UNIVERSIDADE ESTADUAL DA PARAÍBA
CAMPUS VII - PATOS
CENTRO DE CIÊNCIAS EXATAS E SOCIAIS APLICADAS
CURSO DE GRADUAÇÃO EM BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

FRANCILEUDO DA SILVA OLIVEIRA

DESENVOLVIMENTO DE UMA CAMADA DE SEGURANÇA PARA A API *HYPERDRIVE* INTEGRADA AO SISTEMA DARK

PATOS - PB
2024

FRANCILEUDO DA SILVA OLIVEIRA

DESENVOLVIMENTO DE UMA CAMADA DE SEGURANÇA PARA A API *HYPERDRIVE* INTEGRADA AO SISTEMA DARK

Trabalho de Conclusão de Curso apresentado ao curso de Bacharelado em Ciência da Computação do Centro de Ciências Exatas e Sociais Aplicadas da Universidade Estadual da Paraíba, como requisito parcial à obtenção do título de bacharel em Ciência da Computação.

Orientador: Dr. Jucelio Soares dos Santos
Coorientador: Dr. Demetrio Gomes Mestre

PATOS - PB
2024

O48d Oliveira, Francileudo da Silva.

Desenvolvimento de uma camada de segurança para a api *hyperdrive* integrada ao sistema *dark* [manuscrito] / Francileudo da Silva Oliveira. - 2024.

58 f. : il. color.

Digitado.

Trabalho de Conclusão de Curso (Graduação em Ciência da computação) - Universidade Estadual da Paraíba, Centro de Ciências Exatas e Sociais Aplicadas, 2024.

"Orientação : Prof. Dr. Jucelio Soares dos Santos, Coordenação do Curso de Computação - CCEA".

"Coorientação: Prof. Dr. Demetrio Gomes Mestre, None".

1. Identificadores persistentes. 2. Blockchain. 3. Hyperdrive. 4. Serviço WEB. 5. Segurança de métodos. 6. Imultabilidade. I. Título

21. ed. CDD 004.6

Elaborada por Kaliane Eveny Martins de Oliveira - CRB - 15/986 BSC7

FRANCILEUDO DA SILVA OLIVEIRA

DESENVOLVIMENTO DE UMA CAMADA DE SEGURANÇA PARA A API HYPERDRIVE INTEGRADA AO SISTEMA DARK

Trabalho de Conclusão de Curso apresentado à Coordenação do Curso de Ciência da Computação da Universidade Estadual da Paraíba, como requisito parcial à obtenção do título de Bacharel em Ciência da Computação

Aprovada em: 18/11/2024.

Documento assinado eletronicamente por:

- **Jucelio Soares dos Santos** (***.475.114-**), em **26/11/2024 11:49:18** com chave **9d565640ac0511efa2f22618257239a1**.
- **Samuel Alves Medeiros** (***.187.184-**), em **29/11/2024 09:19:27** com chave **2db56562ae4c11ef9bea1a7cc27eb1f9**.
- **Giovanna Trigueiro de Almeida Araújo** (***.352.004-**), em **27/11/2024 10:06:20** com chave **653e4806acc011ef91f606adb0a3afce**.

Documento emitido pelo SUAP. Para comprovar sua autenticidade, faça a leitura do QrCode ao lado ou acesse https://suap.uepb.edu.br/comum/autenticar_documento/ e informe os dados a seguir.
Tipo de Documento: Termo de Aprovação de Projeto Final
Data da Emissão: 29/11/2024
Código de Autenticação: e267e3

Dedico este trabalho a minha família, amigos, professores e todos que me apoiaram nesta jornada. Este trabalho só foi possível graças a vocês.

AGRADECIMENTOS

Primeiramente, agradeço a Deus, que me deu saúde e força para superar as dificuldades ao longo dessa jornada.

Agradeço aos meus pais, pelo amor incondicional, apoio e incentivo em todos os momentos. Vocês são minha inspiração e sem vocês nada disso seria possível.

Agradeço aos meus professores, em especial aos meus orientadores, pela paciência, dedicação e conhecimento compartilhado. Seu apoio foi fundamental para a realização deste trabalho.

Agradeço aos meus colegas de curso, pelo companheirismo, pelas discussões produtivas e pelos momentos de descontração que tornaram essa jornada mais leve.

Por fim, agradeço a todos que, direta ou indiretamente, contribuíram para a minha formação acadêmica e para a realização deste trabalho. Cada um de vocês tem uma parcela neste sucesso e sou grato por isso.

“O importante não é ver o que ninguém viu, mas pensar o que ninguém ainda pensou sobre aquilo que todo mundo vê."

Arthur Schopenhauer

RESUMO

A informação científica, considerada o principal recurso para o avanço da ciência, necessita de processos de representação e organização. O método de organização amplamente utilizado atualmente é o uso de Identificadores Persistentes (PIDs), pois eles podem manter a integridade dos dados não levando em conta sua localidade. Com o uso do sistema Hyperdrive, é possível atribuir, atualizar e recuperar PIDs para a gestão de publicações científicas, melhorando a integridade dos dados e a imutabilidade dos recursos. Dito isso, o objetivo geral deste trabalho é a análise desse sistema, capaz de interagir com o sistema dARK, de maneira segura, prática e eficiente. Além disso, buscou-se o desenvolvimento de uma camada de segurança para os endpoints do Hyperdrive. Esse trabalho tem como metodologia uma revisão bibliográfica sobre os conceitos de PIDs, DOIs, ARK, dARK, blockchain e APIs. Com essa compreensão dos elementos envolvidos no desenvolvimento da API é esperado uma implementação eficaz de ferramentas de autenticação em seus endpoints. Conclui-se que a camada de segurança imposta pela proteção dos endpoints é perceptível que os métodos de segurança apresentados nesse trabalho tornam os endpoints do sistema seguros evitando o acesso mal intencionado de não usuários.

Palavras-chave: Identificadores Persistentes; Blockchain; Hyperdrive; Serviço WEB; Segurança de métodos; Imultabilidade.

ABSTRACT

Scientific information, considered the main resource for the advancement of science, requires representation and organization processes. The method of organization widely used today is the use of Persistent Identifiers (PIDs), as they can maintain the integrity of the data without taking its location into account. Using the Hyperdrive system, it is possible to assign, update and retrieve PIDs for the management of scientific publications, improving data integrity and the immutability of resources. That said, the general aim of this work is to analyze this system, which is capable of interacting with the dARK system in a safe, practical and efficient way. In addition, the aim is to develop a security layer for Hyperdrive endpoints. This work is based on a literature review of the concepts of PIDs, DOIs, ARK, dARK, blockchain and APIs. With this understanding of the elements involved in API development, an effective implementation of authentication tools on its endpoints is expected. It is concluded that the security layer imposed by endpoint protection is noticeable and that the security methods presented in this work make the system's endpoints secure by preventing malicious access by non-users.

Keywords: Persistent Identifiers; Blockchain; Hyperdrive; WEB Service; Method Security; Immutability.

LISTA DE ILUSTRAÇÕES

LISTA DE ABREVIATURAS E SIGLAS

API	*Application Programming Interface*
ARK	*Archival Resource Key*
CSV	*comma separated values*
CURL	*Client for URL*
DOI	*Digital Object Identifier*
HTTP	*Hypertext Transfer Protocol*
IBICT	*Institulo Brasileiro de Informação em Ciência e Tecnologia*
JSON	*JavaScript Object Notation*
JWE	*JSON Web Encryption*
JWS	*JSON Web Signature*
JWT	*Json Web Token*
MAC	*Código de Autenticação de Mensagem*
ORCID	*Open Researcher and Contributor ID*
PID	*Persistent identifier*
P2P	*Peer-to-peer*
REST	*Representational State Transfer*
SGBD	*Sitema Gerenciador de Banco de Dados*
RNP	*Rede Nacional de Ensino e Pesquisa*
URI	*Uniform Resource Identifier*
URL	*Uniform Resource Locator*

SUMÁRIO

1 INTRODUÇÃO

Este capítulo apresenta o contexto, a problemática e a proposta de solução relacionadas ao desenvolvimento da API *Hyperdrive* e sua integração com o sistema dARK.

1.1 Contextualização do Problema

A crescente produção de conhecimento científico e tecnológico impõe desafios significativos em relação à organização, preservação e acessibilidade da informação. A ciência moderna, impulsionada pela colaboração global e pela disseminação rápida de descobertas, exige que os dados e publicações sejam não apenas gerados, mas também gerenciados de forma que garantam sua longevidade e integridade. Nesse contexto, a representação e organização adequadas da informação científica são fundamentais para assegurar que essa informação seja facilmente recuperável e visível para a comunidade acadêmica e para o público em geral (Ramachandran; Bugbee; Murphy, 2021).

Um dos principais instrumentos para alcançar essa organização eficaz é o uso de Identificadores Persistentes (PIDs) (Klein; Balakireva, 2020). Os PIDs são códigos exclusivos atribuídos a objetos digitais, como artigos científicos, *datasets*, e perfis de pesquisadores, que garantem a integridade e a continuidade da identificação desses objetos ao longo do tempo, independentemente de onde estejam armazenados ou de mudanças no ambiente digital (Plomp, 2020). Entre os PIDs mais conhecidos, destacam-se o *Digital Object Identifier* (DOI) (Mondal; Mondal, 2023), o *Open Researcher and Contributor ID* (ORCID) (Silva, 2021) e o *Archival Resource Key* (ARK) (Koster, 2020).

O DOI, por exemplo, é amplamente utilizado para identificar publicações acadêmicas, permitindo que elas sejam facilmente localizadas e citadas na internet, contribuindo para a preservação digital e o acesso contínuo a recursos científicos (Okune; Chan, 2023). Similarmente, o ORCID oferece uma solução para a identificação única de autores e colaboradores, promovendo a correta atribuição de crédito pelas contribuições científicas (Silva, 2021). Já o ARK fornece uma alternativa para a identificação persistente de objetos informativos, sendo especialmente útil em contextos onde a durabilidade e a resiliência da identificação são críticas (Kelly et al., 2021b).

Com o avanço das tecnologias de armazenamento e comunicação de dados, surge a necessidade de integrar esses identificadores persistentes em sistemas mais avançados e seguros. Um exemplo promissor dessa integração é o sistema dARK (Segundo et al., 2023), que utiliza a tecnologia *blockchain* para descentralizar a gestão dos identificadores ARK. A *blockchain*, uma tecnologia de registro distribuído, permite que múltiplas instituições colaborem na manutenção e atualização de identificadores persistentes de maneira segura, graças à sua arquitetura *peer-to-peer* e ao uso de algoritmos de consenso, como o *Proof-of-Work* (Huynh-The et al., 2023).

A descentralização promovida pela *blockchain* oferece várias vantagens, como a resiliência contra falhas de sistema, a transparência nas transações e a imutabilidade dos registros.

No entanto, ela também introduz novos desafios de segurança, especialmente quando se trata de garantir que a comunicação entre diferentes sistemas e nós da rede seja protegida contra acessos não autorizados, interceptações e manipulações maliciosas (Singh; Hosen; Yoon, 2021; Yazdinejad et al., 2020).

1.2 Problema

A produção e disseminação de conhecimento no ambiente acadêmico dependem diretamente da gestão eficaz de PIDs, que asseguram a rastreabilidade e a integridade das publicações e pesquisas. Com o crescimento exponencial do volume de dados gerados por pesquisas e a crescente complexidade das infraestruturas tecnológicas, há uma demanda urgente por soluções que garantam não apenas a eficiência na gestão desses PIDs, mas também a segurança dos dados e das operações realizadas.

Nesse contexto, o *Hyperdrive* (Medeiros, 2024) surge como uma API integrada ao sistema dARK (Segundo et al., 2023), a fim de ampliar as capacidades de gestão e comunicação entre diferentes plataformas acadêmicas. Essa integração assegura que as interações e transações ocorram de forma confiável e segura, sobretudo em um ambiente onde a proteção de dados é essencial. A segurança das informações e das transmissões durante essa interação constitui um dos pilares fundamentais deste projeto, especialmente considerando o cenário acadêmico, onde a integridade dos dados é crítica.

Portanto, a necessidade de uma solução que não apenas gerencie os PIDs de forma eficaz, mas que também garanta um elevado nível de segurança para proteger as operações e os dados envolvidos, é a justificativa central desta pesquisa. O desenvolvimento dessa solução visa contribuir significativamente para o avanço das práticas de gestão de dados no ambiente acadêmico, assegurando a confiabilidade, a integridade e a segurança das informações ao longo do tempo.

1.3 Proposta de Solução

Diante do cenário apresentado, o desenvolvimento de uma API como o *Hyperdrive* (Medeiros, 2024) torna-se indispensável para facilitar a integração e a comunicação com o sistema dARK (Segundo et al., 2023). No entanto, considerando a complexidade e a sensibilidade das operações que serão realizadas por essa API, torna-se imprescindível a implementação de uma camada de segurança robusta. Este projeto é resultado de uma colaboração entre a Universidade Estadual da Paraíba (UEPB) e outras instituições parceiras, conforme estabelecido no termo de outorga (Anexo A).

A proposta de solução deste trabalho é o desenvolvimento de uma camada de segurança que proteja de forma eficaz as operações da API *Hyperdrive*, garantindo a integridade e a confidencialidade dos dados gerenciados. Essa camada de segurança será a base para uma

operação segura e confiável da API, especialmente no contexto de sistemas descentralizados, como o dARK.

A pesquisa segue uma abordagem aplicada, que inicia com uma revisão bibliográfica sobre PIDs, sistemas descentralizados e a tecnologia *blockchain*, progredindo para o desenvolvimento e teste da API e seus sistemas de segurança. Ao final, espera-se que o *Hyperdrive* não apenas facilite a comunicação com o sistema dARK, mas também estabeleça um novo padrão de segurança para APIs em ambientes descentralizados, reforçando a confiança e a eficiência na disseminação e no acesso ao conhecimento científico.

1.4 Objetivos

O objetivo deste trabalho é implementar uma camada de segurança para a API *Hyperdrive*, a fim de garantir proteção eficaz de suas operações e dados. Com essa abordagem, espera-se assegurar que a API *Hyperdrive* opere de maneira segura, preservando a integridade e confidencialidade das informações, além de prevenir acessos não autorizados e ataques maliciosos.

Para se alcançar o objetivo geral deste trabalho, foram necessários atingir os seguintes objetivos específicos:

- Desenvolver uma camada de segurança robusta para a API, assegurando a integridade, a confidencialidade das informações gerenciadas e a proteção das transmissões durante a interação com o sistema dARK, prevenindo acessos não autorizados e ataques maliciosos;
- Utilizar a tecnologia *blockchain* para gerenciar os PIDs, garantindo uma gestão segura e imutável das informações no ambiente acadêmico;
- Garantir que a API *Hyperdrive* opere de maneira eficiente e segura, alinhando-se aos padrões de segurança da informação exigidos no contexto acadêmico.
- Avaliar o comportamento do *middleware* de autenticação do Hyperdrive em diferentes cenários, validando sua capacidade de controlar o acesso a rotas públicas e protegidas, garantindo a segurança do sistema sem comprometer a usabilidade.

1.5 Metodologia

A metodologia científica pode ser classificada segundo diversos critérios, incluindo a natureza da pesquisa, os objetivos, a abordagem e os procedimentos adotados. Considerando o contexto e a proposta do trabalho relacionado ao desenvolvimento da API *Hyperdrive*, a seguir, classificaremos a pesquisa de acordo com esses critérios.

Quanto à natureza, a pesquisa é classificada como **aplicada**, uma vez que visa desenvolver uma solução prática e funcional para um problema específico no contexto acadêmico (Valentino; Juanico, 2020). O objetivo principal é propor uma camada de segurança robusta para a API *Hyperdrive*, garantindo a proteção eficaz dos dados e das operações realizadas no sistema dARK.

A pesquisa aplicada busca gerar conhecimentos com aplicabilidade imediata e impacto direto na prática.

