Utilização de IPFS para Armazenamento Seguro em Sistema de Atendimento Psicológico

Jayane Xavier Gomes[1], Fabio Castro Araujo[1]

[1]Departamento de Computação
Universidade Luterana do Brasil – Palmas – TO

jayanexv@rede.ubra.br, fabio.araujo@ulbra.br

Abstract. *This work presents the development of a backend service dedicated to the storage of clinical data in psychological telecare, applied to the ONTerapia system, an application aimed at organizing the psychologist's clinical routine. The solution is based on the InterPlanetary File System (IPFS) and an audit layer for operation traceability. The proposed approach structures the insertion and retrieval flow of clinical records in a decentralized environment, with replication across nodes and systematic event logging for subsequent verification. The results obtained in a controlled scenario indicate the feasibility of storing, retrieving, and auditing medical records and attachments, with a clear separation between sensitive content and control metadata.*

Resumo. *Este trabalho apresenta o desenvolvimento de um serviço de backend dedicado para armazenamento de dados clínicos no teleatendimento psicológico, aplicado ao sistema ONTerapia, um aplicativo voltado à organização da rotina clínica do psicólogo. A solução é fundamentado no InterPlanetary File System (IPFS) e em uma camada de auditoria para rastreabilidade das operações. A proposta organiza o fluxo de inserção e recuperação de registros clínicos em ambiente descentralizado, com replicação entre nós e registro sistemático de eventos para verificação posterior. Os resultados obtidos em cenário controlado indicam viabilidade para armazenar, recuperar e auditar prontuários e anexos, com separação entre conteúdo sensível e metadados de controle.*

1. Introdução

A pandemia de COVID-19 fragilizou a saúde mental da população ao instaurar um contexto de incerteza e insegurança, quadro que culminou em sofrimento e adoecimento psíquico, intensificados pelo isolamento social e pelas perdas familiares, em que provocou a sobrecarga dos sistemas de saúde, bem como a limitação de recursos para a assistência na área (FINOKETTI et al., 2022; SILVA, 2023). Estudos recentes sugerem que a modalidade híbrida de atendimento psicológico, que integra encontros online e presenciais, desponta como alternativa viável, ao conciliar a praticidade proporcionada pelas tecnologias digitais com a profundidade clínica própria do contato presencial (GONÇALVES; FERREIRA NETO, 2023; BRASIL; BORBA, 2023)

Nesse cenário, a telepsicologia revelou-se uma alternativa acessível, capaz de reduzir desigualdades no acesso à saúde mental, atender grupos impossibilitados de acompanhamento presencial e garantir a manutenção da relação terapêutica (FINOKETTI et al., 2022; SILVA, 2023). De acordo com Sablone et al. (2024), a pandemia de COVID-19 impulsionou a adoção de atendimentos psicológicos por meios digitais, o que assegurou a continuidade do cuidado e ampliou o acesso durante o distanciamento social. Assim, a telepsicologia passou a compor, de modo estruturado, as estratégias contemporâneas de cuidado em saúde mental.

Conforme Domingues, Augustini e Oliveira (2024), o avanço da tecnologia e a crescente dependência de sistemas informatizados transformaram a segurança da informação em um desafio cada vez maior. Os bancos de dados são alvos constantes de ataques

cibernéticos, que podem resultar em danos significativos para as organizações, como o vazamento de informações estratégicas e a perda de dados e prejuízos de ordem financeira. Estudos recentes apontam que falhas de controle de acesso e configurações inseguras estão entre os principais fatores que expõem essas bases a invasões e perdas de informações sensíveis (ZHONG, 2023).

Nesse contexto, a literatura tem demonstrado que vazamentos em bancos centralizados não resultam apenas de ataques externos, mas também de ameaças internas, como uso indevido de privilégios ou ausência de segregação de funções (LI et al., 2023; RUBIRA, 2025). Relatórios recentes de segurança confirmam que incidentes de violação de dados têm crescido em ambientes centralizados, exigindo práticas de gestão de acessos privilegiados, segmentação e monitoramento contínuo como estratégias indispensáveis para mitigar tais riscos (IBM; HARVARD BUSINESS REVIEW, 2024).

Diante desse quadro, em conformidade com a LGPD, a confidencialidade em saúde impõe medidas técnicas e administrativas de proteção, bem como o controle de acesso às informações e a comunicação, em prazo razoável, de incidentes relevantes à Autoridade Nacional de Proteção de Dados (ANPD) e aos titulares com as informações mínimas exigidas. Além disso, por envolver dados sensíveis, as informações relacionadas à saúde possuem tratamento jurídico diferenciado, sendo o seu gerenciamento e compartilhamento condicionados a hipóteses legais específicas, bem como sujeitos a restrições expressas, inclusive quanto à vedação de utilização para fins de vantagem econômica (BRASIL, 2018/2024).

Como solução, adota-se o uso do *InterPlanetary File System* (IPFS), um sistema distribuído de armazenamento e compartilhamento de arquivos baseado em uma arquitetura descentralizada, como base para um repositório de dados resilientes, capaz de manter a disponibilidade e a integridade da informação mesmo sob a falha de nós individuais. Essa característica é sustentada pelo endereçamento por conteúdo, realizado por meio de *Content Identifiers* (CIDs), que consistem em identificadores criptográficos derivados do próprio conteúdo armazenado, o que garante integridade verificável e imutabilidade dos dados (LINHARES, 2024; IPFS, 2025a).

