tipo específico de operação. A maior consistência observada na plataforma com usabilidade sugere que as melhorias visuais, estruturais e de orientação reduziram a necessidade de tentativas adicionais, minimizaram ambiguidades e tornaram os fluxos mais previsíveis para participantes de todos os níveis de experiência.

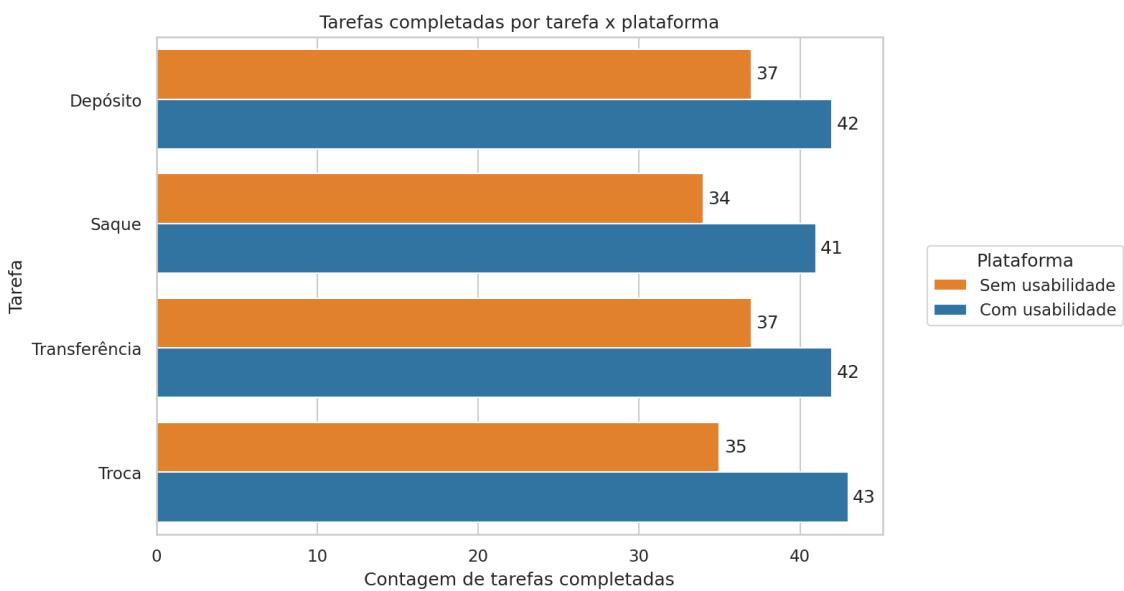


Figura 32. Tarefas completadas por tipo de tarefa em cada plataforma

6.2.2. Cliques Realizados por Plataforma

Para além das tarefas concluídas, a análise dos cliques realizados pelos participantes permite investigar o nível de eficiência operacional e o grau de atrito percebido durante a interação com cada versão da plataforma. Considerando que havia 47 usuários e que cada um deveria realizar quatro tarefas, o número **ideal de cliques por plataforma** seria **188**, assumindo um fluxo linear, sem erros, retrocessos ou repetições. A Figura 33 apresenta o total de cliques registrados em cada protótipo.

Observa-se que a versão **sem usabilidade** acumulou **221 cliques**, ultrapassando de forma significativa o limite esperado. Em contraste, a plataforma **com usabilidade** registrou **186 cliques**, valor praticamente igual ao ideal teórico, sugerindo que a interface minimizou ruídos no processo e reduziu a ocorrência de ações redundantes ou equivocadas. Essa diferença entre as plataformas reflete de forma clara o impacto das melhorias de design, quanto mais intuitiva a interface, menor a necessidade de navegação exploratória e menor o número de cliques excedentes gerados ao longo das tarefas.