Quanto aos objetivos, a pesquisa possui um caráter **descritivo**, pois envolve a descrição detalhada dos conceitos e práticas relacionadas aos PIDs, sistemas descentralizados e segurança da informação (Siedlecki, 2020). Além disso, tem um caráter **explicativo**, pois busca identificar as causas e fatores que exigem a implementação de uma camada de segurança na API, explicando como essa camada pode prevenir ameaças e proteger os dados.

A abordagem adotada é **qualitativa** (Gaus, 2017), uma vez que a proposta da solução se baseia em uma análise teórica e conceitual dos requisitos de segurança e das tecnologias envolvidas, sem a realização de testes quantitativos ou medições de desempenho. A abordagem qualitativa permite uma compreensão aprofundada dos desafios e das necessidades de segurança da API *Hyperdrive*.

Quanto aos procedimentos, a pesquisa é classificada como **experimental** (Ross; Morrison, 2013), pois envolve o desenvolvimento conceitual e a proposição de uma camada de segurança para a API *Hyperdrive* em um ambiente controlado. Embora não tenham sido realizados testes de segurança práticos, a pesquisa define os parâmetros e diretrizes necessários para a implementação da solução, avaliando sua aplicabilidade teórica.

1.6 Estrutura do Trabalho

Este trabalho foi divido em cinco capítulos, sendo organizado da seguinte maneira: no Capítulo 1, apresentamos uma visão geral deste trabalho em relação à contextualização do problema, proposta de solução, objetivos, metodologia e estrutura do trabalho; no Capítulo 2, apresentamos o referencial teórico desta pesquisa; no Capítulo 3, descrevemos a proposta de solução para a camada de segurança do Hyperdrive; no capítulo 4, analisamos e discutimos os resultados alcançados. Por fim, no Capítulo 5, apresentamos as conclusões finais da pesquisa e sugerimos possíveis direções futuras para a pesquisa nesse campo; e ao final, encontra-se as referências e anexo utilizado do decorrer desta pesquisa.

2 REFERENCIAL TEÓRICO

O seguinte capítulo busca construir um fundamento sólido para a plena compreensão das bases que norteiam o *Hyperdrive*, conceituando os PIDs, apresentando os princípios que norteiam a tecnologia *blockchain*. Ademais, é importante destacar que este capítulo procura apresentar uma base para a compreensão e desafios relacionados a integração das tecnologias citadas.

2.1 Identificadores Persistentes (PIDs) e *Uniform Resource Locators* (URLs)

Na Internet, o *Uniform Resource Identifier* (URI) é utilizado para especificar um identificador único que representa um recurso na Web de dados (Zaidan et al., 2018). A *Uniform Resource Locator* (URL), por sua vez, é um tipo específico de URI que identifica um recurso com base em sua principal forma de acesso, geralmente sua localização na rede, em vez de outros atributos que o recurso possa ter (W3C, 2001).

Apesar da sua importância, a Internet ainda enfrentam desafios, como links quebrados e mudanças frequentes de URLs. A diversidade de sistemas de informação e a multiplicidade de documentos criam um cenário mais complexo, exigindo soluções mais robustas para a identificação e localização eficiente de objetos digitais (Guedes; Shintaku; Brito, 2013).

Para documentos científicos, o uso de PIDs é fundamental, pois garante a continuidade e a integridade do acesso a esses documentos ao longo do tempo. Diferentemente das URLs comuns, que podem quebrar ou se tornar obsoletas devido a mudanças na localização ou propriedade dos documentos, os PIDs asseguram que os links permanecem funcionais independentemente dessas alterações (Hardisty et al., 2021).

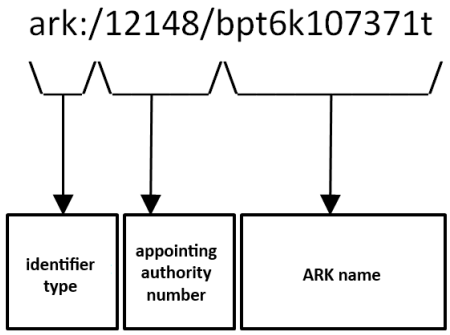
Essa funcionalidade se deve ao fato de que os PIDs não estão ligados a uma localização específica na rede, mas ao próprio objeto digital. Assim, mesmo que o documento seja movido para outra plataforma ou transferido para um novo proprietário, o acesso ao conteúdo continua ininterrupto. Essa robustez é essencial no contexto acadêmico, onde a preservação e a acessibilidade a longo prazo dos recursos são essenciais para a pesquisa e o desenvolvimento científico (Sayão, 2007). Atualmente existe uma variedade de sistemas que utilizam os PIDs, podendo ser tanto de instituições privadas quanto aberto ao público, como o DOI (Mondal; Mondal, 2023), ORCID (Silva, 2021) e o ARK (Koster, 2020).

O DOI é atribuído permanentemente a um objeto, fornecendo um link de rede persistente que direciona a informações atualizadas sobre ele, incluindo sua localização na Internet (Guedes; Shintaku; Brito, 2013). O ORCID, por sua vez, fornece um identificador digital persistente (ORCID iD) que se conecta a informações profissionais do indivíduo, como publicações.

O ARK se destaca no universo dos PIDs por sua notável autossuficiência, permitindo que qualquer organização ou indivíduo crie identificadores sem a necessidade de se submeter a uma autoridade central reguladora. Essa característica única faz com que o ARK seja uma solução ideal para ambientes descentralizados, onde a independência na geração e gerenciamento de

identificadores é importante. Além disso, o ARK é uma ferramenta acessível, pois não impõe taxas para a criação ou manutenção de identificadores, o que o torna uma opção economicamente viável, especialmente em projetos com orçamentos limitados (Kelly et al., 2021b). Para entender melhor como o ARK é estruturado e como ele pode ser aplicado na prática, a Figura 1 apresenta a sua composição.

Figura 1 – Estrutura do ARK



Fonte: Elaborado pelo autor (2024).

A estrutura de um ARK é composta por três partes principais. A primeira parte é o prefixo 'ark:/', que identifica o tipo de identificador. A segunda parte é o número da autoridade nomeadora, que corresponde à organização ou indivíduo responsável pela atribuição do ARK. Por fim, a terceira parte é o próprio nome ARK, que é o identificador exclusivo do recurso.

A autossuficiência do identificador ARK é particularmente vantajosa para sistemas descentralizados, onde a independência e a flexibilidade são essenciais. A ausência de taxas, aliada à flexibilidade dos metadados e à natureza open source do ARK, tornam esse sistema uma solução robusta e atraente para uma ampla gama de aplicações, desde a preservação digital em bibliotecas e arquivos até a gestão de recursos em ambientes de pesquisa científica e acadêmica (Segundo et al., 2023).

Outro aspecto que diferencia o ARK é sua flexibilidade em relação aos metadados (Kelly et al., 2021a). O sistema ARK permite que os usuários adaptem os metadados conforme suas necessidades específicas, oferecendo uma abordagem altamente customizável para a gestão de recursos digitais. Essa flexibilidade é particularmente valiosa em cenários onde os metadados são dinâmicos ou exigem atualizações frequentes. Além disso, a natureza de código aberto do ARK favorece a transparência e a colaboração, permitindo que a comunidade científica e tecnológica participe ativamente de seu desenvolvimento e melhoria contínua (Phillips et al., 2022).

2.2 *Blockchain*

A tecnologia *blockchain* representa um avanço significativo na forma como dados e transações são gerenciados, oferecendo um banco de dados global que é imutável, transparente e confiável (Namasudra et al., 2021). Essa tecnologia foi inicialmente popularizada pelo uso em

criptomoedas (Mahmoud; Lescisin; AlTaei, 2019), como o *Bitcoin*, mas suas propriedades únicas a tornaram aplicável em uma ampla gama de indústrias, desde a gestão da cadeia de suprimentos até sistemas de votação eletrônica (Jafar; Aziz; Shukur, 2021).

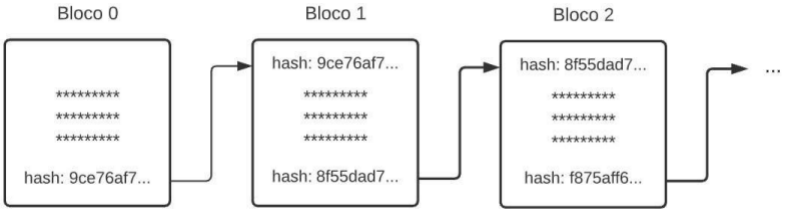
No núcleo da *blockchain* está a capacidade de criar registros descentralizados que não dependem de uma autoridade central para validação (Puthal et al., 2018). Ao contrário dos bancos de dados tradicionais, que são controlados por uma única entidade, as transações em uma *blockchain* são verificadas e validadas por uma rede de usuários distribuídos, conhecidos como *nodes*. Esses *nodes* trabalham em conjunto para garantir a integridade dos dados, tornando a *blockchain* resistente a fraudes e manipulações (Carvalho; Ávila, 2019).

Uma das características mais inovadoras da *blockchain* é a incorporação de *Smart Contracts*, ou Contratos Inteligentes (Taherdoost, 2023). Esses contratos são protocolos autoexecutáveis que contêm os termos do acordo entre comprador e vendedor, diretamente escritos em linhas de código. Os *Smart Contracts* são armazenados na *blockchain*, o que assegura que eles sejam executados automaticamente quando as condições predefinidas são atendidas, sem a necessidade de intermediários (Corrales; Fenwick; Haapio, 2019), resultando em uma maior eficiência e segurança nas transações, além de reduzir custos operacionais.

Além disso, a *blockchain* tem encontrado aplicações em várias áreas. Na indústria financeira, por exemplo, tem sido utilizada para facilitar transações transfronteiriças mais rápidas e baratas (Javaid et al., 2022). No setor de saúde, a *blockchain* está sendo explorada para criar registros médicos imutáveis e acessíveis de maneira segura por diferentes provedores de saúde (Dubovitskaya et al., 2017). Na gestão da cadeia de suprimentos, permite o rastreamento de produtos desde a origem até o consumidor final, aumentando a transparência e reduzindo o risco de contrafação (Sunny; Undralla; Pillai, 2020).

A Figura 2 ilustra a estrutura de uma cadeia de blocos na *blockchain*. Cada bloco na cadeia contém um conjunto de transações e um identificador único, conhecido como *hash*. O primeiro bloco, denominado bloco gênesis, serve como ponto de partida da cadeia. A partir do segundo bloco, cada novo bloco incorpora o *hash* do bloco anterior, criando uma ligação sequencial que assegura a integridade dos dados ao longo de toda a cadeia. Essa interligação entre blocos é fundamental para a segurança da *blockchain*, pois qualquer tentativa de alteração em um bloco resultaria na invalidade de todos os blocos subsequentes (Gouveia, 2021).

Figura 2 – Estrutura de uma cadeia de blocos em uma *blockchain*



Fonte: (Gouveia, 2021).

Na prática, essa estrutura impede que dados sejam manipulados sem que tal alteração seja detectada, reforçando a confiança nas transações registradas na *blockchain*. Como ilustrado na figura, cada bloco é composto por uma lista de transações, seguida pelo *hash* que o conecta ao bloco anterior. Esse processo contínuo de encadeamento de *hashes* cria uma "cadeia"de blocos, daí o nome *blockchain* (Biktimirov et al., 2017).

Contudo, a adoção da *blockchain* enfrenta desafios, como a escalabilidade, o consumo de energia e questões regulatórias. A maioria das *blockchains* atuais, como a do *Bitcoin*, requer um grande poder computacional para validar transações, o que tem gerado preocupações ambientais. Além disso, a falta de uma regulamentação clara em muitos países cria incertezas para empresas que desejam adotar essa tecnologia (Khan; Jung; Hashmani, 2021; Ahl et al., 2022).

Apesar dos desafios associados à escalabilidade, consumo de energia e questões regulatórias, a tecnologia *blockchain* continua a evoluir de maneira significativa. Inovações como as *blockchains* de segunda e terceira geração têm surgido com o objetivo de superar essas limitações, introduzindo melhorias substanciais em termos de eficiência e capacidade. Exemplos notáveis incluem as *blockchains* baseadas em prova de participação (*Proof of Stake*), que oferecem um modelo mais sustentável de validação de transações, e o desenvolvimento de redes interoperáveis, que permitem a comunicação entre diferentes *blockchains*, ampliando as possibilidades de integração e aplicação em diversos setores (Gouveia, 2021; Zheng et al., 2018).

Nesse contexto de inovação, surge o sistema *Decentralized Archival Resource Key* (dARK), que aproveita a robustez e a segurança proporcionadas pela *blockchain* para gerenciar identificadores persistentes de maneira descentralizada e colaborativa. O dARK utiliza a infraestrutura da *blockchain* para garantir a imutabilidade e a rastreabilidade dos dados, assegurando que as informações permaneçam acessíveis e protegidas contra falhas institucionais. Essa integração entre as tecnologias será explorada em maior detalhe na seção seguinte, onde serão discutidas as características específicas e as vantagens do dARK em um ambiente descentralizado.

2.3 *Decentralized Archival Resource Key* (dARK)

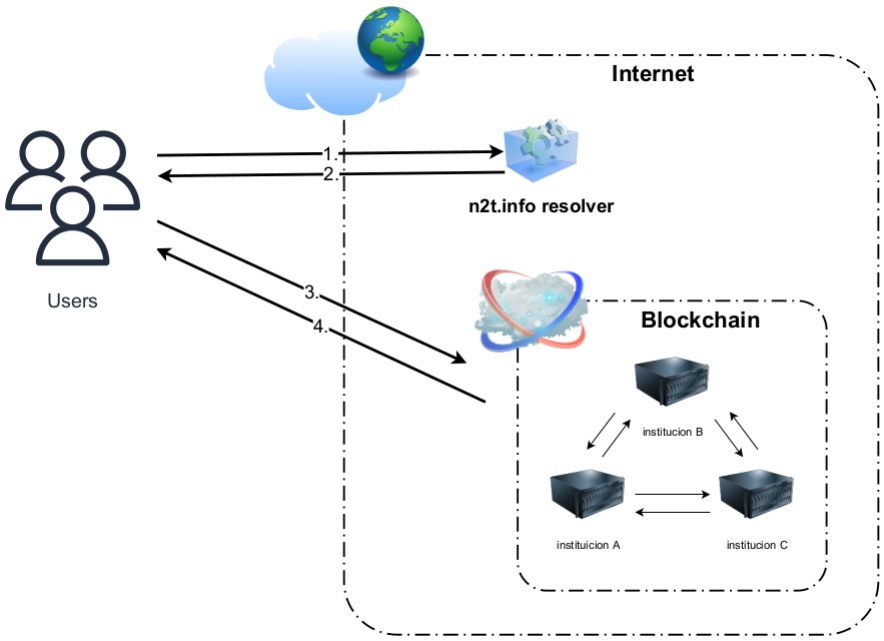
O sistema de identificadores ARK descentralizados, denominado *Decentralized Archival Resource Key* (dARK), representa uma evolução significativa no campo da gestão de identificadores persistentes, especialmente no contexto de preservação digital e acesso contínuo a recursos acadêmicos e científicos. O dARK utiliza uma metodologia que permite a múltiplas instituições gerenciar, de forma colaborativa, seus identificadores ARK em uma infraestrutura descentralizada, baseada em uma rede *blockchain* em consórcio (Segundo et al., 2023). Esse arranjo distribuído não apenas assegura a persistência dos identificadores ARK, mas também facilita a recuperação confiável dos dados através de URIs específicas, mantendo a integridade e a disponibilidade dos dados ao longo do tempo.

Uma das características mais notáveis do dARK é sua arquitetura descentralizada, que alavanca a segurança intrínseca da tecnologia *blockchain*. Nesta arquitetura, os dados associados aos identificadores ARK são replicados de forma segura em múltiplos nós da rede *blockchain*.

Essa replicação distribuída garante que, mesmo em cenários onde uma instituição participante se torna inativa ou deixa de operar, os dados continuam acessíveis e preservados, minimizando riscos de perda de informação crítica. Conforme discutido por (Segundo et al., 2023), o sistema dARK funciona como um ambiente cliente-servidor, onde a *blockchain* age como o servidor descentralizado e o cliente pode ser implementado em uma variedade de tecnologias, permitindo flexibilidade e compatibilidade com diferentes sistemas institucionais.