Adicionalmente, os dados no IPFS são organizados segundo a estrutura *Merkle Directed Acyclic Graph* (Merkle DAG), na qual os objetos são interligados por *hashes* criptográficos, que permite a verificação da integridade de cada bloco e favorece a deduplicação e a mitigação de pontos únicos de falha no armazenamento distribuído (IPFS, 2025b; IPFS, 2025c). Em conjunto com esses mecanismos, o sistema incorpora criptografia de conteúdo em nível de aplicação e autenticação segura, de modo a proteger os pontos de controle dos nós e reforçar a confidencialidade e a segurança das informações armazenadas. Desse modo, o fluxo de inserção e recuperação evidencia a separação entre dados sensíveis (*off-chain*, no IPFS) e metadados/provas (*on-chain*), esse modelo favorece a escala, além de preservar a confidencialidade em cenários de alto volume de prontuários (AZBEG; OUCHETTO; ANDALOUSSI, 2022).

Diante do exposto, o objetivo geral deste trabalho consiste em desenvolver um microserviço de armazenamento descentralizado baseado no IPFS, com a incorporação de mecanismos de controle de acesso e trilhas de auditoria por meio de registros de atividades e eventos, voltado ao armazenamento de dados sensíveis relacionados ao reconhecimento de emoções e anamneses no contexto da saúde digital, aplicado ao sistema ONTerapia, um aplicativo destinado à centralização e organização da rotina clínica do psicólogo, que integra recursos de inteligência artificial para apoio à análise de registros clínicos e suporte ao processo terapêutico no ambiente de teleatendimento psicológico.

2. Referencial Teórico

2.1. Atendimento Psicológico Online

No Brasil, a regulamentação específica do atendimento psicológico mediado por tecnologias teve marco inicial com a Resolução CFP nº 11/2018, que disciplinou os serviços por TICs (Tecnologias da Informação e Comunicação) e instituiu exigências como o cadastro no e-Psi (Psicologia Digital) (CFP, 2018). Durante a pandemia de Covid-19, a Resolução CFP nº 04/2020 flexibilizou condições para o atendimento remoto, preservando a observância ao Código de Ética (CFP, 2020). Em 2024, a Resolução CFP nº 9/2024 revogou as normas anteriores (11/2018 e 04/2020) e passou a regular de forma permanente o exercício profissional mediado por TDICs (CFP, 2024).

Nesse novo escopo, a Resolução CFP nº 9/2024 estabelece a responsabilidade técnica e ética da(o) psicóloga(o) na escolha e no uso das TDICs. Define, ainda, a obrigatoriedade de identificar o profissional e a pessoa atendida, bem como a garantia de condições de qualidade, privacidade e sigilo, em observância integral ao Código de Ética (CFP, 2024). O sigilo profissional constitui um dever basilar, e sua quebra é cogitável somente nas hipóteses legais e com justificativa técnica e ética, conforme o Código de Ética Profissional do Psicólogo (CFP, 2005).

O tratamento de informações no atendimento online envolve dados pessoais sensíveis e, por isso, deve observar a Lei Geral de Proteção de Dados Pessoais (Lei nº 13.709/2018) quanto às bases legais, à transparência e às medidas de segurança (BRASIL, 2018). Na prática, em conformidade com a Resolução nº 9/2024, recomenda-se a utilização de plataformas com criptografia e controle de acessos, o estabelecimento de termos claros sobre a finalidade e a guarda dos dados, a manutenção de prontuário seguro e a definição de procedimentos para incidentes (CFP, 2024).

2.2. Segurança da Informação em Sistemas de Saúde

A segurança dos dados e a privacidade na saúde são pilares fundamentais e representam um desafio global necessário para o avanço da saúde digital. O compartilhamento de informações clínicas é importante para a sustentabilidade do setor, pois aumenta a eficiência e a resolutividade no atendimento, com potencial para reduzir custos. Contudo, a Estratégia Global sobre Saúde Digital 2020-2025 da OMS (Organização Mundial da Saúde) destaca que uso de tecnologias digitais na área da saúde deve ser desenvolvido com princípios de privacidade, segurança e confidencialidade, sendo necessário que os países estabeleçam uma base legal e regulatória para proteger a privacidade, confidencialidade, integridade e disponibilidade dos dados na saúde (PELINSON, 2022).

A segurança da informação é, de maneira geral, orientada por princípios fundamentais que formam a base para a proteção de ativos digitais, conhecidos como a tríade CIA (Confidencialidade, Integridade, Disponibilidade). O pilar da Confidencialidade busca a proteção contra acessos não autorizados, de modo que apenas indivíduos ou sistemas autorizados acessem os dados. A Integridade refere-se à capacidade de garantir que a informação permaneça consistente e completa ao longo de seu ciclo de vida, pois qualquer alteração indevida pode comprometer sua confiabilidade. Ademais, a Disponibilidade assegura que a informação esteja acessível quando necessária, o que exige a manutenção contínua de software e hardware para evitar interrupções (FERNANDES et al., 2025).

Assim, no contexto da proteção de dados sensíveis em bancos de dados, constam ataques que incidem sobre a tríade CIA. Primeiramente, o *phishing* consiste em comunicações fraudulentas que induzem a entrega de informações confidenciais ou à instalação de *malware*. Em seguida, a *SQL Injection* corresponde à inserção de consultas ou

comandos maliciosos em campos editáveis para atingir o banco de dados. Por fim, o *ransomware* decorre do acesso a sites sem segurança ou da abertura de anexos de e-mails não confiáveis, o que leva à tomada de controle do sistema e à exigência de resgate. Portanto, esses vetores comprometem a confidencialidade, a integridade e a disponibilidade, pilares que restringem o ingresso não autorizado, preservam a correção dos dados e asseguram a continuidade operacional da informação (DOMINGUES; AUGUSTINI; OLIVEIRA, 2024).