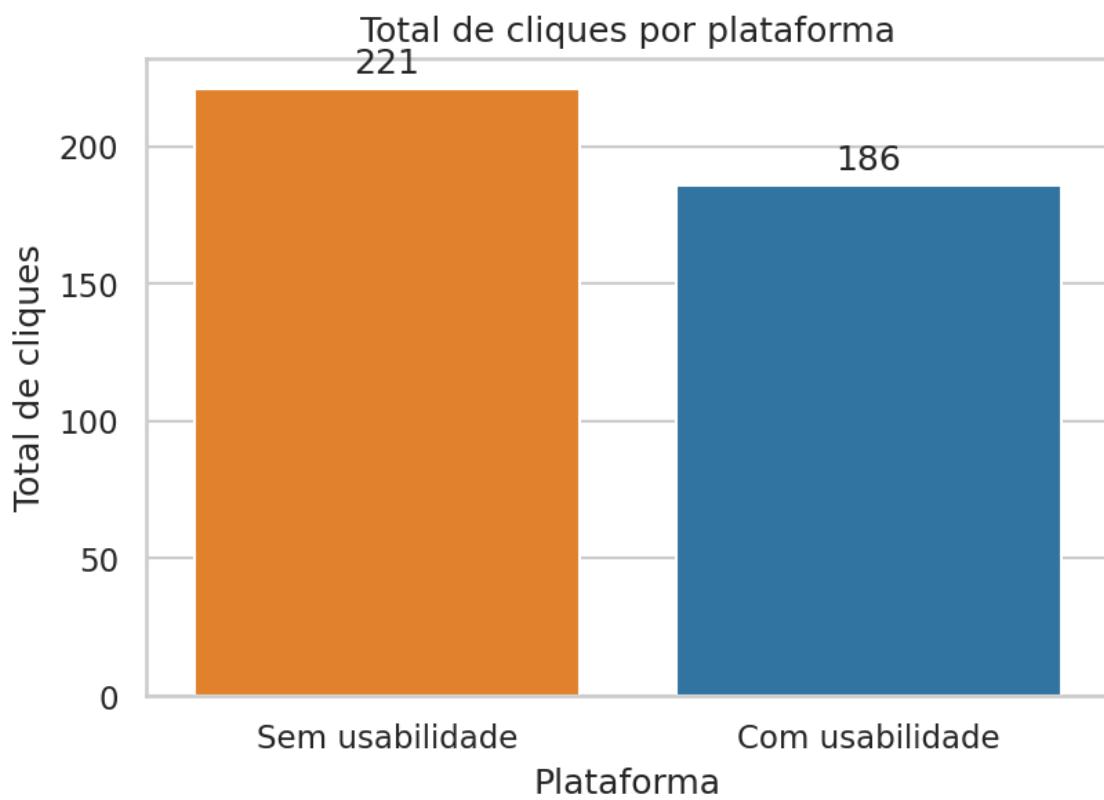


Figura 33. Total de cliques por plataforma

A Figura 34 detalha a distribuição dos cliques por categoria de ação. Assim como no caso das tarefas concluídas, espera-se que cada participante execute cada ação uma única vez, portanto, o valor ideal para cada categoria seria de **47 cliques**. Valores superiores indicam que usuários precisaram repetir etapas, buscar informações adicionais ou corrigir erros durante o fluxo.

Os resultados revelam novamente que todas as categorias, a plataforma **sem usabilidade** apresentou um número muito maior de cliques repetidos. A plataforma **com usabilidade**, por outro lado, manteve valores bem mais próximos do limite superior de 47 cliques esperados. Embora ainda existam repetições, estas ocorreram em magnitude consideravelmente menor, o que reforça a hipótese de que a interface aprimorada reduziu ambiguidades, evitou confusões no fluxo e diminuiu a necessidade de cliques exploratórios.