A Figura 3 ilustra detalhadamente a arquitetura e o funcionamento do dARK. À esquerda, são apresentados os usuários que interagem com o sistema através de uma interface web. Essa interação inicial envolve a submissão de uma requisição ao serviço de resolução de identificadores *n2t.info*, que é responsável por realizar a resolução dos identificadores nos passos 1 e 2. Após a resolução, o sistema faz a ponte com a *blockchain*, que, neste exemplo, está integrada a três diferentes instituições (Instituição A, Instituição B e Instituição C). Cada instituição pode armazenar seu próprio conjunto de dados ou transações na *blockchain*, contribuindo para a descentralização e segurança global do sistema.

Figura 3 – Arquitetura do sistema dARK e seu funcionamento



Fonte: (Segundo et al., 2023).

Além de facilitar a gestão descentralizada de identificadores, o dARK oferece um robusto sistema de rastreamento e auditoria. Cada ação realizada em um PID é registrada na *blockchain*, criando um histórico imutável que documenta quem realizou a ação, o que foi modificado e quando a modificação ocorreu. Essa rastreabilidade é essencial para garantir a transparência e a integridade dos dados geridos pelo sistema dARK. Ela elimina ambiguidades sobre a origem e as modificações dos dados, oferecendo um grau elevado de confiabilidade para as instituições que utilizam a plataforma.

Outro aspecto importante do dARK é sua interoperabilidade. O sistema é projetado para ser compatível com uma ampla gama de tecnologias e infraestruturas institucionais, permitindo que diferentes organizações, independentemente de seu porte ou recursos tecnológicos, possam integrar-se à rede. Essa adaptabilidade torna o dARK uma solução escalável, capaz de atender às necessidades crescentes de gestão de identificadores persistentes em diversos contextos, incluindo ambientes acadêmicos, científicos e industriais. Além disso, a integração com a *blockchain* em consórcio oferece uma camada adicional de segurança e confiabilidade, essencial para a preservação de dados de longo prazo.

Portanto, o dARK se posiciona como uma ferramenta inovadora para a gestão de identificadores persistentes, proporcionando uma infraestrutura descentralizada que combina segurança, flexibilidade e transparência, atendendo às demandas contemporâneas de preservação e acessibilidade de informações digitais.

A implementação do sistema dARK estabelece uma gestão dos identificadores persistentes em um ambiente descentralizado, utilizando a infraestrutura da *blockchain* para garantir a integridade e a acessibilidade dos dados. No entanto, para maximizar o potencial dessa infraestrutura e facilitar a integração com outras plataformas e sistemas, foi desenvolvido o Hyperdrive (Medeiros, 2024). Essa *Application Persistence Interface* (API) atua como uma ponte entre o dARK e os aplicativos externos, permitindo a comunicação eficiente e segura com a *blockchain*. O Hyperdrive, ao expandir as funcionalidades do dARK, oferece um meio estruturado e flexível para a consulta, modificação e gerenciamento de dados armazenados na *blockchain*, integrando-se perfeitamente ao ecossistema descentralizado criado pelo dARK. A seguir, será detalhada a estrutura do Hyperdrive e suas principais funcionalidades, destacando como ele complementa e potencializa as capacidades do dARK.

2.4 Estrutura do Hyperdrive

O Hyperdrive é uma (API), um conjunto de regras ou protocolos que permite a comunicação entre diferentes aplicativos de *software* para a troca de dados, recursos e funcionalidades (Medeiros, 2024). Funcionando do lado do cliente, o Hyperdrive foi projetado para realizar a comunicação com o sistema dARK, conforme discutido na seção anterior.

Este projeto foi desenvolvido em colaboração com a Rede Nacional de Ensino e Pesquisa (RNP) e o Instituto Brasileiro de Informação em Ciência e Tecnologia (IBICT), tendo como objetivo principal promover a obtenção de PIDs armazenados na *blockchain* de maneira eficaz e segura.

A estrutura do Hyperdrive foi concebida para modificar e consultar dados presentes na *blockchain* por meio de identificadores únicos, em colaboração com o sistema dARK. Esse sistema permite o estabelecimento de um ARK ou *hash*, elementos essenciais para a utilização dos métodos do Hyperdrive. Esses identificadores podem ser associados aos seguintes parâmetros:

- ***External URL***: Permite associar um identificador a uma URL externa, facilitando a

referência ou o acesso a recursos relacionados.

- ***External PID***: Habilita a ligação com PIDs externos, aumentando a interoperabilidade com outras infraestruturas.
- ***Payload***: Utilizado para fornecer uma descrição detalhada do objeto ou para incluir informações importantes associadas ao identificador.

No desenvolvimento do Hyperdrive, foi implementado o módulo central, conhecido como *core module*, que inclui três *endpoints* principais:

- ***GET*: recuperar um PID**: Este *endpoint* utiliza um dARK ID como parâmetro de rota para localizar e retornar um recurso correspondente em formato JSON. Se o recurso vinculado ao dARK ID não for localizado, o *endpoint* retornará um erro.
- ***ADD*: adicionar atributos**: Permite a inclusão de atributos a um PID existente, como uma URL externa ou um PID externo. A operação é especificada no corpo da requisição através das flags `"external_pid"` ou `"external_url"` e o resultado é disponibilizado em formato JSON.
- ***SET*: definir atributos**: Usado para reescrever atributos de um PID existente, como a atualização de um *payload* ou uma URL externa. Assim como o *endpoint* ADD, o *endpoint* SET é especificado no corpo da requisição e retorna os resultados em formato JSON.

A Figura 4 apresenta a estrutura do JSON com o parâmetro de URL externa. Conforme descrito por Medeiros (2024), assim que o PID é gerado, ele é considerado um rascunho, aguardando a primeira operação, que deve ser obrigatoriamente um ADD de uma URL externa. Essa abordagem garante a integridade e a eficácia do sistema.

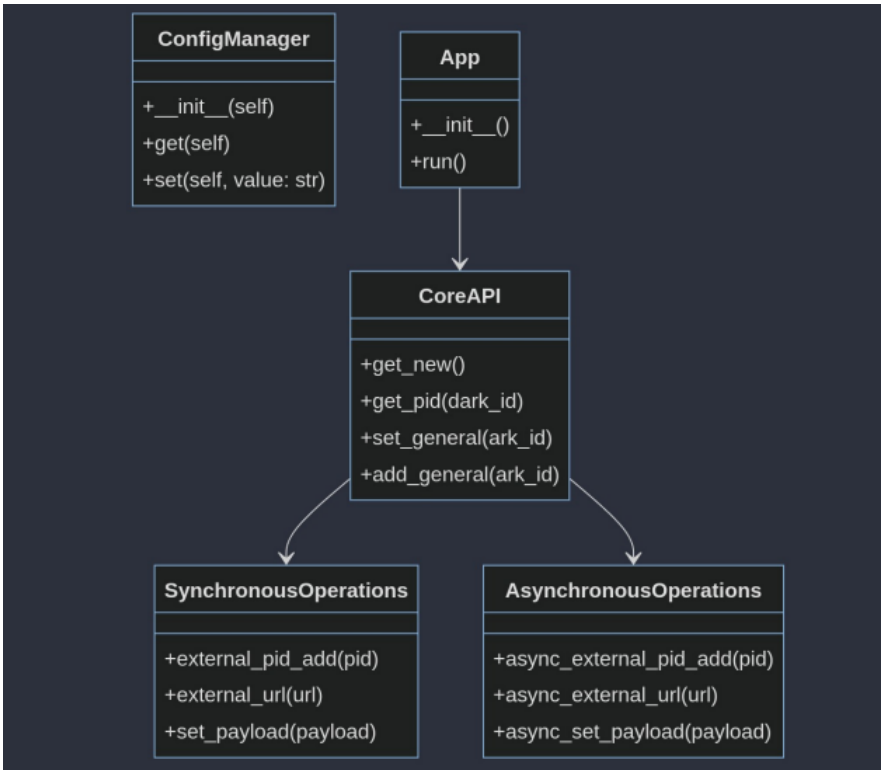
Figura 4 – Representação do envio de *external url*

Fonte: (Medeiros, 2024).

O Hyperdrive permite a recuperação de dados de forma eficiente e descentralizada, com chamadas aos *endpoints* que podem ser realizadas tanto de forma síncrona quanto assíncrona. No modo síncrono, o usuário interage com o sistema em tempo real, garantindo a disponibilidade dos dados, embora as respostas possam demorar mais. Já no modo assíncrono, a interação é mais rápida e permite a realização de múltiplas requisições paralelamente.

Em seguida, a Figura 5 apresenta um diagrama de classe que exemplifica os métodos do Hyperdrive, fornecendo uma visão geral dos principais componentes e funcionalidades do sistema proposto.

Figura 5 – Diagrama de Classe do Hyperdrive



Fonte: (Medeiros, 2024).

A arquitetura do Hyperdrive foi projetada para suportar totalmente requisições via cURL, uma biblioteca amplamente usada para fazer solicitações HTTP, facilitando a integração do Hyperdrive com sistemas já existentes em universidades e periódicos científicos.

A versatilidade do Hyperdrive permite que os usuários configurem os métodos para atender a diferentes necessidades através de variáveis de ambiente (Medeiros, 2024), como descrito a seguir:

- **HYPERDRIVE EXTERNAL PID VALIDATION**: Define o método de validação para PIDs externos, podendo assumir os valores `NONE` (sem validação) ou `BASIC` (validação simples baseada em formatos pré-definidos).
- **HYPERDRIVE URL VALIDATION**: Controla a validação de URLs fornecidas ao sistema, com valores `NONE` e `BASIC`, onde o modo `BASIC` aceita apenas URLs com formatação validada pelo sistema.
- **HYPERDRIVE PAYLOAD VALIDATION**: Define a validação do *payload*, assumindo os valores `NONE` e `BASIC`. No modo `BASIC`, o *payload* deve ser enviado em formato JSON válido.
- **HYPERDRIVE OPERATION MODE**: Define o modo de operação do sistema, podendo ser `SYNC` (síncrono) ou `ASYNC` (assíncrono). O modo assíncrono permite que o sistema execute operações sem bloquear outras atividades.

- **HYPERDRIVE AUTH**: Gerencia a autenticação para métodos que requerem proteção por meio de tokens de acesso. Os valores `TRUE` e `NONE` controlam se a autenticação está ativada ou desativada, respectivamente. Quando ativada, essa variável garante que apenas usuários autorizados possam executar operações críticas, reforçando a segurança do sistema.

A seleção das tecnologias utilizadas no desenvolvimento do Hyperdrive foi criteriosa, com o objetivo de garantir a funcionalidade, segurança e escalabilidade do projeto. As tecnologias empregadas incluem:

- ***Flask***: *Framework* da linguagem *Python* escolhido por sua simplicidade e eficiência no desenvolvimento de APIs. O Flask permite adicionar apenas os recursos necessários, garantindo flexibilidade ao projeto.
- ***PostgreSQL***: Sistema gerenciador de banco de dados (SGBD) objeto-relacional, de código aberto e altamente confiável, ideal para o processamento de grandes volumes de dados.
- ***JSON Web Tokens (JWT)***: Padrão para a transferência segura de informações entre partes, utilizado para autenticação e autorização de usuários.
- ***Docker***: Plataforma para desenvolvimento, envio e execução de aplicativos em contêineres, facilitando a implantação da API.
- ***Git***: Sistema de controle de versão distribuído, essencial para rastreamento e gerenciamento de mudanças no código-fonte.

3 PROPOSTA DE SOLUÇÃO PARA A CAMADA DE SEGURANÇA DO HYPERDRIVE

A segurança é um pilar fundamental para a construção de qualquer sistema, especialmente aqueles que lidam com o armazenamento e gerenciamento de dados sensíveis. Um sistema considerado seguro deve não apenas cumprir os requisitos de segurança especificados, mas também ser capaz de mitigar ameaças imprevistas, operando de maneira confiável tanto em cenários de uso regular quanto diante de tentativas maliciosas de exploração (Uto; Melo, 2009).

No contexto do Hyperdrive, a implementação de mecanismos de segurança é indispensável para assegurar a integridade e proteção dos dados fornecidos pelos usuários, além de impedir que informações sejam registradas na *blockchain* sem a devida autenticação. Embora esta proposta aborde soluções iniciais para a proteção dos endpoints, ela estabelece uma base sólida para o desenvolvimento de uma camada de segurança mais abrangente e eficaz para o Hyperdrive.

3.1 Arquitetura da Camada de Segurança do Hyperdrive

A arquitetura de segurança do Hyperdrive foi projetada para proteger todas as rotas e métodos da API contra acessos não autorizados. Essa proteção é implementada por meio de um *middleware* de autenticação que valida a presença e a validade dos tokens *JSON Web Tokens* (JWT) em cada requisição feita à API.

3.1.1 Funcionamento do Middleware de Autenticação

O *middleware* foi desenvolvido utilizando a biblioteca *Flask-JWT-Extended* (Ukpongson, 2023), que facilita a geração, verificação e gerenciamento de tokens JWT. Estes tokens são fundamentais para garantir que apenas usuários autenticados tenham acesso aos recursos protegidos. A cada requisição, o sistema verifica o cabeçalho da requisição HTTP em busca do token de autenticação (Khorasani; Abdou; Fernández, 2022). Caso o token esteja ausente ou inválido, o sistema impede o acesso ao recurso solicitado e retorna uma mensagem JSON, juntamente com o código de status HTTP 401 (não autorizado).

Para garantir a flexibilidade necessária durante o desenvolvimento e testes, algumas rotas, como a de *login* e rotas públicas, são configuradas para não exigir autenticação. Nesses casos, o *middleware* permite que as requisições passem sem validação de *token*. Contudo, em ambiente de produção, é essencial que a maioria das rotas esteja protegida para assegurar a integridade e confidencialidade dos dados do sistema (Silva, 2009).

Para que essa funcionalidade seja implementada de maneira eficiente, o *middleware* de autenticação verifica se a autenticação está ativada por meio da variável de ambiente *HYPERDRIVE AUTH*. Essa variável controla se o sistema deve exigir um token JWT para as requisições realizadas. Caso a autenticação esteja habilitada, o *middleware* confere se a rota solicitada faz parte da lista de rotas que não requerem autenticação, como a de *login*. Se a rota estiver na lista

de exceções, a requisição é permitida sem a necessidade de validação do token. No entanto, para as demais rotas, o *middleware* exige um token válido e, na ausência de um token ou se o token for inválido, o acesso é negado com uma mensagem de erro e o status HTTP 401. A Figura 6 apresenta o trecho de código que exemplifica esse processo de verificação.

Figura 6 – Função *middleware* para autenticação JWT no Hyperdrive

```
def authentication_middleware():
    USE_AUTH = config_manager.get_hyperdrive_auth()

    if USE_AUTH == "TRUE":
        token = request.headers.get('Authorization')
        non_auth_routes = ['/user/login', '/core/get']

        if request.path in non_auth_routes:
            return

        try:
            jwt_required()(lambda: None)()
        except:
            return jsonify({'message': 'Invalid token'}), 401

        if not token:
            return jsonify({'message': 'Missing authentication token'}), 401
```

Fonte: Elaboração Própria (2024).

Esse código ilustra como o *middleware* atua para verificar as rotas e a presença de tokens JWT. Ao garantir que apenas rotas específicas estejam isentas de autenticação, a aplicação consegue equilibrar a flexibilidade necessária para desenvolvimento e testes, sem comprometer a segurança nas rotas mais sensíveis. Essa abordagem modularizada também facilita futuras atualizações ou a introdução de novas rotas, permitindo uma fácil adaptação sem comprometer a integridade do sistema.