2.3. Armazenamento Descentralizado

O armazenamento descentralizado consiste na distribuição e manutenção de dados em múltiplos nós de uma rede, sem dependência de um servidor central, com o objetivo de eliminar ponto único de falha e reforçar a disponibilidade e a resistência à censura. Para isso, os sistemas organizam-se como redes de armazenamento descentralizado (DSNs) que reúnem recursos de diferentes provedores, operam por meio de clientes e "*miners*" (armazenamento e recuperação) e executam funções de inserir, gerir e obter dados (LI et al., 2024).

Nesse contexto, o modelo *peer-to-peer* descreve uma arquitetura de sistemas distribuídos na qual os processos assumem papéis semelhantes e atuam como pares, sem distinção rígida entre cliente e servidor, utilizando os recursos de numerosos computadores participantes para executar a atividade, com distribuição de cargas de armazenamento, processamento e comunicação, além de replicação de objetos para aumentar a resiliência e a disponibilidade (COULOURIS; DOLLIMORE; KINDBERG, 2013). Dessa forma, esse modelo pode ser encontrado em sistemas distribuídos.

Em contraste com o armazenamento centralizado, no qual os dados permanecem em servidores remotos administrados por um terceiro que concentra a governança e limita o controle do usuário, as redes de armazenamento descentralizado distribuem os arquivos entre múltiplos provedores independentes e podem utilizar arquiteturas P2P, adotam mecanismos de verificação pública e empregam incentivos econômicos que deslocam a formação de preços para um mercado aberto. Além disso, enquanto ambientes centralizados reportam métricas sob responsabilidade dos próprios operadores, os sistemas descentralizados utilizam provas e auditorias realizadas pela rede para assegurar disponibilidade e integridade dos dados (KHALID et al., 2023).

O *InterPlanetary File System* (IPFS) é um sistema *peer-to-peer* (P2P) de arquivos com endereçamento por conteúdo e distribuição entre pares, proposto para tornar dados verificáveis, cuja integridade pode ser confirmada, versionáveis, que permitem o controle de diferentes versões de um mesmo conteúdo e resilientes a falhas, característica que garante a disponibilidade das informações mesmo diante de falhas em parte da rede (Benet, 2014).

Como vantagem central, o IPFS permite recuperar um objeto de qualquer nó que o possua, por meio de seu identificador de conteúdo, o que reduz a dependência de um servidor único e melhora a disponibilidade. Em contrapartida, no armazenamento tradicional (*on-premises* ou servidor único) e em nuvens centralizadas, os dados são endereçados por localização (URL, caminho, provedor), o que resulta na existência de pontos únicos de falha (IPFS DOCS, 2024).

Trautwein et al. (2022) descrevem que o IPFS adota um modelo de endereçamento baseado em conteúdo, no qual cada arquivo recebe um Identificador de Conteúdo (CID) calculado a partir de um *hash* criptográfico. Para esse fim, o sistema utiliza, por padrão, o algoritmo SHA-256, além de empregar o formato *multihash*, o qual permite a adoção de diferentes funções criptográficas. Dessa forma, o CID torna-se independente da localização física do conteúdo, que assegura propriedades de imutabilidade, verificabilidade e extensibilidade.

Desse modo, o *Mutable File System* (MFS) consiste em um recurso nativo do IPFS que permite a manipulação de arquivos e diretórios por nomes, de forma semelhante a um

sistema de arquivos tradicional, enquanto o próprio IPFS gerencia a atualização dos *hashes* e elos associados, sendo o acesso realizado por meio dos comandos *files* na CLI e na API, cuja localização de quem "fornece" um CID é descoberta via roteamento de conteúdo (ex.: DHT). Ademais, o IPFS *Cluster* constitui uma aplicação distribuída responsável pela orquestração e replicação de dados entre múltiplos *daemons* do IPFS, que possibilita a escalabilidade da operação e assegura uma maior disponibilidade do conteúdo. Por fim, ressalta-se que o IPFS não realiza a cifragem dos dados por padrão, a orientação oficial é criptografar antes de adicionar (*client-side* E2EE), de modo que apenas quem possui a chave consiga ler o conteúdo, mesmo que o CID seja público (IPFS DOCS, 2024, 2025).

2.4. Trabalhos relacionados

Nunes et al. (2021) apontam a inadequação do modelo centralizado de dados do SUS para informações sensíveis, em razão da existência de ponto único de falha, da suscetibilidade a vazamentos e alterações indevidas de dados. Propõe-se arquitetura descentralizada em que os documentos clínicos permanecem no IPFS e o *hash*/CID de cada arquivo se registra em *blockchain*, o que garante imutabilidade e rastreabilidade. Para preservar a privacidade, indica-se criptografia assimétrica (OpenPGP) antes do envio ao IPFS. Em consequência, a solução favorece a transparência, integridade, integração entre sistemas e segurança em conformidade com a LGPD.

Linhares (2024) apresenta protótipo web com arquitetura híbrida (*Blockchain* + IPFS + SQLite). No fluxo proposto, o PDF do exame, arquivo digital que contém o laudo laboratorial do paciente, é enviado ao IPFS por meio de serviço de *pinning* (Pinata); o CID retorna e se insere em contrato inteligente, que realiza o controle de acesso com carteiras (ex.: *MetaMask*) para perfis autorizados (pacientes, médicos, administradores). Ademais, o trabalho reporta testes de desempenho com arquivos de tamanhos variados e discute o tempo de resposta do sistema.