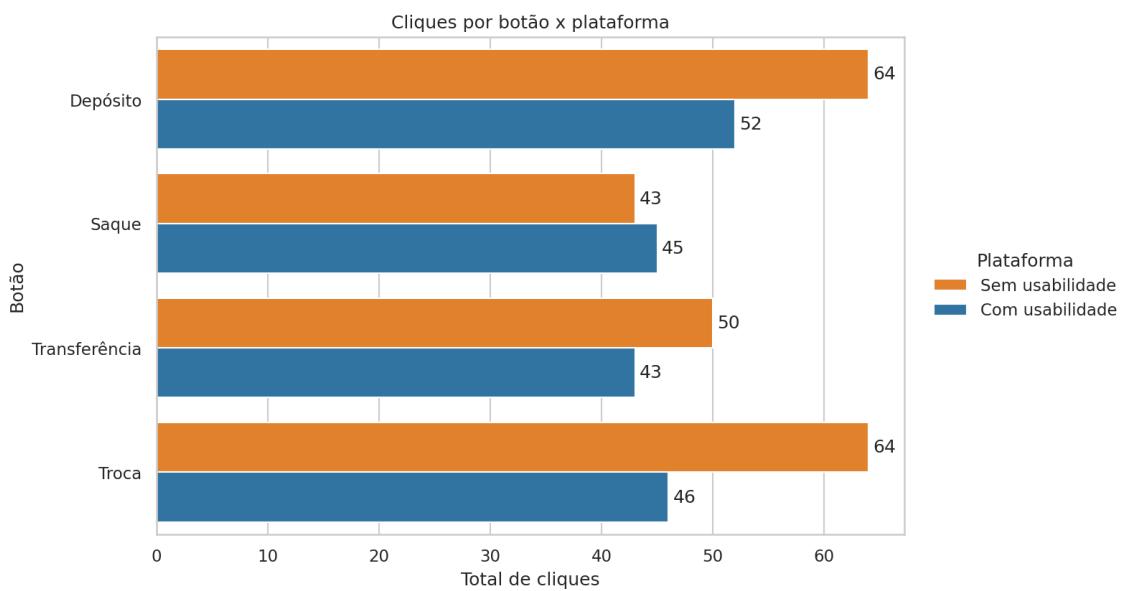


Figura 34. Cliques por botão nas duas plataformas

De modo geral, a análise dos cliques complementa e reforça os achados observados nas tarefas concluídas, a versão com usabilidade não apenas permitiu maior completnude das operações, mas também reduziu substancialmente o esforço interacional medido por cliques excedentes. Essa convergência entre indicadores objetivos (cliques) e comportamentais (tarefas concluídas) fortalece a conclusão de que as heurísticas de design implementadas diminuíram o atrito cognitivo e tornaram a experiência dos usuários mais fluida, eficiente e previsível.

7. Conclusão

Os resultados do estudo mostram que a aplicação de princípios heurísticos de usabilidade em interfaces *DeFi* reduz de maneira consistente o atrito cognitivo e melhora a experiência do usuário. Em conjunto, as evidências empíricas indicam que a segurança técnica, embora essencial, não é suficiente para promover adoção em larga escala. É necessário que o design de interação esteja alinhado às expectativas e limitações dos usuários.

Como implicações práticas, recomenda-se a redução de confirmações e etapas redundantes, a oferta de feedbacks imediatos e comprehensíveis, a normalização de unidades e a manutenção de consistência entre termos e fluxos. Sugere-se também a adoção de mecanismos de abstração de carteiras, permitindo um processo de autenticação mais próximo do que já é amplamente compreendido em ambientes *Web2*, como *login* por *e-mail* ou *passkey*. A inclusão de *handlers* personalizados, capazes de apresentar informações de maneira simplificada e contextualizada, pode ainda reduzir dúvidas recorrentes e melhorar a fluidez da interação.

Pesquisas futuras podem ampliar o escopo para diferentes perfis de experiência e para cenários com maior complexidade operacional, como operações com múltiplas etapas, autorizações e empréstimos. Também é possível investigar métricas adicionais de desempenho e de confiança percebida ao longo do tempo. Esses caminhos podem acelerar

a maturidade das interfaces *DeFi* e torná-las mais acessíveis, eficientes e confiáveis para um público cada vez mais diversificado.