3.1.2 Configuração de Autenticação com Variáveis de Ambiente

A flexibilidade da camada de autenticação do Hyperdrive é gerenciada por meio da variável de ambiente *HYPERDRIVE AUTH*. Esta variável controla se a autenticação via tokens JWT está habilitada ou desabilitada, permitindo a personalização do comportamento de segurança da API conforme o ambiente em que o sistema está sendo executado. Quando a variável *HYPERDRIVE AUTH* é definida como "TRUE", a autenticação via JWT torna-se obrigatória para todas as rotas protegidas, ou seja, o acesso a esses recursos só é permitido a usuários que possuam um token JWT válido. Por outro lado, quando configurada como "FALSE", a camada

de autenticação é desativada, o que permite o acesso irrestrito às rotas da API, sem exigir a validação do token.

Essa configuração torna-se particularmente útil em ambientes de desenvolvimento ou testes, onde os desenvolvedores podem optar por desativar a autenticação para facilitar o fluxo de trabalho e acelerar os processos de depuração. Ao desativar a autenticação, é possível testar funcionalidades sem a necessidade de gerar e validar tokens continuamente, simplificando a simulação de cenários e garantindo que o foco esteja nas funcionalidades principais da aplicação, e não nos mecanismos de segurança.

Essa flexibilidade deve ser usada com extremo cuidado em ambientes de produção. Desativar a autenticação em um cenário real abre as portas para graves vulnerabilidades de segurança, como acessos não autorizados e exposição de dados sensíveis (Büttner; Gruschka, 2024). Sem a camada de proteção proporcionada pelos tokens JWT, seria como deixar uma porta trancada com as chaves penduradas do lado de fora – qualquer pessoa, com ou sem permissão, poderia acessar áreas críticas do sistema, colocando em risco a integridade e a confidencialidade das informações (Shukla et al., 2022). Por esse motivo, é fundamental que a variável *HYPERDRIVE AUTH* esteja sempre configurada como "TRUE"em produção, assegurando que cada requisição passe pelo crivo da autenticação, permitindo acesso apenas aos usuários devidamente autorizados.

Além disso, a configuração de autenticação via variáveis de ambiente oferece uma maneira prática e eficiente de ajustar o comportamento da camada de segurança sem a necessidade de alterar o código diretamente (Lomazina; Surovtsova; Ivanov, 2021). Além de proporcionar uma flexibilidade operacional significativa, permite que as equipes de desenvolvimento e operações modifiquem as políticas de autenticação de forma dinâmica, de acordo com as necessidades do ambiente (Tomasin et al., 2024). Por exemplo, em caso de manutenção ou migração de sistemas, é possível desativar temporariamente a autenticação em um ambiente de teste sem comprometer o código da aplicação. Após a conclusão dos testes, a autenticação pode ser facilmente reativada ao definir a variável de ambiente de volta para "TRUE", restaurando a segurança sem interrupções no fluxo de desenvolvimento.

Essa abordagem baseada em variáveis de ambiente também melhora a consistência entre diferentes ambientes (desenvolvimento, teste e produção), permitindo que cada um seja configurado de acordo com suas necessidades específicas, ao mesmo tempo em que preserva a camada de segurança onde ela é mais necessária.

3.1.3 Funcionamento dos Tokens JWT

O sistema Hyperdrive utiliza tokens JWT como padrão de autenticação e autorização (Jones; Campbell; Mortimore, 2015). Os tokens JWT oferecem uma forma compacta e segura de transmitir informações entre duas partes — o servidor e o cliente —, sendo amplamente utilizados em aplicações modernas por sua eficiência e flexibilidade (Venčkauskas et al., 2023). Um JWT é composto por três partes principais: o *header* (cabeçalho), o *payload* (carga útil) e a *signature* (assinatura) (Shingala, 2019). O *header* contém o tipo de token e o algoritmo

de assinatura, enquanto o *payload* carrega as informações, ou reivindicações, sobre o usuário, como seu ID, permissões e tempo de expiração. Já a *signature* assegura a integridade do token, garantindo que não foi alterado durante a transmissão (Jones; Campbell; Mortimore, 2015).

No Hyperdrive, os tokens JWT são gerados automaticamente após o usuário realizar um login bem-sucedido. Durante o processo de login, o sistema verifica as credenciais fornecidas (como e-mail e senha) e, se essas informações estiverem corretas, um token JWT é emitido e enviado ao cliente. Esse token é armazenado pelo cliente, geralmente no *local storage* ou em cookies, e é incluído nas requisições subsequentes feitas à API como parte do cabeçalho *Authorization*. Cada requisição protegida ao servidor deve incluir esse token para que o sistema possa validar a autenticidade do usuário.

O Hyperdrive estabelece uma política de expiração de tokens, definindo que cada JWT gerado tem uma validade de 6 horas. Após esse período, o token expira automaticamente, exigindo que o usuário faça login novamente para obter um novo token. Essa medida de expiração limita o tempo de acesso autorizado e minimiza o risco de acessos prolongados indevidos, como o uso de um token perdido ou roubado. Além disso, a expiração de tokens JWT melhora a segurança em sistemas que lidam com informações sensíveis, pois reduz o tempo em que um token comprometido pode ser utilizado.

Outro aspecto relevante do JWT no Hyperdrive é sua capacidade de ser assinado digitalmente ou criptografado (Trivedi; Sharma, 2022; How; Heng, 2022). Na maioria dos casos, o JWT é assinado usando algoritmos de chave secreta (como o HMAC) (Naveen; Poongodi, 2023) ou de chave pública/privada (como o RSA) (Imam; Anwer; Nadeem, 2022). A assinatura garante que o token não possa ser modificado por partes mal-intencionadas enquanto estiver em trânsito. No entanto, em cenários que exigem um nível adicional de confidencialidade, o conteúdo do *payload* pode ser criptografado, impedindo que mesmo as informações presentes no token sejam visíveis durante a transmissão. No Hyperdrive, o uso de tokens JWT assinados assegura a integridade dos dados, enquanto a criptografia pode ser aplicada para proteger informações particularmente sensíveis.

Adicionalmente, a natureza compacta dos tokens JWT os torna especialmente adequados para sistemas distribuídos e aplicativos que exigem uma transmissão eficiente de dados. Como são codificados em formato *Base64URL*, os JWT podem ser facilmente serializados e transmitidos por protocolos como HTTP, sem a necessidade de manter informações de sessão no servidor. Tal característica favorece a escalabilidade da aplicação, aliviando a carga sobre o servidor e eliminando a complexidade de gerenciar sessões de usuário, uma vez que toda a informação necessária está contida no próprio token.

Na Figura 7, temos um exemplo de como uma requisição cURL utiliza um token JWT no cabeçalho para autenticar uma solicitação ao Hyperdrive.

Figura 7 – Exemplo de Requisição cURL com Token JWT

```
curl -X POST http://$API_HOST:$API_PORT/core/set/$ARK_ID
-H 'Content-Type: application/json'
-H 'Authorization: Bearer $JWT_TOKEN'
-d '{"external_url":"$valid_url"}'
```

Fonte: Elaboração Própria (2024).

A requisição cURL envia um cabeçalho *Authorization* com um JWT válido, permitindo ao cliente acessar recursos protegidos na API Hyperdrive. O token é transmitido como uma string, precedida pelo identificador "Bearer", que especifica o tipo de token utilizado. A API, por sua vez, valida o token antes de processar a requisição, garantindo que apenas usuários autenticados possam acessar os endpoints.

Essa abordagem com tokens JWT proporciona uma maneira eficiente, segura e escalável de gerenciar a autenticação e a autorização no Hyperdrive, garantindo tanto a proteção dos dados quanto a experiência fluida do usuário.

3.2 Desenvolvimento da Camada de Segurança

O desenvolvimento da camada de segurança do Hyperdrive foi realizado com foco na confiabilidade, escalabilidade e robustez do sistema, utilizando tecnologias amplamente testadas no mercado. A escolha dessas tecnologias foi orientada pela necessidade de garantir a proteção dos dados dos usuários e a integridade das operações, sem comprometer o desempenho da aplicação (Trinder et al., 2017; Bezerra; Koch; Westphall, 2022).

Para o processo de autenticação, optou-se pelo uso de JWT, um padrão amplamente utilizado para garantir que apenas usuários autenticados possam acessar determinados recursos do sistema. O JWT oferece uma solução eficiente para autenticação em aplicações distribuídas, pois permite que o cliente armazene o token e o envie com cada requisição subsequente, eliminando a necessidade de o servidor manter o estado da sessão de cada usuário. Adicionalmente, o JWT pode ser assinado e, opcionalmente, criptografado, oferecendo uma camada extra de segurança (Jánoky; Levendovszky; Ekler, 2018).

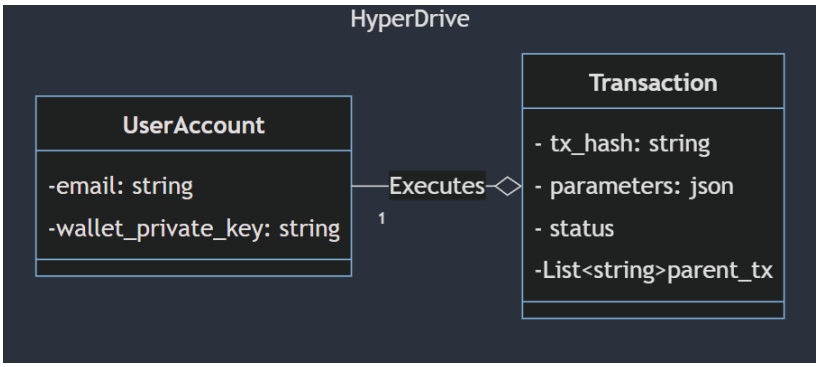
No que se refere ao armazenamento de dados, o *PostgreSQL* foi escolhido como sistema gerenciador de banco de dados (SGBD) (Obe; Hsu, 2017). O PostgreSQL é conhecido por sua robustez e por oferecer suporte a um amplo conjunto de tipos de dados, o que facilita a criação de sistemas de autenticação seguros. Nesse contexto, o banco de dados foi projetado para armazenar as credenciais dos usuários de forma segura, utilizando técnicas de *hashing* para senhas e registros de transações de usuários. O PostgreSQL também facilita a implementação de boas práticas de segurança, como o controle de acesso baseado em funções e a encriptação de dados em repouso (Viloria et al., 2019; Obe; Hsu, 2017).

Além do mais, todo o ambiente de desenvolvimento e produção do Hyperdrive foi configurado utilizando contêineres *Docker* (Docker, 2020). O Docker permite a criação de

ambientes isolados e reproduzíveis, onde todos os componentes da aplicação, incluindo o banco de dados e a API, são executados em contêineres separados. Garantindo a consistência entre os ambientes de desenvolvimento e produção, evitando conflitos de configuração e problemas relacionados a dependências de software. O uso de contêineres também facilita o processo de escalabilidade, permitindo a rápida replicação de ambientes e a manutenção de diferentes versões da aplicação sem causar interrupções no serviço (Miell; Sayers, 2019).

A Figura 8 apresenta o diagrama do banco de dados inicial, que mostra a estrutura básica utilizada para armazenar as credenciais de usuários e as transações realizadas. O banco de dados foi desenvolvido com foco em segurança e eficiência, garantindo que as informações críticas, como senhas e tokens de autenticação, sejam armazenadas de forma segura e estejam protegidas contra acessos indevidos.

Figura 8 – Diagrama do Banco de Dados Inicial do Hyperdrive



Fonte: Elaboração Própria (2024).

O diagrama do banco de dados inicial apresenta duas tabelas principais: a tabela *UserAccount*, que armazena as credenciais de login dos usuários, como e-mail e senha, e a tabela *Transaction*, que registra todas as transações executadas por cada usuário. Essas transações podem incluir, por exemplo, a criação de novos registros na *blockchain*, operações realizadas pelo usuário e outras atividades que envolvam interação com o sistema. No entanto, para o escopo deste trabalho, a tabela *UserAccount* é a mais relevante, pois ela está diretamente relacionada ao sistema de autenticação e à segurança dos *endpoints* da API.

A tabela *UserAccount* foi projetada para armazenar as credenciais de forma segura, utilizando algoritmos de *hashing* como *bcrypt*, que aplicam múltiplas camadas de criptografia para proteger as senhas. Além disso, a tabela armazena outras informações importantes para o gerenciamento de segurança, como a data e a hora do último acesso, permitindo a implementação de políticas de controle de sessão, como a expiração de tokens JWT após um período de inatividade.

O banco de dados também foi configurado para realizar auditorias das transações, registrando logs detalhados sobre cada operação realizada pelos usuários. Incluindo o monitoramento de tentativas de login falhas, o que ajuda a identificar possíveis tentativas de ataques de *brute force* (Apostol, 2012; Stiawan et al., 2019). A auditoria dos registros de acesso e transações

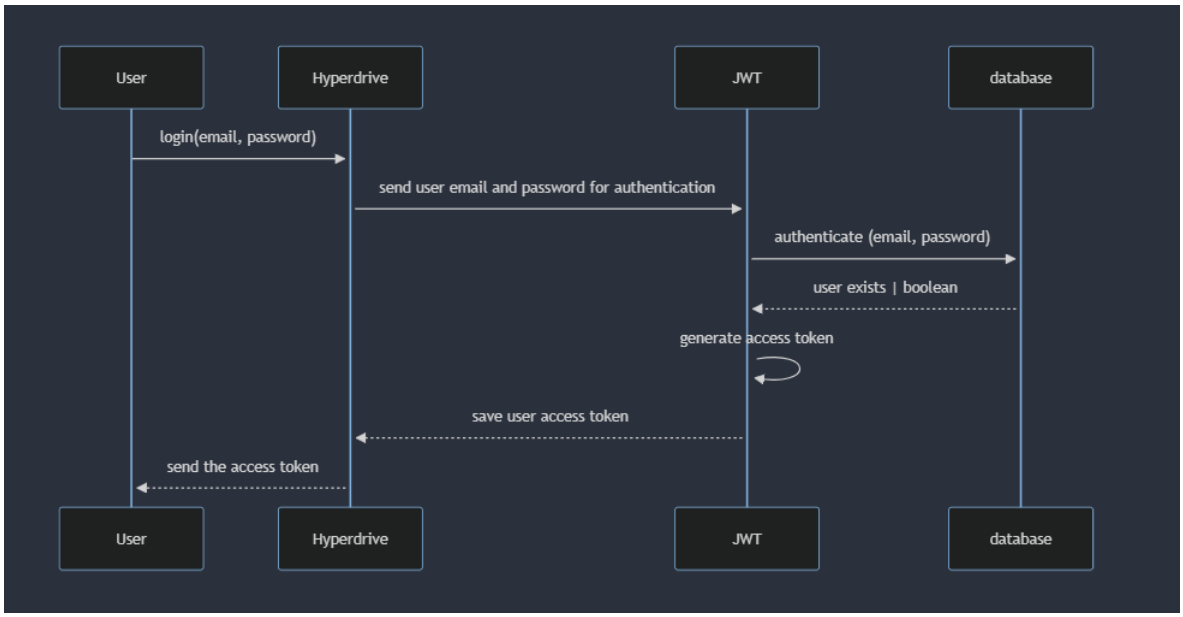
permite que administradores do sistema detectem comportamentos suspeitos e tomem medidas preventivas, como o bloqueio de contas comprometidas ou a geração de alertas automáticos (Kim et al., 2019).

3.3 Integração com o Sistema dARK

A integração entre o Hyperdrive e o sistema dARK foi projetada para garantir a segurança e a eficiência na troca de informações entre os dois sistemas. O Hyperdrive, ao atuar como intermediário para o armazenamento e comunicação de PIDs com a *blockchain*, precisa garantir que as operações realizadas no sistema dARK sigam as mesmas normas de segurança aplicadas ao Hyperdrive, minimizando riscos e assegurando que apenas usuários devidamente autenticados e autorizados possam acessar os recursos sensíveis.

Um dos principais mecanismos de segurança que possibilita essa integração segura é a utilização de tokens JWT, conforme detalhado nas seções anteriores. O processo de autenticação via JWT protege os *endpoints* da API, garantindo que apenas usuários que possuem um token válido possam realizar operações no sistema dARK, como a criação, modificação e consulta de PIDs ou outros dados críticos. O JWT é gerado no Hyperdrive após o login bem-sucedido do usuário, sendo transmitido em todas as requisições subsequentes, o que assegura que as interações entre o Hyperdrive e o dARK sejam autenticadas e seguras, conforme esquematizado na Figura 9.