Diante desse quadro, os dois estudos convergem em três elementos técnicos aplicáveis ao atendimento psicológico online: (i) uso do endereçamento por conteúdo (CID) no IPFS para armazenar prontuários, anexos clínicos e registros de sessão, com alta disponibilidade; (ii) criptografia cliente-a-cliente (E2EE) antes do envio, a qual assegura sigilo profissional e aderência à LGPD; e (iii) controle de acesso por perfis (psicólogo, paciente, supervisão/gestão), com registro de auditoria. Embora ambos os trabalhos façam uso de blockchain como mecanismo adicional de auditoria e verificação, o sistema proposto implementa um controle de auditoria próprio, suficiente para atender aos requisitos do contexto analisado. Desse modo, consolida-se a escolha do IPFS como base de armazenamento seguro para o sistema proposto, o que eleva a disponibilidade, a verificabilidade e, sobretudo, a proteção e a confidencialidade dos dados psicológicos.

3. Materiais e Métodos

3.1. Materiais

Para a implementação, foram empregadas algumas ferramentas de *software*, incluindo:

- *Node.js* com *NestJS*, o *backend* da aplicação foi desenvolvido com o *NestJS*, um *framework* progressivo de *Node.js* projetado para a construção de aplicações de servidor eficientes, confiáveis e escaláveis (*NestJS* DOCS, 2025).
- *Supabase*, foi adotada na camada de persistência de dados, uma plataforma de código aberto que provê um conjunto de ferramentas de *backend*, como um banco de dados PostgreSQL e APIs instantâneas (*Supabase* DOCS, 2025).

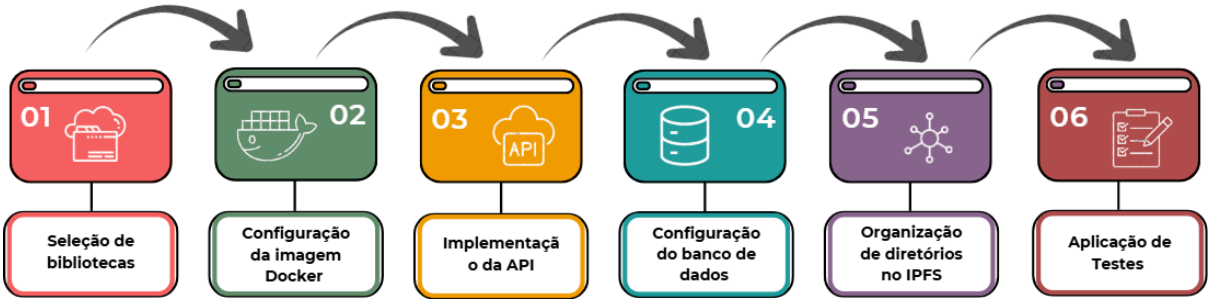
- *Docker*, foi empregado para a conteinerização da aplicação, uma abordagem que permite empacotar e executar o software de forma isolada e consistente entre diferentes ambientes (*Docker* DOCS, 2025).
- *IPFS* (*Kubo*), foi utilizado como a rede de arquivos distribuída por meio do *Kubo*, sua implementação oficial de referência, responsável por prover o *daemon*, a interface de linha de comando e as APIs necessárias para o armazenamento, a recuperação e a distribuição de conteúdo (IPFS DOCS, 2025).
- RSA, um algoritmo de criptografia assimétrica para o gerenciamento de chaves implementado para assegurar a confidencialidade e a integridade dos dados (RSA, 2025).
- JWT (*JSON Web Tokens*), responsável pelo controle de acesso e a autorização de usuários por meio de um padrão de *token* autossuficiente para a troca segura de informações (JWT.IO, 2025).
- *Swagger* (*OpenAPI Specification*), foi utilizado para elaborar a documentação da API em uma interface interativa para a visualização e o teste dos *endpoints*, com base na *OpenAPI Specification* (*Swagger* DOCS, 2025).

Esses materiais mostraram-se adequados para o processo que resultou no desenvolvimento apresentado neste artigo. O processo metodológico para alcançar esse resultado é descrito a seguir.

3.2. Métodos

O desenvolvimento deste projeto foi estruturado em uma metodologia composta em etapas, conforme ilustrado na Figura 1.

Figura 1. Fluxo da Metodologia



A Figura 1 apresenta a metodologia do projeto. A etapa 1 consistiu na seleção das bibliotecas necessárias para o desenvolvimento do microserviço de Armazenamento Descentralizado, com base em critérios de aderência às tecnologias adotadas. Na sequência, a fase 2 envolveu a configuração da imagem *Docker* do serviço Kubo IPFS, com a definição de variáveis de ambiente, montagem de volumes, exposição de portas e a implementação de políticas de rede, com o intuito de garantir a operação segura e eficiente do sistema descentralizado de arquivos.

Em seguida, na etapa 3, a API foi desenvolvida com o *framework NestJS*, sendo responsável por orquestrar a comunicação entre os microserviços e permitir a interação com o IPFS para o armazenamento dos dados. Posteriormente, a fase 4 compreendeu a configuração do banco de dados no *Supabase*, onde foram armazenados os logs das operações, com informações sobre o tipo de requisição, o identificador do usuário, o CID dos arquivos e os caminhos dos arquivos armazenados.