Referências

- Albayati, H., Kim, S. K., and Rho, J. J. (2021). A study on the use of cryptocurrency wallets from a user experience perspective. *Human Behavior and Emerging Technologies*, 3(5):720–738.
- Antonopoulos, A. M. (2014). *Mastering Bitcoin: Unlocking Digital Cryptocurrencies*. O'Reilly Media.
- Antonopoulos, A. M. and Wood, G. (2018). *Mastering Ethereum: Building Smart Contracts and DApps*. O'Reilly Media.
- ASSILA, A., Oliveira, K., and Ezzedine, H. (2016). Integration of subjective and objective usability evaluation based on iso/iec 15939: A case study for traffic supervision systems. *International Journal of Human-Computer Interaction*, 32.
- Back, A. (2002). Hashcash - a denial of service counter-measure. In *Technical Report*. Acesso em: 9 out. 2025.
- Bashir, I. (2018). *Mastering Blockchain: Distributed ledger technology, decentralization, and smart contracts explained*. Packt Publishing, 2nd edition.
- Bastien, J. M. C. (2010). Usability testing: a review of some methodological and technical aspects of the method. *International Journal of Medical Informatics*, 79(4):e18–e23.
- Buterin, V. (2014). Ethereum: A next-generation smart contract and decentralized application platform. <https://ethereum.org/en/whitepaper/>.
- Byers, J. C., Bittner, A. C., and Hill, S. G. (1989). Traditional and raw task load index (tlx) correlations: Are paired comparisons necessary? In *Advances in Industrial Ergonomics and Safety*, volume 1, pages 481–485. Taylor & Francis. Accessed: 18 Oct 2025.
- Dai, W. (1998). b-money. <http://www.weidai.com/bmoney.txt>. Acesso em: 9 out. 2025.
- Daniels, N. (2025). What is blockchain technology and how does it work? *VPNOverview*. Accessed: 18 Oct 2025.
- Fitts, P. M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*.
- for Standardization, I. O. (2018). *Ergonomics of human-system interaction – Part 11: Usability: Definitions and concepts*. International Organization for Standardization, Geneva, Switzerland, iso 9241-11:2018(e) edition.
- Foster, M. E., Giuliani, M., and Knoll, A. (2009). Comparing objective and subjective measures of usability in a human-robot dialogue system. In *Proceedings of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 879–887, Suntec, Singapore. Association for Computational Linguistics.