Figura 9 – Diagrama de Geração de Token



Fonte: Elaboração Própria (2024).

Como mostrado na Figura 9, o processo de autenticação começa quando o usuário tenta acessar um dos *endpoints* protegidos do Hyperdrive. O sistema primeiro verifica as credenciais do usuário no banco de dados, validando informações como e-mail e senha. Se as credenciais

forem válidas, o sistema gera um token JWT que é enviado ao usuário como parte da resposta da requisição. Esse token contém informações essenciais, como o ID do usuário e suas permissões, e é assinado digitalmente para garantir sua integridade.

Uma vez que o token JWT tenha sido emitido, ele é utilizado em todas as requisições subsequentes, sendo incluído no cabeçalho *Authorization* de cada requisição enviada pelo cliente. Quando o usuário tenta acessar um recurso no sistema dARK via Hyperdrive, o token JWT é verificado para garantir que ele ainda seja válido e que o usuário tenha as permissões necessárias para realizar a operação solicitada. Se o token for considerado válido e dentro do prazo de validade de 6 horas, o sistema autoriza a execução da operação no dARK. Caso o token esteja expirado ou inválido, o acesso é negado e o usuário é solicitado a realizar o login novamente.

A validade limitada de 6 horas para cada token JWT garante que o acesso aos sistemas seja constantemente revisado, minimizando o risco de acessos não autorizados devido ao uso prolongado de um token comprometido. Este ciclo de expiração, combinado com a verificação constante da autenticidade do token, oferece um equilíbrio entre a segurança e a usabilidade do sistema, proporcionando uma experiência de autenticação segura sem exigir login constante dos usuários durante o uso normal (Elhejazi; Muragaa, 2024).

Ademais, a integração com o sistema dARK é feita de modo a garantir a sincronização dos dados entre os dois sistemas em tempo real. Dessa forma, qualquer alteração realizada no Hyperdrive, como a criação de um novo PID ou a atualização de um registro, é imediatamente refletida no sistema dARK, assegurando a consistência entre ambos. O uso de tokens JWT possibilita que essa sincronização ocorra com segurança, garantindo que apenas transações autorizadas sejam processadas e que qualquer modificação nos dados seja efetuada por usuários devidamente autenticados.

A comunicação entre o Hyperdrive e o sistema dARK segue rigorosos padrões de segurança estabelecidos para sistemas distribuídos, incluindo o uso de assinaturas digitais nos tokens JWT e criptografia de ponta a ponta. Tal abordagem assegura que as operações ocorram de maneira protegida e sincronizada, garantindo a integridade total dos dados. Dessa forma, o Hyperdrive atua como uma camada confiável de comunicação e segurança, facilitando interações seguras entre os usuários e o sistema dARK.

3.4 Proteção dos Endpoints da API

A proteção dos *endpoints* da API do Hyperdrive é um aspecto fundamental da arquitetura de segurança do sistema. Todos os *endpoints* críticos, que envolvem a manipulação ou consulta de dados sensíveis, são protegidos por uma camada de autenticação que utiliza tokens JWT para validar as requisições (Moen et al., 2024). O JWT é inserido no cabeçalho da requisição HTTP, e a presença e validade desse token são verificadas antes que o servidor permita o acesso ao recurso solicitado. Garantindo que apenas usuários autenticados possam acessar ou modificar dados protegidos, como informações pessoais ou transações no sistema.

A implementação dessa camada de proteção é realizada por meio de um *middleware* de

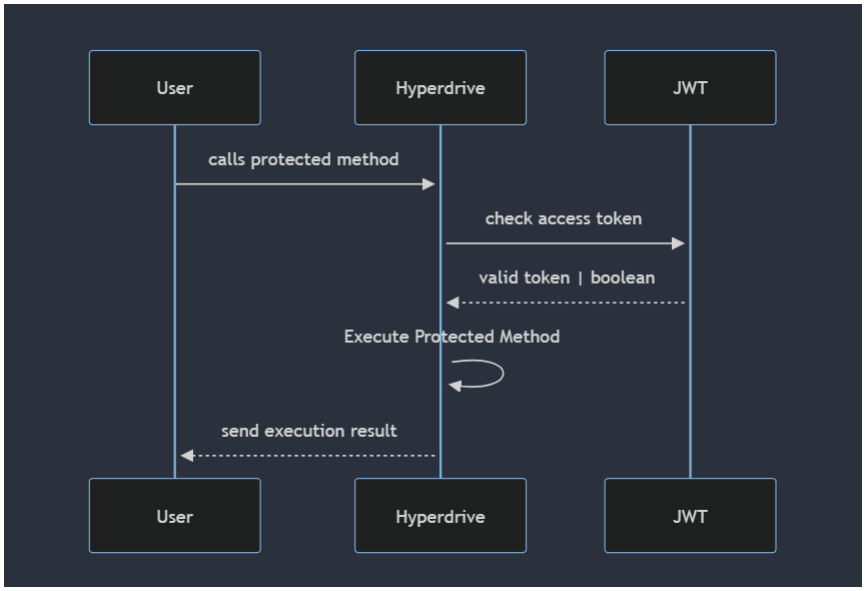
autenticação que intercepta todas as requisições aos *endpoints* protegidos (Nebbione; Calzarossa, 2020). Caso o token esteja presente e seja considerado válido, a requisição é autorizada e processada normalmente. Caso contrário, o servidor retorna uma resposta com um código de erro HTTP 401 (Não Autorizado), informando que a autenticação é necessária para acessar aquele recurso. Outrossim, o sistema pode fornecer mensagens detalhadas de erro, indicando se o token está ausente ou se ele é inválido (por exemplo, expirado ou adulterado).

No entanto, há uma flexibilidade importante incorporada na proteção dos *endpoints* do Hyperdrive: algumas rotas, como as de *login* e de geração de PIDs, estão disponíveis sem a necessidade de autenticação. De tal forma que, permite que os usuários realizem operações como o login inicial ou o registro de dados, sem exigências prévias de autenticação. Essas rotas sem proteção são estrategicamente implementadas para equilibrar usabilidade e segurança, oferecendo facilidade de uso inicial sem comprometer a integridade dos *endpoints* protegidos.

Sobretudo, o uso de variáveis de ambiente para ativar ou desativar a proteção dos *endpoints* proporciona flexibilidade ao sistema. Esse mecanismo permite que o administrador ajuste o nível de segurança de acordo com o ambiente em que o sistema está operando — por exemplo, ambientes de desenvolvimento ou teste podem ter uma política de segurança mais relaxada, enquanto em produção, a autenticação deve ser obrigatória para todos os *endpoints* críticos. O uso de variáveis como *HYPERDRIVE_AUTH* controla essas configurações de autenticação de maneira dinâmica, sem a necessidade de modificar o código da aplicação.

A Figura 10 ilustra o processo de validação de um token JWT ao chamar um método protegido. Quando uma requisição é feita para um *endpoint* protegido, o *middleware* intercepta a requisição e verifica o token JWT contido no cabeçalho da requisição.

Figura 10 – Diagrama de Validação de Token



Fonte: Elaboração Própria (2024).

Se o token for considerado válido (isto é, se não estiver expirado e se a assinatura digital

estiver intacta), a requisição é processada e o método solicitado é executado. Caso o token esteja ausente, inválido ou expirado, o servidor responde com uma mensagem de erro, informando ao usuário que ele precisa de um novo token de autenticação.

A proteção dos *endpoints* da API não se limita apenas à autenticação. Medidas adicionais de segurança também podem ser implementadas para fortalecer ainda mais a camada de proteção, como a limitação de acessos por controle de origem, implementação de políticas de *Cross-Origin Resource Sharing* para controlar o acesso de diferentes domínios, e o uso de criptografia *Transport Layer Security* para garantir que todas as comunicações entre o cliente e o servidor sejam seguras e protegidas contra interceptações.

Nesta proposta de solução, apresentamos os principais elementos da camada de segurança do Hyperdrive, incluindo a implementação do *middleware* de autenticação com tokens JWT, a estrutura do banco de dados para armazenar com segurança as credenciais dos usuários, e a integração com o sistema dARK. Embora se trate de uma solução inicial, ela fornece uma base robusta para a adoção de medidas de segurança mais sofisticadas. Essa abordagem assegura que os dados dos usuários estejam protegidos contra acessos não autorizados e que o sistema seja capaz de resistir a ameaças comuns, como ataques de força bruta, interceptação de dados e manipulação de tokens. Com a evolução do sistema, novas estratégias de segurança, como autenticação multifatorial e a revogação de tokens, poderão ser implementadas para reforçar ainda mais a proteção dos *endpoints*.

4 DESIGN DA PESQUISA

Nesta seção, detalhamos o design da pesquisa utilizado para validar a eficácia do *middleware* de autenticação do Hyperdrive. O design foi estruturado para garantir que todos os aspectos críticos do sistema fossem testados em condições reais de uso, abrangendo tanto cenários de acesso autorizado quanto não autorizado. A seguir, descrevemos os principais elementos do design da pesquisa, incluindo as variáveis de controle, cenários de teste, procedimentos adotados e critérios de avaliação.

4.1 Objetivos do Design

O objetivo central desta pesquisa foi avaliar o comportamento do *middleware* de autenticação do Hyperdrive em diferentes cenários, validando sua capacidade de controlar o acesso a rotas públicas e protegidas, garantindo a segurança do sistema sem comprometer a usabilidade. Especificamente, buscamos responder às seguintes perguntas de pesquisa:

1. O *middleware* permite o acesso irrestrito a rotas públicas sem a exigência de um token JWT?
2. O *middleware* bloqueia corretamente o acesso a rotas protegidas quando não há um token JWT ou quando o token fornecido é inválido?
3. O *middleware* concede acesso a rotas protegidas quando um token JWT válido é fornecido?

4.2 Variáveis de Controle

Para garantir a consistência e a comparabilidade dos resultados, as seguintes variáveis de controle foram estabelecidas:

- **Variável de Configuração de Autenticação**: A variável *HYPERDRIVE AUTH* foi configurada como *TRUE* para garantir que as rotas protegidas exigissem autenticação, enquanto as rotas públicas permanecessem acessíveis sem a necessidade de um token JWT.
- **Ambiente de Teste**: Todos os testes foram realizados em um ambiente controlado, onde as rotas e os parâmetros de autenticação foram previamente definidos e ajustados de acordo com o cenário testado.
- **Tipo de Requisição**: As requisições foram enviadas utilizando *cURL*, com parâmetros padrão para simular interações reais com a API do Hyperdrive.

4.3 Cenários de Teste

Nesta seção, descrevemos os cenários de teste utilizados para validar o comportamento do *middleware* de autenticação do Hyperdrive. Cada cenário foi desenhado para explorar diferentes

aspectos do sistema, garantindo que ele funcione conforme o esperado tanto em condições normais quanto em situações de erro. Os cenários de teste foram divididos em dois grupos principais: rotas não autenticadas e rotas autenticadas, permitindo uma cobertura completa das funcionalidades de autenticação e controle de acesso.

O objetivo principal dos testes foi validar que o sistema seja capaz de identificar corretamente quando uma requisição deve ser autenticada e, ao mesmo tempo, garantir o acesso irrestrito a rotas que não exigem autenticação. A seguir, apresentamos os cenários detalhados.

4.3.1 Cenários de Acesso a Rotas Não Autenticadas

As rotas não autenticadas são aquelas que, por sua natureza, não exigem que o usuário esteja autenticado para acessá-las. Essas rotas desempenham um papel importante na funcionalidade do sistema, especialmente em operações preliminares que envolvem a autenticação ou a consulta de informações públicas. Nos testes de acesso a rotas não autenticadas, o foco foi garantir que o *middleware* permitisse o acesso a essas rotas sem exigir um token JWT, assegurando a usabilidade do sistema.

Os principais cenários testados para rotas não autenticadas incluem:

- **Acesso à rota de login:** A rota de login é uma das mais importantes no sistema, pois permite que o usuário obtenha suas credenciais (token JWT) para acessar rotas protegidas. Este teste verificou se o *middleware* permitia o acesso à rota de login sem exigir um token JWT, visto que o objetivo dessa rota é fornecer as credenciais de acesso.
- **Acesso a informações públicas de um PID:** Outro cenário testado envolveu o acesso a informações públicas associadas a um PID. Essas informações não envolvem dados sensíveis, portanto, o sistema deve garantir que estejam acessíveis sem a necessidade de autenticação. O teste verificou se o *middleware* permitia corretamente o acesso a essas informações públicas sem exigir um token JWT.

O objetivo principal dos testes de rotas não autenticadas foi garantir que o sistema seja flexível o suficiente para permitir o acesso a determinadas operações sem a exigência de autenticação, o que melhora a experiência do usuário e evita sobrecarregar o sistema com verificações desnecessárias. Ao permitir o acesso a rotas como login e consulta de informações públicas, o *middleware* assegura que usuários novos ou não autenticados possam realizar ações preliminares sem serem bloqueados.

Para cada teste, uma requisição *cURL* foi enviada para as rotas públicas, sem o cabeçalho `Authorization` contendo um token JWT. As respostas do sistema foram então analisadas para verificar se o acesso foi permitido conforme o esperado.

O sistema deve permitir o acesso às rotas não autenticadas sem exigir a presença de um token JWT. As respostas esperadas incluem a correta execução das operações solicitadas, como o retorno de um token JWT na rota de login ou a exibição de informações públicas na rota de consulta de PIDs.

Esperava-se que o sistema retornasse os dados corretos sem a necessidade de autenticação para as rotas testadas. Na rota de login, por exemplo, o sistema deve processar as credenciais do usuário e retornar um novo token JWT. Para a consulta de informações públicas, o sistema deve retornar os dados solicitados sem exigir autenticação.

4.3.2 Cenários de Acesso a Rotas Autenticadas

As rotas autenticadas, por sua vez, são aquelas que exigem que o usuário esteja devidamente autenticado para acessá-las. Essas rotas geralmente envolvem operações sensíveis ou a manipulação de dados que requerem proteção adicional. O *middleware* do Hyperdrive foi projetado para proteger essas rotas, garantindo que apenas usuários autenticados possam acessá-las. Nos testes de rotas autenticadas, o foco foi verificar se o sistema bloqueia corretamente o acesso quando não há um token JWT válido ou quando o token fornecido é inválido.

Os cenários testados para rotas autenticadas incluem:

- **Acesso sem token JWT:** Este teste envolveu o envio de uma requisição para uma rota protegida sem incluir um token JWT no cabeçalho da requisição. O objetivo foi verificar se o *middleware* bloqueava corretamente o acesso quando o token de autenticação estava ausente.

- **Acesso com token JWT inválido:** O segundo cenário envolveu o envio de uma requisição com um token JWT inválido. Tokens inválidos podem ser expirados, malformados ou adulterados. O teste buscou garantir que o *middleware* identificasse o token inválido e bloqueasse o acesso.

- **Acesso com token JWT válido:** O último cenário testou o comportamento do sistema quando um token JWT válido foi enviado. O objetivo foi garantir que o sistema permitisse o acesso às rotas protegidas quando um token válido estivesse presente, assegurando que as operações sensíveis fossem realizadas apenas por usuários autenticados.

O objetivo dos testes de rotas autenticadas foi assegurar que o sistema controlasse rigorosamente o acesso a recursos sensíveis, impedindo acessos não autorizados e garantindo que apenas usuários autenticados pudessem acessar e manipular esses recursos.

Para os testes de acesso a rotas protegidas, foram enviadas requisições *cURL* para as rotas que exigem autenticação. O comportamento do sistema foi monitorado para verificar se o acesso foi corretamente bloqueado ou permitido, dependendo da presença e validade do token JWT.