A etapa 5 consistiu na organização da estrutura de diretórios no IPFS, com a criação de subdiretórios por paciente e a categorização dos arquivos. A segurança e a integridade dos

dados sensíveis foram garantidas por meio de autenticação via JWT e restrições de permissões de acesso, o que proporcionou a conformidade com as exigências legais e éticas.

Por fim, a etapa 6 compreendeu a aplicação de testes de desempenho e disponibilidade do sistema, por meio de scripts destinados à avaliação das métricas de latência, *throughput* e estabilidade do *cluster* IPFS, com o objetivo de validar a confiabilidade e a eficiência da solução proposta.

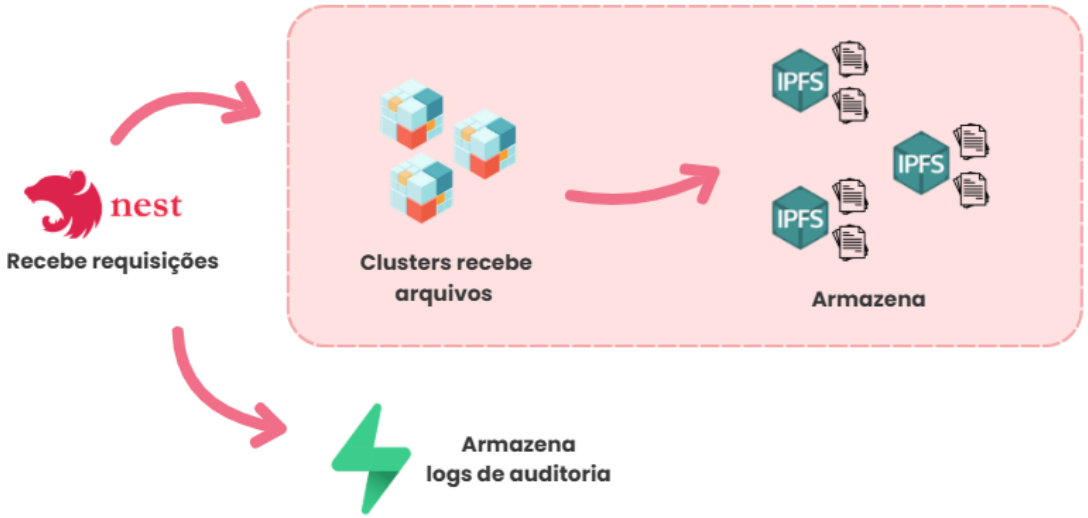
4. Resultados

Esta seção apresenta os resultados obtidos com o desenvolvimento da pesquisa. A partir das decisões metodológicas previamente estabelecidas, foi implementada uma arquitetura de solução composta por tecnologias específicas para cada domínio funcional do sistema.

No que tange ao armazenamento descentralizado, a escolha recaiu sobre o Kubo IPFS, decisão fundamentada em sua capacidade de utilizar *clusters* para o gerenciamento de nós. A comunicação com essa infraestrutura descentralizada foi viabilizada pela biblioteca ipfs-http-client. Para a segurança dos dados, empregou-se o algoritmo RSA de criptografia assimétrica, enquanto os processos de autenticação e controle de permissões foram estruturados com base no padrão *JSON Web Token* (JWT).

Adicionalmente, em virtude da ausência de mecanismos de auditoria nativos no IPFS, integrou-se a plataforma *Supabase*. Essa conexão foi estabelecida mediante um serviço intermediário, desenvolvido em *NestJS*, ao qual coube a função de gerenciar e registrar as informações. Para tal finalidade, a biblioteca *Supabase JS Client* foi implementada para garantir a persistência dos dados. Na sequência, esta seção apresenta a arquitetura proposta e descreve seus componentes e fluxos de integração.

Figura 2. Arquitetura do projeto

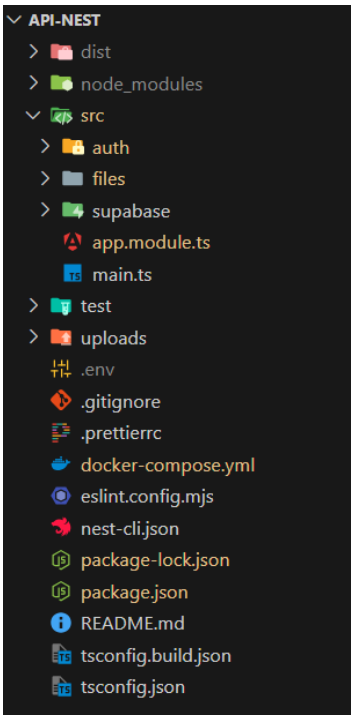


A Figura 2 ilustra a arquitetura do projeto desenvolvido. Inicialmente, o *backend* da aplicação feito em *NestJS* é responsável por receber as requisições e encaminhá-las ao *cluster* de armazenamento distribuído. Em seguida, os pares que compõem o *cluster* remetem os objetos aos nós da rede IPFS, nos quais os conteúdos permanecem endereçados por identificadores de conteúdo (CID). Em paralelo, o componente de persistência e auditoria (*Supabase*) registra os logs e preserva as trilhas de acesso e operação. Dessa forma, a arquitetura integra orquestração de requisições, armazenamento distribuído e governança por meio de registros. Cada componente é detalhado nas seções seguintes.

4.1. API

Primeiramente, foi necessária a implementação da API que recebe as requisições. O *framework NestJS* foi adotado para este fim, que permite a disponibilização do serviço na rede e, consequentemente, a realização de testes com o armazenamento descentralizado. A estrutura da API, conforme apresentada na Figura 3, reflete essa implementação.