- Froehlich, M., Wagenhaus, M. R., Schmidt, A., and Alt, F. (2021). Don't stop me now! exploring challenges of first-time cryptocurrency users. In *Proceedings of the 2021 ACM Designing Interactive Systems Conference*, page 138–148, New York, NY, USA. Association for Computing Machinery.
- Glomann, L., Schmid, M., and Kitajewa, N. (2019). Improving the blockchain user experience - an approach to address blockchain mass adoption issues from a human-centred perspective. In *International Conference on Applied Human Factors and Ergonomics*.
- Gogel, D., Deshmukh, S., Geest, A., Resas, D., Sillaber, C., and Werbach, K. (2021). Defi beyond the hype: The emerging world of decentralized finance.
- Grier, R. A. (2015). How high is high? a meta-analysis of nasa-tlx global workload scores. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 59, pages 1727–1731. SAGE Publications. Accessed: 18 Oct 2025.
- Group, N. N. (2023). Usability (user) testing 101. Acessado em: 26 set. 2025.
- Hart, S. G. (2006). Nasa-task load index (nasa-tlx); 20 years later. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 50, pages 904–908. SAGE Publications. Accessed: 18 Oct 2025.
- Hart, S. G. and Staveland, L. E. (1988). Development of nasa-tlx (task load index): Results of empirical and theoretical research. In *Advances in Psychology*, volume 52, pages 139–183. Elsevier. Accessed: 18 Oct 2025.
- Hewett, T. T., Baecker, R., Card, S., Carey, T., Gasen, J., Mantei, M., Perlman, G., Strong, G., and Verplank, W. (1992). Acm sigchi curricula for human-computer interaction. Technical report, Association for Computing Machinery.
- Hick, W. E. (1952). On the rate of gain of information. *Quarterly Journal of Experimental Psychology*.
- Jang, H., Han, S.-H., and Kim, J. H. (2020). User perspectives on blockchain technology: User-centered evaluation and design strategies for dapps. *IEEE Access*, 8:226213–226223.
- Mayer, R. C., Davis, J. H., and Schoorman, F. D. (1995). An integrative model of organizational trust. *Academy of Management Review*.
- McGrenere, J. and Ho, W. (2000). Affordances: Clarifying and evolving a concept. In *Proceedings of the Graphics Interface Conference*.
- Moniruzzaman, M., Chowdhury, F., and Ferdous, M. S. (2020). Examining usability issues in blockchain-based cryptocurrency wallets. In Bhuiyan, T., Rahman, M. M., and Ali, M. A., editors, *Cyber Security and Computer Science. ICONCS 2020*, volume 325 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 637–653. Springer, Cham.
- Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. <https://bitcoin.org/bitcoin.pdf>. Acesso em: 9 out. 2025.
- Nielsen, J. (1994). *Usability Engineering*. Morgan Kaufmann, San Francisco, CA.
- Nielsen, J. and Molich, R. (1990). Heuristic evaluation of user interfaces. In *International Conference on Human Factors in Computing Systems*.

- Norman, D. A. (2013). *The Design of Everyday Things*. Basic Books, revised and expanded edition edition.
- Preece, J., Rogers, Y., and Sharp, H. (2002). *Interaction Design: Beyond Human-Computer Interaction*. Wiley, New York, NY, 1st edition.
- Rosenbaum, K. (2019). *Groking Bitcoin*. Manning Publications, Shelter Island, NY.
- Rubin, J. and Chisnell, D. (2008). *Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests*. Wiley Publishing, Inc., Indianapolis, IN, 2 edition.
- Saldivar, J., Martínez-Vicente, E., Rozas, D., Valiente, M.-C., and Hassan, S. (2023). Blockchain (not) for everyone: Design challenges of blockchain-based applications. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*, New York, NY, USA. Association for Computing Machinery.
- Saltzer, J. H. and Schroeder, M. D. (1975). The protection of information in computer systems. *Proceedings of the IEEE*.
- Sauro, J. and Lewis, J. R. (2012). *Quantifying the User Experience: Practical Statistics for User Research*. Morgan Kaufmann.
- Shneiderman, B. and Plaisant, C. (2004). *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Pearson/Addison Wesley, Boston, 4 edition.
- Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., Elmquist, N., and Diakopoulos, N. (2009). *Designing the User Interface: Strategies for Effective Human-Computer Interaction*.
- Si, J. J., Sharma, T., and Wang, K. Y. (2024). Understanding user-perceived security risks and mitigation strategies in the web3 ecosystem. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, New York, NY, USA. Association for Computing Machinery.
- Tullis, T. and Albert, B. (2008). *Measuring the User Experience: Collecting, Analyzing, and Presenting Usability Metrics*. Morgan Kaufmann.
- Voskobojnikov, A., Wiese, O., Koushki, M. M., Roth, V., and Beznosov, K. (2021). The u in crypto stands for usable: An empirical study of user experience with mobile cryptocurrency wallets. *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*.
- Whitten, A. and Tygar, J. D. (1999). Why johnny can't encrypt: A usability evaluation of pgp 5.0. In *Proceedings of the 8th USENIX Security Symposium*.
- Wood, G. (2014). Ethereum: A secure decentralised generalised transaction ledger. Technical report, Ethereum Project Yellow Paper. Acesso em: 9 out. 2025.