O sistema deve bloquear o acesso a rotas protegidas quando não houver um token JWT ou quando o token for inválido. Apenas requisições com um token JWT válido devem ser permitidas. As mensagens de erro apropriadas (como `"Missing authentication token"` ou `"Invalid token"`) devem ser retornadas nos casos de ausência ou invalidez do token.

Esperava-se que o *middleware* bloqueasse corretamente todas as tentativas de acesso não autorizado, retornando um status HTTP 401 (Não Autorizado) e uma mensagem de erro

adequada. Para requisições com um token JWT válido, o sistema deveria permitir o acesso e processar a operação com sucesso, retornando um status HTTP 200 (OK).

4.4 Procedimentos de Teste

Os procedimentos de teste foram cuidadosamente projetados para garantir que o *middleware* de autenticação do Hyperdrive fosse testado em uma variedade de cenários. A fim de assegurar a integridade dos resultados, os testes foram realizados em um ambiente controlado, utilizando a ferramenta *cURL* para enviar requisições HTTP às rotas do sistema. A ferramenta *cURL* permite simular requisições do cliente para o servidor, possibilitando a criação de testes que imitam o comportamento real dos usuários ao interagir com o sistema.

Os procedimentos de teste foram estruturados para cobrir uma ampla gama de situações que poderiam ocorrer durante o uso do sistema, desde acessos não autenticados até operações protegidas por autenticação JWT (JSON Web Token). Cada cenário foi desenhado para garantir que o *middleware* operasse de acordo com as especificações de segurança e usabilidade, respondendo adequadamente a cada tipo de requisição.

As principais etapas do procedimento incluem:

1. **Configuração do ambiente de teste:** Antes de iniciar os testes, o ambiente foi preparado para simular as condições de produção, garantindo que todos os parâmetros relevantes estivessem adequadamente configurados. A variável *HYPERDRIVE AUTH*, responsável por controlar a exigência de autenticação nas rotas protegidas, foi ativada (`TRUE`) para forçar a verificação de tokens JWT nas rotas que requerem autorização. Além disso, foi estabelecido um banco de dados de teste contendo informações fictícias, com dados de usuários e PIDs (Persistent Identifiers), para simular um cenário realista de acesso às rotas protegidas e públicas.

2. **Envio de requisições simulando diferentes cenários de acesso:** Para cada cenário de teste, requisições *cURL* foram geradas, replicando os comportamentos esperados de usuários reais ao interagir com a API. As requisições incluíram testes com e sem o cabeçalho `Authorization`, que carrega o token JWT, além de testes com tokens válidos e inválidos. As rotas testadas incluíram tanto rotas públicas (que não exigem autenticação) quanto rotas protegidas (que requerem um token JWT válido). A ferramenta *cURL* foi utilizada para simular tanto as requisições legítimas quanto as requisições malformadas, como tokens inválidos ou ausentes.

3. **Análise das respostas do sistema:** Cada requisição enviada foi acompanhada pela coleta e análise das respostas retornadas pelo servidor. Para isso, observamos os seguintes elementos:

    - **Mensagens de erro**: Verificamos se as mensagens de erro retornadas pelo sistema estavam de acordo com os cenários testados. Por exemplo, para requisi-

ções sem autenticação em rotas protegidas, a mensagem esperada era `"Missing authentication token"`, enquanto para tokens inválidos esperava-se a mensagem `"Invalid token"`.

- **Status HTTP**: Os status HTTP retornados também foram analisados, uma vez que eles indicam o sucesso ou falha de uma requisição. Para casos onde o acesso foi bloqueado corretamente, o status esperado era 401 (Não Autorizado), enquanto para requisições bem-sucedidas, o sistema deveria retornar um status 200 (OK).
- **Dados retornados**: Para as rotas públicas, verificamos se o sistema retornava os dados corretos, como informações públicas sobre PIDs, enquanto nas rotas protegidas, o foco foi em garantir que os dados retornados só estivessem acessíveis para usuários autenticados.

4. **Comparação dos resultados obtidos com os resultados esperados:** Cada resposta do sistema foi comparada com os resultados esperados para o cenário em questão. Essa comparação foi fundamental para determinar se o comportamento do *middleware* estava alinhado com as especificações de segurança e design do sistema. Quaisquer discrepâncias entre o comportamento observado e o comportamento esperado foram analisadas para identificar possíveis falhas na implementação do *middleware*.

Durante o processo, todos os dados de saída, incluindo logs e respostas HTTP, foram registrados para análise posterior, o que garantiu que os resultados pudessem ser revisados e validados. Cada etapa do procedimento foi executada múltiplas vezes para garantir a consistência dos resultados e minimizar a possibilidade de falsos positivos ou negativos.

4.5 Critérios de Avaliação

Os critérios de avaliação foram estabelecidos para garantir que os testes cobrissem todos os aspectos relevantes do *middleware* de autenticação, desde a segurança até a usabilidade do sistema. A seguir, detalhamos os principais critérios que foram utilizados para avaliar o sucesso de cada teste:

- **Segurança**: Este foi o critério mais importante de avaliação, uma vez que o principal objetivo do *middleware* é garantir a proteção de rotas sensíveis. O sistema foi avaliado com base em sua capacidade de bloquear corretamente acessos não autorizados e impedir que usuários sem um token JWT válido acessassem rotas protegidas. Além disso, testamos se o *middleware* rejeitava tokens expirados ou malformados, garantindo que apenas usuários autenticados pudessem acessar os recursos protegidos. A segurança foi considerada satisfatória quando:
  - A ausência de um token JWT resultava no bloqueio de acesso às rotas protegidas, com retorno de um status HTTP 401.

  - O fornecimento de um token JWT inválido era detectado e resultava no bloqueio do acesso, com retorno de uma mensagem de erro apropriada.
  - O fornecimento de um token JWT válido permitia o acesso correto às rotas protegidas, sem erros.

- **Usabilidade**: Além de proteger os recursos sensíveis, o sistema também deve ser usável e flexível. A avaliação de usabilidade focou na verificação de que o *middleware* permitia o acesso irrestrito às rotas públicas, como as rotas de login e consulta de dados públicos sobre PIDs. A experiência do usuário não deve ser comprometida por exigências desnecessárias de autenticação para rotas que, por definição, não envolvem dados sensíveis. A usabilidade foi considerada adequada quando:
  - O sistema permitia o acesso a rotas públicas sem exigir um token JWT.
  - Operações simples, como login e consulta de dados públicos, podiam ser realizadas sem bloqueios indevidos.

- **Conformidade com o Design**: Para garantir que o sistema estivesse implementado de acordo com as especificações de design, o comportamento do *middleware* foi comparado com o design originalmente proposto. Esse critério visava garantir que o *middleware* estivesse separando corretamente as rotas públicas e protegidas e aplicando as regras de autenticação de maneira consistente. A conformidade foi considerada satisfatória quando:
  - As rotas públicas estavam corretamente configuradas para permitir acesso sem autenticação.
  - As rotas protegidas exigiam autenticação válida, bloqueando corretamente acessos não autorizados.

Esses critérios de avaliação garantiram que o sistema estivesse funcionando conforme esperado e que tanto a segurança quanto a usabilidade fossem preservadas. Cada cenário de teste foi avaliado com base nesses critérios, e quaisquer falhas identificadas durante os testes foram analisadas e corrigidas para garantir o bom funcionamento do sistema.

5 RESULTADOS E DISCUSSÕES

Para garantir a eficácia do *middleware* de autenticação desenvolvido para o *Hyperdrive*, foram realizados diversos testes manuais em diferentes cenários. Esses testes visam validar o comportamento do *middleware* tanto em situações de acesso autorizado quanto não autorizado, assegurando que ele funcione conforme o esperado. Os casos de teste analisados abrangem desde a verificação de rotas públicas, que não exigem autenticação, até rotas protegidas, que requerem tokens de autenticação válidos para acesso. A seguir, detalhamos os principais casos de teste, descrevendo os objetivos, procedimentos, resultados esperados e obtidos para cada um.

5.1 Acesso a Rotas Não Autenticadas

O objetivo principal deste teste foi verificar a capacidade do *middleware* de permitir o acesso a rotas não autenticadas sem exigir um token JWT. Essas rotas incluem operações que, por sua natureza, não envolvem manipulação de dados sensíveis ou transações que exigem proteção, como a rota de *login* e a rota de consulta de informações públicas sobre um PID. A variável *HYPERDRIVE AUTH* foi configurada como *"TRUE"*, de forma que as rotas protegidas exigissem autenticação, enquanto as rotas públicas permanecessem acessíveis sem a necessidade de fornecer um token de autenticação.

Essa configuração é essencial para garantir que usuários novos ou não autenticados possam realizar ações preliminares, como efetuar login ou consultar dados de acesso público, sem serem bloqueados pelo sistema de segurança. A implementação correta dessa flexibilidade é crítica, pois evita sobrecarregar as operações simples com requisitos desnecessários de autenticação, o que pode prejudicar a usabilidade e a eficiência do sistema.

5.1.1 Teste de Acesso à Rota de Login

Um dos primeiros testes realizados foi a verificação da rota de *login*, uma rota essencial que não deve requerer autenticação prévia, uma vez que o próprio objetivo desta rota é fornecer as credenciais do usuário para obter um token de acesso válido. A Figura 11 apresenta a requisição cURL enviada para essa rota, onde o usuário fornece suas credenciais (email e senha). Nenhum token JWT foi enviado com a requisição, conforme esperado para essa operação.

Figura 11 – Requisição cURL de login sem token

```
curl -X POST http://$API_HOST:$API_PORT/user/login -H
'Content-Type: application/json' -d
'{"email":"valid_email", "password" : "valid_password" }'
```

Fonte: Elaboração Própria (2024).

O resultado esperado era que o sistema processasse a requisição de login sem exigir um token JWT, retornando um novo token ao usuário após validar suas credenciais. Como previsto, o teste foi bem-sucedido, com o *middleware* permitindo o acesso à rota de *login* e o sistema retornando um token JWT válido para o usuário. Esse comportamento garante que novos usuários possam autenticar-se no sistema sem a necessidade de uma autenticação prévia, assegurando o fluxo correto de login.

Essa verificação confirma que a lógica de *middleware* foi corretamente configurada para permitir acessos a rotas não autenticadas quando necessário, sem comprometer a segurança das operações críticas, como o acesso a dados sensíveis.

5.1.2 Teste de Acesso a Informações Públicas de um PID

Além da rota de *login*, o sistema também deve garantir o acesso irrestrito a dados públicos, como informações básicas sobre um PID, sem exigir um token de autenticação. Esse cenário se aplica a qualquer consulta feita a informações que não envolvem dados privados ou transações sensíveis. A Figura 12 ilustra uma requisição cURL para obter informações públicas de um PID específico.

Figura 12 – Requisição cURL para informações públicas de um identificador

```
curl -X POST http://$API_HOST:$API_PORT/core/get/$ARK_ID
```

Fonte: Elaboração Própria (2024).

Neste caso, a requisição cURL foi enviada sem um token de autenticação, conforme o esperado para esse tipo de operação. A resposta esperada seria o retorno de informações públicas associadas ao PID, como a URL externa e outros dados descritivos que não requerem proteção adicional. O objetivo deste teste era garantir que o sistema não bloqueasse o acesso a essas informações públicas por falta de um token JWT.

O resultado obtido confirmou as expectativas, com o sistema retornando as informações públicas corretamente. A funcionalidade foi validada com sucesso, indicando que o *middleware* foi configurado adequadamente para reconhecer e permitir o acesso a rotas públicas sem interferir com a necessidade de autenticação.

5.1.3 Implicações dos Resultados

Os resultados dos testes de acesso a rotas não autenticadas demonstram que o *middleware* do *Hyperdrive* foi corretamente configurado para distinguir entre rotas que exigem autenticação e rotas públicas que podem ser acessadas sem um token JWT. Essa separação é essencial para garantir uma experiência de usuário eficiente e intuitiva, permitindo que usuários realizem tarefas como login e consulta de informações públicas sem encontrar barreiras desnecessárias.

Além disso, os testes reforçam a importância de configurar corretamente a variável *HYPERDRIVE AUTH*, permitindo flexibilidade para alternar entre diferentes políticas de autenticação dependendo do ambiente (desenvolvimento ou produção). Em ambientes de produção, a autenticação deve ser sempre exigida para rotas protegidas, mas a capacidade de manter rotas públicas acessíveis sem um token é fundamental para a usabilidade do sistema.

Outro ponto importante a ser destacado é que o acesso irrestrito a informações públicas contribui para a transparência e acessibilidade, especialmente em contextos onde a disseminação de dados abertos é fundamental. No entanto, é necessário garantir que apenas as informações designadas como públicas estejam disponíveis sem autenticação, enquanto todos os dados sensíveis devem ser devidamente protegidos.

Os testes realizados confirmam que o *Hyperdrive* está preparado para lidar adequadamente com ambos os cenários, garantindo uma segurança eficaz sem sacrificar a usabilidade. No próximo conjunto de testes, a atenção será voltada para a verificação do comportamento do *middleware* em rotas que exigem autenticação e proteção adicional.

5.2 Acesso a Rotas Autenticadas

Nesta seção, abordamos os testes realizados para validar o comportamento do *middleware* de autenticação nas rotas protegidas do *Hyperdrive*. O principal objetivo foi garantir que o *middleware* bloqueie corretamente o acesso a essas rotas quando não houver um token JWT válido ou quando o token fornecido for inválido. Também foram realizados testes para verificar o comportamento do sistema quando um token JWT válido é utilizado, assegurando que o acesso às rotas protegidas seja concedido apenas a usuários autenticados.

A proteção de rotas é essencial para garantir que recursos sensíveis da aplicação sejam acessados apenas por usuários autorizados, preservando a integridade e segurança do sistema. Assim, os testes realizados focaram em três cenários principais: acesso sem token, acesso com um token inválido e acesso com um token válido. Cada um desses cenários foi cuidadosamente analisado para verificar se o *middleware* respondia corretamente às solicitações de acesso, bloqueando ou concedendo o acesso conforme necessário.

5.2.1 Acesso sem Token de Autenticação

O primeiro cenário testado foi o acesso a uma rota protegida sem fornecer um token JWT no cabeçalho da requisição. Esse teste é fundamental, pois o sistema deve impedir que usuários sem autenticação acessem rotas que manipulam dados ou executam operações sensíveis.

Para simular esse cenário, foi realizada uma requisição cURL para configurar uma URL externa, uma rota que exige autenticação. A Figura 13 ilustra essa requisição. Como esperado, o *middleware* identificou a ausência do token JWT e bloqueou o acesso à rota.

Figura 13 – Requisição cURL com Token JWT para rota autenticada

```
curl -X POST http://$API_HOST:$API_PORT/core/set/$ARK_ID
-H 'Content-Type: application/json'
-H 'Authorization: Bearer $JWT_TOKEN'
-d '{"external_url":"$valid_url"}'
```

Fonte: Elaboração Própria (2024).

O comportamento esperado era que o sistema retornasse a seguinte resposta JSON:

```
{"message": "Missing authentication token"}
```

Além disso, o status HTTP retornado deveria ser 401 (Não Autorizado), o que indica que a requisição foi negada devido à ausência do token de autenticação. O resultado obtido confirmou as expectativas: o *middleware* impediu corretamente o acesso à rota protegida e retornou a mensagem de erro correspondente. Esse resultado demonstra que o sistema está configurado para exigir autenticação nas rotas críticas, reforçando a segurança do *Hyperdrive*.

O bloqueio adequado do acesso sem token é uma medida crucial para evitar acessos não autorizados e garantir que as operações realizadas nas rotas protegidas sejam executadas apenas por usuários devidamente autenticados. Este comportamento também assegura que, em um ambiente de produção, o sistema não ficará vulnerável a tentativas de invasão ou manipulação de dados sem a devida autenticação.

5.2.2 Acesso com Token Inválido

O segundo teste envolveu o envio de uma requisição com um token JWT inválido. O objetivo foi verificar se o *middleware* identificaria corretamente um token inválido e bloquearia o acesso à rota protegida. Tokens inválidos podem surgir por vários motivos, incluindo tokens expirados, adulterados ou gerados incorretamente. Testar esse cenário é importante para garantir que o sistema não aceite tokens comprometedores e mantenha o controle estrito de acesso.