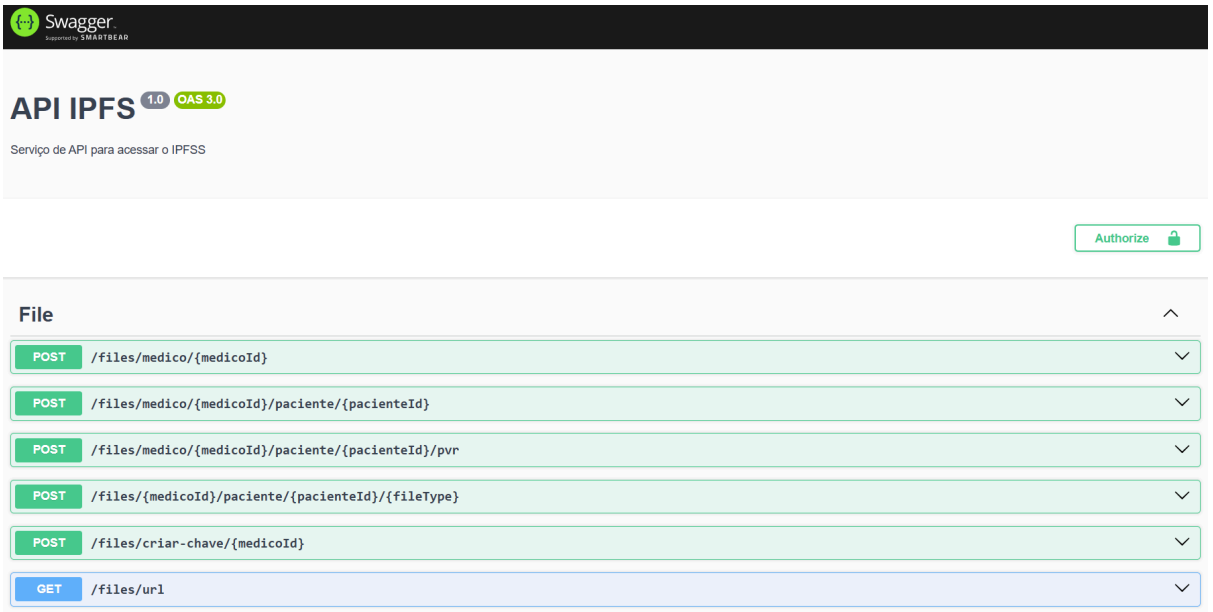
Figura 3. Estrutura de pastas da API



A Figura 3 apresenta a estrutura da API, cuja a estrutura do projeto foi organizada de forma modular para promover a separação de responsabilidades e facilitar a manutenibilidade. Nesse contexto, cada diretório no módulo possui uma finalidade específica. O módulo `auth` concentra as funcionalidades de autenticação e controle de acesso. Por sua vez, o módulo `files` é responsável pelo tratamento dos *uploads* e pela comunicação direta com a infraestrutura descentralizada IPFS. Em suma, o módulo `supabase` implementa os serviços que realizam a persistência e a consulta de dados na plataforma *Supabase*.

Com base nesta estrutura, a aplicação possui um conjunto de *endpoints* principais. A ferramenta *Swagger* foi integrada para documentar e facilitar o teste interativo dessas rotas, o que permite apresentar de forma clara todos os pontos de acesso ao serviço IPFS, conforme ilustra a Figura 4.

Figura 4. *Swagger* da aplicação

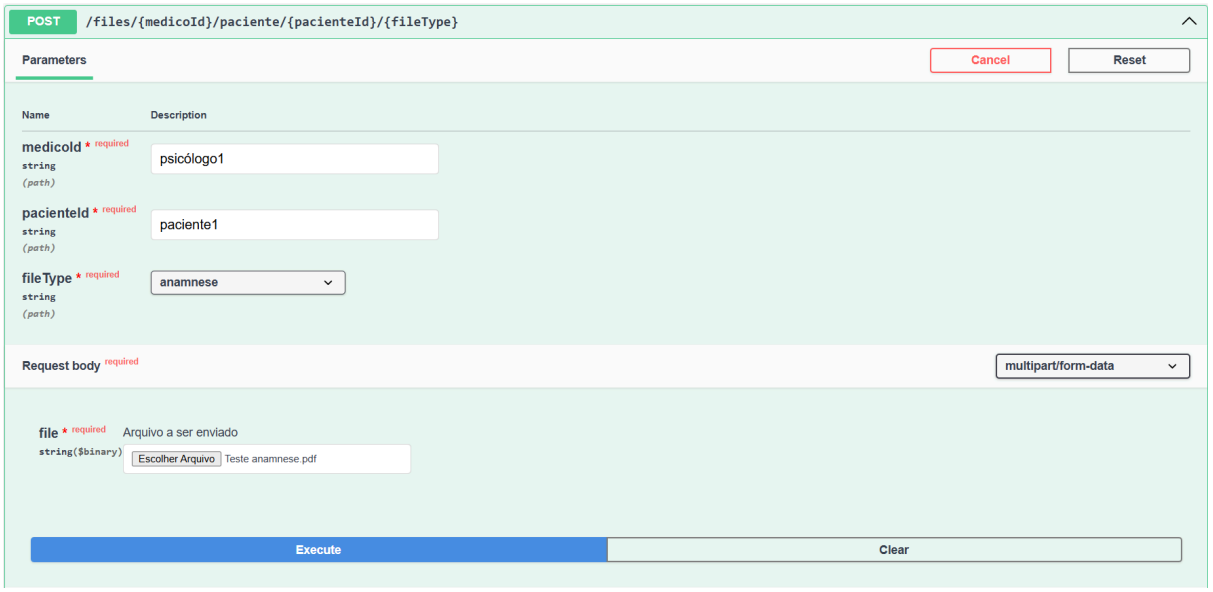


A Figura 4 ilustra a interface de *endpoints* da API, documentada com o *Swagger* com o conjunto de operações para o gerenciamento de diretórios e arquivos na infraestrutura descentralizada. Inicialmente, a arquitetura de dados é estabelecida por meio de rotas específicas de criação de diretórios. No primeiro *endpoint* `/files/medico/{medicoId}`, define-se a criação do repositório base de armazenamento para cada profissional específico no MFS (*Mutable File System*) do IPFS. De modo complementar, no segundo *endpoint* `/files/medico/{medicoId}/paciente/{pacienteId}`, cria um subdiretório para um paciente, aninhado dentro da estrutura do médico correspondente. Adicionalmente, o terceiro *endpoint* `/files/medico/{medicoId}/paciente/{pacienteId}/pvr`, serve como um inicializador que, de uma só vez, cria toda a estrutura de subdiretórios padrão para um paciente (*chat*, relatórios, emoção, anamnese), o que prepara o ambiente para futuras submissões de dados.

No fluxo de manipulação de arquivos, há duas operações centrais. A submissão de dados ocorre no quarto *endpoint* `/files/{medicoId}/paciente/{pacienteId}/{fileType}`, que recebe um arquivo e o armazena no subdiretório apropriado, conforme o tipo especificado na rota (chat, relatórios, emoção, anamnese). Após a inserção do arquivo no IPFS, a aplicação registra um *log* da transação na base de dados *Supabase*, com informações como o identificador do médico, o paciente, o caminho do arquivo e seu CID (*Content Identifier*), o que materializa a camada de auditoria do sistema. Por sua vez, a recuperação de dados é viabilizada pelo sexto *endpoint* `/files/url` que, a partir de um caminho no MFS, retorna uma URL de acesso público ao arquivo por meio de um *gateway* IPFS.

Em complemento, o quinto *endpoint* `/files/criar-chave/{medicoId}`, executa uma função de segurança auxiliar, cuja finalidade é gerar um par de chaves criptográficas exclusivo para cada médico, que fica associado à sua identidade no sistema para fins de integridade e autenticidade. E como exemplo de execução, o quarto endpoint citado foi utilizado como referência prática, conforme ilustrado na Figura 5.

Figura 5. Exemplo de execução



A Figura 5 apresenta um exemplo de execução do *endpoint* de submissão de documentos na interface *Swagger*. No cenário ilustrado, o identificador do profissional é "psicólogo 1", o do paciente é "paciente 1", o tipo do arquivo é "anamnese" e o nome do arquivo é "Teste anamnese.pdf". Após a requisição, o sistema cria no *Supabase* um log com esses dados e define, no MFS do IPFS, o caminho para armazenamento e posterior recuperação em `files/medicos/psicologo1/pacientes/paciente1/anamnese`. Como resposta, o *endpoint* retorna uma mensagem de confirmação da operação, acompanhada do CID gerado para o arquivo armazenado.

4.2. Serviço IPFS

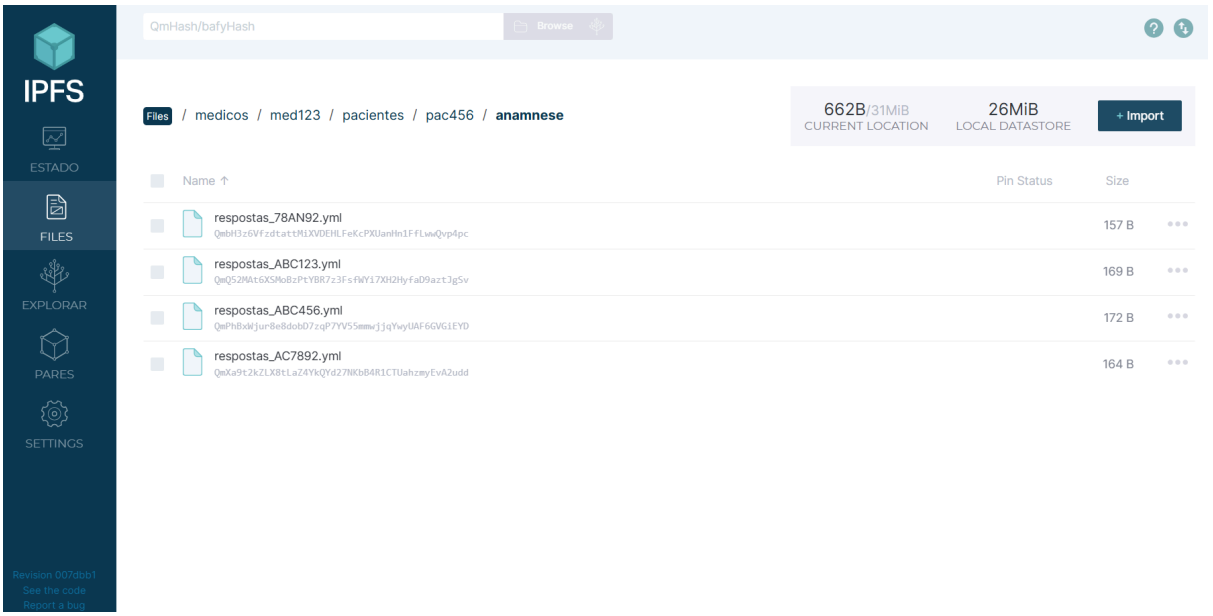
Inicialmente, a topologia define três nós IPFS e três pares de coordenação do IPFS *Cluster*. Um dos pares atua como nó de ingresso (*bootstrap*) e os demais referenciam esse endereço, de modo que todos passam a integrar a mesma rede lógica. Para evitar conflitos entre dependências, estabelece-se a ordem de inicialização dos serviços.