Neste caso, foi feita uma requisição com um token JWT incorreto, e o comportamento esperado era que o *middleware* retornasse a seguinte resposta JSON:

```
{"message": "Invalid token"}
```

O status HTTP esperado também era 401 (Não Autorizado), indicando que a autenticação falhou devido ao token inválido. O *middleware* respondeu corretamente, bloqueando o acesso e retornando a mensagem de erro apropriada. Esse teste confirmou que o *middleware* é capaz de detectar e rejeitar tokens inválidos, impedindo que usuários não autorizados acessem recursos protegidos.

Essa validação é essencial para impedir que tokens manipulados ou expirados sejam usados para acessar o sistema, reforçando a segurança e a integridade das operações. Ao identificar

corretamente tokens inválidos, o sistema mantém o controle rigoroso sobre quem pode realizar operações sensíveis, assegurando que apenas usuários com permissões válidas possam acessar as rotas protegidas.

5.2.3 Acesso com Token Válido

O terceiro teste envolveu o envio de uma requisição com um token JWT válido. O objetivo era garantir que o sistema, ao receber um token correto, permitisse o acesso às rotas protegidas, processando a requisição de forma bem-sucedida.

Neste cenário, foi realizada uma requisição cURL para adicionar uma URL externa, uma operação que requer autenticação. A Figura 13 ilustra o envio dessa requisição com um token JWT válido. O comportamento esperado era que o *middleware* validasse o token e permitisse o acesso à rota, processando a operação conforme solicitado.

O status HTTP retornado foi 200 (OK), indicando que a operação foi realizada com sucesso e que o *middleware* permitiu corretamente o acesso à rota protegida. O sistema retornou a resposta esperada, confirmando que o token foi validado com êxito e que a requisição foi autorizada. Esse resultado demonstra que o *middleware* foi configurado corretamente para permitir o acesso a usuários autenticados, garantindo que operações protegidas possam ser realizadas apenas por aqueles que possuem as permissões necessárias.

Esse teste validou a eficácia da autenticação por JWT no *Hyperdrive*, demonstrando que o sistema processa adequadamente as requisições de usuários autenticados, enquanto mantém a proteção de dados sensíveis.

5.2.4 Conclusão dos Testes de Acesso a Rotas Protegidas

Os testes realizados para verificar o acesso a rotas protegidas confirmaram que o *middleware* de autenticação do *Hyperdrive* funciona corretamente em diferentes cenários, bloqueando o acesso quando não há um token ou quando o token é inválido, e permitindo o acesso apenas quando um token JWT válido é fornecido.

Essa funcionalidade é essencial para garantir a segurança do sistema, impedindo acessos não autorizados a dados sensíveis e assegurando que apenas usuários autenticados possam realizar operações protegidas. Os resultados obtidos evidenciam que o *middleware* foi corretamente configurado para manter o controle rigoroso sobre quem pode acessar e modificar recursos críticos no *Hyperdrive*, assegurando a integridade e a confiabilidade das operações.

6 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

No contexto acadêmico, onde a produção e disseminação de conhecimento são centrais, a gestão eficaz de PIDs se torna indispensável. Com o aumento do volume de pesquisas e publicações, há uma demanda crescente por uma infraestrutura que não apenas suporte essa complexidade, mas que também ofereça eficiência e segurança. O desenvolvimento do *Hyperdrive*, ao integrar a tecnologia *blockchain*, responde de forma inovadora a essas necessidades, estabelecendo uma base sólida para a gestão de PIDs de maneira segura e escalável.

O *Hyperdrive* se destaca pela utilização de tecnologias como o *framework Flask* e a linguagem de programação *Python*, escolhidas por sua eficiência no desenvolvimento ágil de APIs. O *PostgreSQL* foi adotado como banco de dados por ser uma solução de código aberto confiável, capaz de lidar com grandes volumes de dados de forma eficiente. Esses componentes tecnológicos garantem que o *Hyperdrive* esteja bem equipado para atender às demandas atuais, mas também preparado para evoluir conforme as necessidades futuras do ambiente acadêmico. Os resultados mostraram que:

- **QP1.** Os testes confirmaram que o *middleware* foi configurado corretamente para permitir o acesso irrestrito a rotas públicas que não exigem autenticação. A verificação ocorreu em cenários como a rota de login e a consulta de informações públicas sobre PIDs, onde o sistema processou as requisições sem exigir um token JWT. Esse comportamento garante que usuários não autenticados possam acessar essas rotas sem encontrar barreiras desnecessárias, assegurando a usabilidade do sistema para operações não sensíveis;

- **QP2.** O *middleware* bloqueia corretamente o acesso a rotas protegidas em ambos os cenários: quando o token JWT está ausente e quando o token fornecido é inválido. Nos testes de acesso sem token, o sistema retornou a mensagem de erro `"Missing authentication token"` com o status HTTP 401, conforme esperado. Da mesma forma, quando um token JWT inválido foi enviado, o *middleware* rejeitou o acesso, retornando a mensagem `"Invalid token"` e o mesmo status 401. Esses resultados demonstram que o sistema está configurado para impedir acessos não autorizados, preservando a segurança das rotas protegidas.

- **QP3.** O *middleware* concede corretamente o acesso a rotas protegidas quando um token JWT válido é fornecido. Nos testes em que um token JWT válido foi enviado, o sistema processou as requisições com sucesso, retornando um status HTTP 200 (OK) e permitindo o acesso às rotas protegidas. Esse comportamento assegura que apenas usuários autenticados e autorizados possam acessar os recursos e realizar operações sensíveis no sistema, mantendo a integridade das operações.

As respostas para as questões de pesquisa confirmam que o *middleware* de autenticação do Hyperdrive foi implementado de forma eficaz, atendendo aos requisitos de segurança e

usabilidade. O sistema mostrou-se capaz de diferenciar corretamente entre acessos autorizados e não autorizados, bloqueando corretamente as tentativas não autenticadas e permitindo o acesso a usuários com credenciais válidas.

Diante desse resultado, são propostas as seguintes recomendações para trabalhos futuros:

- Com o crescimento do *Hyperdrive*, será essencial realizar atualizações regulares no banco de dados para acomodar novos tipos de dados e volumes maiores de informações. Essas atualizações devem incluir a implementação de novas camadas de segurança para assegurar a integridade das informações armazenadas e prevenir possíveis violações. Adicionalmente, a otimização do desempenho do banco de dados será necessária para manter a eficiência à medida que o sistema se expande;

- Dado o aumento previsto no volume de dados, é recomendável o desenvolvimento de estratégias robustas de *backup*. Permitindo a realização de *backups* regulares, que podem ser completos (*snapshots* de todo o banco de dados) ou incrementais (apenas as alterações desde o último *backup*). Essas práticas garantirão a recuperação de dados em caso de falhas ou perda de informações, protegendo a continuidade do serviço. A automatização desses processos de *backup* deve ser considerada para minimizar erros humanos e assegurar a confiabilidade dos dados;

- Para melhorar a experiência do usuário e facilitar a interação com a API, a criação de um *frontend* é uma evolução natural do projeto. Uma interface amigável permitirá que os usuários acessem facilmente as funcionalidades do sistema, além de melhorar a apresentação e a formatação dos dados retornados pelos *endpoints*.O sistema se tornará não apenas mais acessível, mas também proporcionará uma experiência de uso mais intuitiva e agradável, o que deve resultar em maior adoção e engajamento por parte dos usuários;

- Embora a versão atual do *Hyperdrive* ofereça uma camada de segurança robusta para os *endpoints*, ainda há espaço para melhorias significativas. Uma das recomendações para o futuro é a implementação de autenticação multifator. Esse método adicionaria uma camada extra de segurança, exigindo que os usuários, além de fornecerem e-mail e senha, também insiram um código enviado para seu e-mail. Essa medida reduziria ainda mais o risco de acessos não autorizados, fortalecendo a proteção do sistema contra ameaças cibernéticas em constante evolução.

As recomendações apresentadas visam não apenas garantir a segurança e a funcionalidade atuais, mas também preparar o sistema para os desafios futuros. Ao implementar essas melhorias, o *Hyperdrive* poderá não apenas manter sua relevância, mas também expandir suas capacidades, atendendo de forma ainda mais eficaz às demandas do ecossistema acadêmico em constante mudança.

REFERÊNCIAS

AHL, A. et al. Challenges and opportunities of blockchain energy applications: Interrelatedness among technological, economic, social, environmental, and institutional dimensions. *Renewable and Sustainable Energy Reviews*, Elsevier, v. 166, p. 112623, 2022. Citado na página 19.

APOSTOL, K. *Brute-force attack*. [S.l.]: SaluPress, 2012. Citado na página 30.

BEZERRA, W. d. R.; KOCH, F.; WESTPHALL, C. B. A comparative qualitative and quantitative analysis of the performance of security options for message protocols: Fog computing scenario. *arXiv preprint arXiv:2210.12917*, 2022. Citado na página 29.

BIKTIMIROV, M. et al. Blockchain technology: universal structure and requirements. *Automatic Documentation and Mathematical Linguistics*, Springer, v. 51, p. 235–238, 2017. Citado na página 19.

BÜTTNER, A.; GRUSCHKA, N. Evaluating the influence of multi-factor authentication and recovery settings on the security and accessibility of user accounts. *arXiv preprint arXiv:2403.15080*, 2024. Citado na página 27.

CARVALHO, C. A. de; ÁVILA, L. V. A tecnologia blockchain aplicada aos contratos inteligentes. *Revista Em Tempo*, v. 18, n. 01, p. 156–176, 2019. Citado na página 18.

CORRALES, M.; FENWICK, M.; HAAPIO, H. *Legal tech, smart contracts and blockchain*. [S.l.]: Springer, 2019. Citado na página 18.

DOCKER, I. Docker. *línea].[Junio de 2017]. Disponible en: https://www. docker. com/what-docker*, 2020. Citado na página 29.

DUBOVITSKAYA, A. et al. Secure and trustable electronic medical records sharing using blockchain. In: AMERICAN MEDICAL INFORMATICS ASSOCIATION. *AMIA annual symposium proceedings*. [S.l.], 2017. v. 2017, p. 650. Citado na página 18.

ELHEJAZI, M. F.; MURAGAA, W. H. A. Improving the security and reliability of sdn controller rest apis using json web token (jwt) with openid and oauth2.0. In: *2024 IEEE 4th International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering (MI-STA)*. [S.l.]: IEEE, 2024. p. 398–402. Citado na página 32.

GAUS, N. Selecting research approaches and research designs: A reflective essay. *Qualitative Research Journal*, Emerald Publishing Limited, v. 17, n. 2, p. 99–112, 2017. Citado na página 15.

GOUVEIA, L. D. Blockchain. Universidade Federal de Campina Grande, 2021. Citado 2 vezes nas páginas 18 e 19.

GUEDES, M. G.; SHINTAKU, M.; BRITO, R. F. d. Atribuição de identificadores digitais para publicações científicas: Doi para o seer/ojs. Instituto Brasileiro de Informação em Ciência e Tecnologia (IBICT), 2013. Citado na página 16.

HARDISTY, A. et al. A choice of persistent identifier schemes for the distributed system of scientific collections (dissco). *Research Ideas and Outcomes*, Pensoft Publishers, v. 7, p. e67379, 2021. Citado na página 16.

HOW, H.-B.; HENG, S.-H. Blockchain-enabled searchable encryption in clouds: a review. *Journal of Information Security and Applications*, Elsevier, v. 67, p. 103183, 2022. Citado na página 28.

HUYNH-THE, T. et al. Blockchain for the metaverse: A review. *Future Generation Computer Systems*, Elsevier, v. 143, p. 401–419, 2023. Citado na página 12.

IMAM, R.; ANWER, F.; NADEEM, M. An effective and enhanced rsa based public key encryption scheme (xrsa). *International Journal of Information Technology*, Springer, v. 14, n. 5, p. 2645–2656, 2022. Citado na página 28.

JAFAR, U.; AZIZ, M. J. A.; SHUKUR, Z. Blockchain for electronic voting system—review and open research challenges. *Sensors*, MDPI, v. 21, n. 17, p. 5874, 2021. Citado na página 18.

JÁNOKY, L. V.; LEVENDOVSZKY, J.; EKLER, P. An analysis on the revoking mechanisms for json web tokens. *International Journal of Distributed Sensor Networks*, SAGE Publications Sage UK: London, England, v. 14, n. 9, p. 1550147718801535, 2018. Citado na página 29.

JAVAID, M. et al. A review of blockchain technology applications for financial services. *BenchCouncil Transactions on Benchmarks, Standards and Evaluations*, Elsevier, v. 2, n. 3, p. 100073, 2022. Citado na página 18.

JONES, M.; CAMPBELL, B.; MORTIMORE, C. *Json web token (jwt) profile for oauth 2.0 client authentication and authorization grants*. [S.l.], 2015. Citado 2 vezes nas páginas 27 e 28.

KELLY, M. et al. Advancing arks in the historical ontology space. *Code4Lib Journal*, n. 50, 2021. Citado na página 17.

KELLY, M. et al. Archival resource keys for collaborative historical ontology publication. In: PUBPUB. *Proceedings of the ICTeSSH 2021 conference*. [S.l.], 2021. Citado 2 vezes nas páginas 12 e 17.

KHAN, D.; JUNG, L. T.; HASHMANI, M. A. Systematic literature review of challenges in blockchain scalability. *Applied Sciences*, MDPI, v. 11, n. 20, p. 9372, 2021. Citado na página 19.

KHORASANI, M.; ABDOU, M.; FERNÁNDEZ, J. H. Authentication and application security. In: *Web Application Development with Streamlit: Develop and Deploy Secure and Scalable Web Applications to the Cloud Using a Pure Python Framework*. [S.l.]: Springer, 2022. p. 203–227. Citado na página 25.

KIM, J. et al. Insider threat detection based on user behavior modeling and anomaly detection algorithms. *Applied Sciences*, MDPI, v. 9, n. 19, p. 4018, 2019. Citado na página 31.

KLEIN, M.; BALAKIREVA, L. On the persistence of persistent identifiers of the scholarly web. In: SPRINGER. *Digital Libraries for Open Knowledge: 24th International Conference on Theory and Practice of Digital Libraries, TPDL 2020, Lyon, France, August 25–27, 2020, Proceedings 24*. [S.l.], 2020. p. 102–115. Citado na página 12.

KOSTER, L. Persistent identifiers for heritage objects. *Code4Lib Journal*, n. 47, 2020. Citado 2 vezes nas páginas 12 e 16.

LOMAZINA, T. A.; SUROVTSOVA, T. G.; IVANOV, D. A. Development of a cryptocurrency iot wallet with automatic authentication. In: IEEE. *2021 International Conference on Quality Management, Transport and Information Security, Information Technologies (IT&QM&IS)*. [S.l.], 2021. p. 318–323. Citado na página 27.

MAHMOUD, Q. H.; LESCISIN, M.; ALTAEI, M. Research challenges and opportunities in blockchain and cryptocurrencies. *Internet Technology Letters*, Wiley Online Library, v. 2, n. 2, p. e93, 2019. Citado na página 18.

MEDEIROS, S. A. *Hyperdrive: uma api para o dARK*. 2024. Trabalho de Conclusão de Curso (Graduação em Computação) - Universidade Estadual da Paraíba, Patos, 2024. Disponível em: http://dspace.bc.uepb.edu.br/jspui/handle/123456789/32064. Citado 4 vezes nas páginas 13, 21, 22 e 23.

MIELL, I.; SAYERS, A. *Docker in practice*. [S.l.]: Simon and Schuster, 2019. Citado na página 30.

MOEN, A. et al. *Securing API Authentication and Authorisation with Integration of Digital Identities*. Dissertação (Trabalho de Conclusão de Curso) — Norwegian University of Science and Technology (NTNU), 2024. Citado na página 32.

MONDAL, H.; MONDAL, S. Digital object identifier: What it is and why it matters? *Indian Journal of Skin Allergy*, Scientific Scholar, v. 2, n. 2, p. 77–80, 2023. Citado 2 vezes nas páginas 12 e 16.

NAMASUDRA, S. et al. The revolution of blockchain: State-of-the-art and research challenges. *Archives of Computational Methods in Engineering*, Springer, v. 28, p. 1497–1515, 2021. Citado na página 17.

NAVEEN, P.; POONGODI, A. Development of secure framework in mobile cloud computing using aes-hmac encryption approach. In: SPRINGER. *International Conference on Advancements in Smart Computing and Information Security*. [S.l.], 2023. p. 192–206. Citado na página 28.

NEBBIONE, G.; CALZAROSSA, M. C. Security of iot application layer protocols: Challenges and findings. *Future Internet*, v. 12, n. 3, p. 55, 2020. Citado na página 33.

OBE, R. O.; HSU, L. S. *PostgreSQL: up and running: a practical guide to the advanced open source database*. [S.l.]: "O'Reilly Media, Inc.", 2017. Citado na página 29.

OKUNE, A.; CHAN, L. Digital object identifier: Privatising knowledge governance through infrastructuring. In: *Routledge Handbook of Academic Knowledge Circulation*. [S.l.]: Routledge, 2023. p. 278–287. Citado na página 12.

PHILLIPS, M. et al. Session 3gl persistent identifiers: Using archival resource keys (arks) to keep it all together. Texas Digital Library, 2022. Citado na página 17.

PLOMP, E. Going digital: Persistent identifiers for research samples, resources and instruments. *Data Science Journal*, v. 19, p. 46–46, 2020. Citado na página 12.

PUTHAL, D. et al. The blockchain as a decentralized security framework [future directions]. *IEEE Consumer Electronics Magazine*, IEEE, v. 7, n. 2, p. 18–21, 2018. Citado na página 18.

RAMACHANDRAN, R.; BUGBEE, K.; MURPHY, K. From open data to open science. *Earth and Space Science*, Wiley Online Library, v. 8, n. 5, p. e2020EA001562, 2021. Citado na página 12.

ROSS, S. M.; MORRISON, G. R. Experimental research methods. In: *Handbook of research on educational communications and technology*. [S.l.]: Routledge, 2013. p. 1007–1029. Citado na página 15.

SAYÃO, L. F. Interoperabilidade das bibliotecas digitais: o papel dos sistemas de identificadores persistentes-urn, purl, doi, handle system, crossref e openurl. *Transinformação*, SciELO Brasil, v. 19, p. 65–82, 2007. Citado na página 16.

SEGUNDO, W. et al. *dARK: A decentralized blockchain implementation of ARK Persistent Identifiers*. [S.l.], 2023. Citado 5 vezes nas páginas 12, 13, 17, 19 e 20.

SHINGALA, K. Json web token (jwt) based client authentication in message queuing telemetry transport (mqtt). *arXiv preprint arXiv:1903.02895*, 2019. Citado na página 27.

SHUKLA, A. et al. System security assurance: A systematic literature review. *Computer Science Review*, Elsevier, v. 45, p. 100496, 2022. Citado na página 27.

SIEDLECKI, S. L. Understanding descriptive research designs and methods. *Clinical Nurse Specialist*, v. 34, n. 1, p. 8–12, 2020. Citado na página 15.

SILVA, C. A. d. Gestão da segurança da informação: um olhar a partir da ciência da informação. PUC-Campinas, 2009. Citado na página 25.

SILVA, J. A. Teixeira da. Orcid: Issues and concerns about its use for academic purposes and research integrity. *Annals of Library and Information Studies (ALIS)*, v. 67, n. 4, p. 246–250, 2021. Citado 2 vezes nas páginas 12 e 16.

SINGH, S.; HOSEN, A. S.; YOON, B. Blockchain security attacks, challenges, and solutions for the future distributed iot network. *Ieee Access*, IEEE, v. 9, p. 13938–13959, 2021. Citado na página 13.

STIAWAN, D. et al. Investigating brute force attack patterns in iot network. *Journal of Electrical and Computer Engineering*, Wiley Online Library, v. 2019, n. 1, p. 4568368, 2019. Citado na página 30.

SUNNY, J.; UNDRALLA, N.; PILLAI, V. M. Supply chain transparency through blockchain-based traceability: An overview with demonstration. *Computers & Industrial Engineering*, Elsevier, v. 150, p. 106895, 2020. Citado na página 18.

TAHERDOOST, H. Smart contracts in blockchain technology: A critical review. *Information*, MDPI, v. 14, n. 2, p. 117, 2023. Citado na página 18.

TOMASIN, S. et al. Analysis of challenge-response authentication with reconfigurable intelligent surfaces. *IEEE Transactions on Information Forensics and Security*, IEEE, 2024. Citado na página 27.

TRINDER, P. et al. Scaling reliably: Improving the scalability of the erlang distributed actor platform. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, ACM New York, NY, USA, v. 39, n. 4, p. 1–46, 2017. Citado na página 29.

TRIVEDI, U. B.; SHARMA, S. Digitally signed document chain (dsdc) blockchain. In: SPRINGER. *Proceedings of Third International Conference on Computing, Communications, and Cyber-Security: IC4S 2021*. [S.l.], 2022. p. 715–727. Citado na página 28.

UKPONGSON, M. Rest api for a store management system using flask. Sumy State University, 2023. Citado na página 25.

UTO, N.; MELO, S. P. de. Vulnerabilidades em aplicações web e mecanismos de proteção. *Sociedade Brasileira de Computação*, 2009. Citado na página 25.

VALENTINO, A. L.; JUANICO, J. F. Overcoming barriers to applied research: A guide for practitioners. *Behavior Analysis in Practice*, v. 13, n. 4, p. 894–904, 2020. Citado na página 14.

VENČKAUSKAS, A. et al. Enhancing microservices security with token-based access control method. *Sensors*, MDPI, v. 23, n. 6, p. 3363, 2023. Citado na página 27.

VILORIA, A. et al. Integration of data mining techniques to postgresql database manager system. *Procedia Computer Science*, Elsevier, v. 155, p. 575–580, 2019. Citado na página 29.

W3C. *URI Clarification*. 2001. Acessado em: 04 maio 2024. Disponível em: https://www.w3.org/TR/uri-clarification/. Citado na página 16.

YAZDINEJAD, A. et al. Decentralized authentication of distributed patients in hospital networks using blockchain. *IEEE journal of biomedical and health informatics*, IEEE, v. 24, n. 8, p. 2146–2156, 2020. Citado na página 13.

ZAIDAN, F. H. et al. Wittgenstein e o significado dos nomes na web semântica. *Ciência da Informação*, v. 47, n. 3, 2018. Citado na página 16.

ZHENG, Z. et al. Blockchain challenges and opportunities: A survey. *International journal of web and grid services*, Inderscience Publishers (IEL), v. 14, n. 4, p. 352–375, 2018. Citado na página 19.

ANEXO A – TERMO DE OUTORGA E ACEITAÇÃO DE BOLSA

TERMO DE OUTORGA E ACEITAÇÃO DE BOLSA

A **REDE NACIONAL DE ENSINO E PESQUISA - RNP**, associação civil qualificada como Organização Social pelo Decreto nº 4.077 de 09 de janeiro de 2002, com sede na Rua Lauro Müller nº 116 sala 1.101 a 1.104, Botafogo, Rio de Janeiro (RJ), CEP 22290-906, inscrita no CNPJ sob o número 03.508.097/0001-36, Inscrição Municipal nº 02.838.109, aqui designada simplesmente **OUTORGANTE**, neste ato representada pela sua colaboradora da área de recursos humanos, com delegação de competência nº 164, concede a **Francileudo da Silva Oliveira** inscrito(a) no CPF sob o número [redacted] doravante denominado(a) **BOLSISTA**, a bolsa-auxílio especificada no presente Termo, mediante cláusulas e condições a seguir:

CLÁUSULA PRIMEIRA - DO OBJETO

1.1 Constitui objeto deste Termo de Outorga, o apoio pela OUTORGANTE ao(à) BOLSISTA, por meio do financiamento de bolsa-auxílio de pesquisa e desenvolvimento, seguindo as regras estabelecidas no *"Regulamento do Programa de Bolsas de Incentivo à PD&I da RNP"*, para atividades de desenvolvimento do Projeto **dARK: Aplicacão blockchain para atribuição de identificadores persistentes ARK**, de acordo com o *Plano de Trabalho* em anexo.

1.2 O projeto em questão:

(x) foi selecionado por meio de edital público.
( ) foi aprovado para execução com recursos de Lei de Incentivo.
( ) é parte de um acordo de cooperação firmado entre a RNP e outra instituição do Sistema Nacional de Ciência e Tecnologia.
( ) é um projeto de PD&I induzido pela própria RNP.

1.3 O(a) BOLSISTA terá como Orientador(a) do Projeto o(a) prof. **Washington Segundo**.

1.4 A vigência da bolsa será de: 01/08/2023 até 31/03/2024

1.5 A bolsa terá o valor de **R$ 800,00 (OITOCENTOS REAIS)** mensal.

CLÁUSULA SEGUNDA – DAS RESPONSABILIDADES DO(A) BOLSISTA

2.1 Cumprir, com todo empenho e interesse, toda programação estabelecida em seu *Plano de Trabalho*.

2.2 Manter conduta compatível com a ética e a probidade administrativa nas atividades inerentes a bolsa.

2.3 Observar, obedecer e cumprir as normas internas da OUTORGANTE, preservando o sigilo e a confidencialidade das informações que tiver acesso.

2.4 Apresentar, nos prazos que lhe forem determinados, informações ou documentos referentes ao desenvolvimento do *Plano de Trabalho*.

2.5 Apresentar documentos comprobatórios da regularidade da sua situação escolar, sempre que solicitado pela OUTORGANTE.

2.6 Informar de imediato, qualquer alteração na sua situação escolar, tais como: trancamento de matrícula, abandono, conclusão de curso ou transferência de Instituição de Ensino.

2.7 Propor, quando julgar necessário, alterações em seu *Plano de Trabalho*, sujeitas à prévia análise e autorização do(a) Orientador(a) e do Coordenador de P&D da RNP designado para acompanhar o projeto.

2.8 Elaborar e enviar mensalmente, com a anuência do(a) Coordenado(a) Acadêmico(a), um *Relatório de Acompanhamento de Atividades*, que deve conter um resumo das tarefas realizadas pelo bolsista.

2.9 Em caso de publicação de resultados provenientes do projeto, é obrigatório fazer referência ao nome do projeto e ao apoio da RNP em rubrica "agradecimentos" (ou *acknowledgements*).

CLÁUSULA TERCEIRA – DAS RESPONSABILIDADES DA OUTORGANTE

3.1. Zelar pelo cumprimento do presente Termo.

3.2. Designar um coordenador de P&D que seja funcionário de seu quadro de pessoal, para orientar e acompanhar o(a) BOLSISTA no desenvolvimento das atividades do projeto.

3.3. Efetuar o pagamento da bolsa-auxílio diretamente ao(à) BOLSISTA.

3.4. Entregar, por ocasião do término da vigência da bolsa, certificado de participação no projeto de P&D.

3.5. Manter em arquivo e à disposição da fiscalização os documentos firmados que comprovem a relação de bolsa.

3.6. Proporcionar ao(à) BOLSISTA condições para o exercício das atividades práticas previstas no plano de trabalho do projeto, tais como o financiamento de despesas decorrentes de viagens (passagens, estadias e alimentação), mediante prévia aprovação do Coordenador de P&D da RNP, ficando as despesas enquadradas nas mesmas bases das viagens realizadas pelos funcionários da RNP, nos termos fixados em sua Norma específica.

3.6.1. Poderá ser aplicado regras específicas para o financiamento e prestação de contas de viagens, dependendo da origem dos recursos que financiam o projeto de pesquisa e desenvolvimento.

CLÁUSULA QUARTA – DO TRATAMENTO DA PROPRIEDADE INTELECTUAL

4.1 Todo o conhecimento gerado a partir do que for desenvolvido durante o período de concessão da bolsa e/ou execução do projeto, passível de proteção ou não no Instituto Nacional da Propriedade Industrial (INPI), será de propriedade da RNP, reconhecendo a autoria do(a) BOLSISTA no processo de tratamento da propriedade intelectual gerada, quando houver;

4.2 O(a) BOLSISTA compromete-se a verificar, em tempo hábil, se a execução do projeto produz ou poderá produzir resultado potencialmente, no todo ou em parte, objeto de proteção por Patente de Invenção, Modelo de Utilidade, Desenho Industrial, Registro de Software ou qualquer outra forma de proteção por direitos de Propriedade Intelectual;

4.2.1. No caso das atividades realizadas originarem resultados materiais ou criações intelectuais passíveis de proteção, a RNP recomenda que

os resultados sejam divulgados, sob qualquer forma, somente após o protocolo de pedido de proteção no INPI, para que o requisito de novidade seja mantido.

4.3 Ao (à) BOLSISTA é vedado prestar qualquer informação a terceiro sobre a documentação técnica envolvida ou segredos de negócio, salvo com consentimento prévio da RNP.

CLÁUSULA QUINTA – DA RESCISÃO

5.1 A bolsa-auxílio cessará em momento anterior ao estipulado, pela ocorrência de um dos seguintes motivos:

a) Descumprimento reiterado, por uma das partes, das suas obrigações.

b) Mútuo acordo das partes ou alteração das circunstâncias.

c) Prestação de falsas declarações.

d) Conclusão antecipada do plano de atividades.

e) Desistência do(a) BOLSISTA, comunicada à RNP com a antecedência mínima de 30 dias.

f) Constituição de vínculo empregatício do(a) BOLSISTA com a RNP.

CLÁUSULA SEXTA – DA VIGÊNCIA

6.1 O presente Termo cessa após o período de vigência fixado na cláusula primeira, salvo se a bolsa for renovada por meio de Termo Aditivo.

6.2 Sempre que a bolsa for renovada, cabe ao Coordenador de P&D da RNP validar e aprovar as entregas previstas no Plano de Trabalho.

CLÁUSULA SÉTIMA – DAS DISPOSIÇÕES FINAIS

7.1 Fica ressalvado que poderão ocorrer atrasos no pagamento da bolsa mensal de forma justificada, em razão do atraso na liberação de recursos dos projetos, devendo a OUTORGANTE, comunicar ao(à) BOLSISTA sua ocorrência, não configurando nesse caso, em descumprimento contratual, tampouco causa de rescisão contratual.

7.2 Fica expressa e inequivocamente entendido pelas partes que o presente Termo não gera qualquer relação de emprego entre a RNP e o(a) BOLSISTA, não lhe sendo aplicável a legislação trabalhista, reconhecendo-se que a atividade aqui regulamentada, será exercida sem subordinação e sem o ânimo definitivo próprio do vínculo empregatício, nos termos do artigo 9º, §1º e 4º, da Lei nº 10.973/2004, tendo característica jurídica de doação, para fins previdenciários e tributários.

7.3 É permitido ao(à) BOLSISTA o recebimento de complementação financeira proveniente de outras fontes, desde que não configure falta às regras estabelecidas pela sua instituição sede e por outras fontes pagadoras, tais como agências de fomento, ficando o(a) BOLSISTA responsável por estar em conformidade com as leis e regras aplicáveis.

7.4 As partes ajustam a possibilidade de revisão deste Termo, por meio de Termo Aditivo, em caso de ocorrência de acontecimentos novos, imprevisíveis pelas partes e a elas não-imputáveis, tais como, crise econômica no pais, rompimento de contrato com a instituição financiadora do projeto, seja devido a falência, inadimplência ou outro motivo que reflita sobre a economia ou na execução das atividades do projeto, para ajustá-lo às circunstâncias supervenientes, no que diz respeito, ao período, metas e valor da bolsa previstos neste Termo.

Rio de Janeiro, 07 de Junho de 2023

Franciledo da Silva Oliveira

BOLSISTA

ORIENTADOR(A)

GESTOR DO PROJETO NA RNP

OUTORGANTE (Representante da área de RH da RNP)