Em seguida, a camada de coordenação aplica políticas de replicação e de fixação de conteúdo. Cada par mantém visão consistente das fixações (*pins*) e orquestra a distribuição entre os nós IPFS. Quando um objeto recebe fixação, o cluster define o fator de replicação e aloca os nós responsáveis conforme a estratégia configurada. Paralelamente, cada par comunica-se com o *daemon* IPFS local. Em caso de falha de um nó, as réplicas permanecem acessíveis nos demais, preservando disponibilidade e continuidade do serviço.

No plano de dados, os objetos ingressam pela aplicação e seguem ao cluster, que propaga a instrução de fixação aos nós selecionados. Os *daemons* IPFS (Kubo) fragmentam os arquivos em blocos, calculam os CIDs e anunciam provedores na DHT (*Distributed Hash Table*). Além disso, a troca de blocos entre pares por meio do protocolo de distribuição reduz a dependência de um único provedor e favorece a recuperação a partir de múltiplos nós do próprio cluster.

Ademais, o *InterPlanetary File System* compõe o núcleo do armazenamento distribuído. Os objetos são endereçados por identificadores de conteúdo, o que viabiliza rastreabilidade e imutabilidade. Assim, o acesso aos registros independe de um ponto central de controle. Essa estrutura de diretórios do IPFS, conforme apresenta a Figura 6, resulta em um armazenamento de dados mais ordenado.

Figura 6. Ilustração da hierarquia de diretórios no IPFS



A Figura 6 apresenta a estrutura hierárquica no IPFS, uma organização de dados semelhante a sistemas de arquivos tradicionais. Tal abordagem é viabilizada pelo MFS, uma camada de abstração cujo principal objetivo é fornecer uma interface mutável sobre a base de dados de conteúdo imutável da rede. O MFS resolve o desafio inerente ao IPFS, no qual qualquer modificação em um arquivo gera um novo endereço criptográfico (CID), ao permitir a utilização de caminhos estáveis e legíveis.

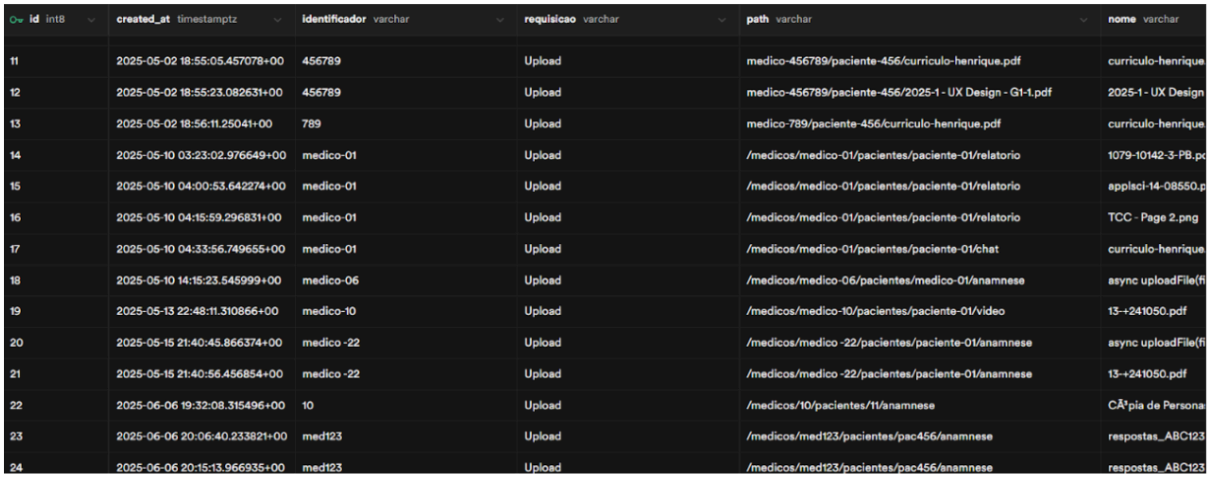
Nesta implementação, a hierarquia tem início em um diretório raiz `/medicos`, seguido pelo identificador de cada profissional (`/med123`), que por sua vez contém as pastas de cada paciente (`/pac456`), finalmente organizadas em subdiretórios temáticos como anamnese. Essa organização, implementada de forma programática pela API, combina a integridade verificável do armazenamento por conteúdo com a usabilidade de um sistema convencional, o que resulta em um método de armazenamento lógico e escalável para as constantes atualizações de dados que a aplicação demanda.

4.3. Controle de Auditoria

De forma complementar, o controle de auditoria registra eventos de acesso e operação em logs estruturados, centralizando evidências para verificação posterior e suporte à conformidade. A utilização do *Supabase* como repositório possibilita organizar registros verificáveis, o que fortalece os mecanismos de rastreabilidade e controle.

Em termos de organização, a tabela de *logs* dispõe de campos destinados ao registro de cada operação. Em primeiro lugar, há um identificador único do registro e o momento exato da transação. Em seguida, campos adicionais armazenam informações contextuais: identificação do profissional, identificação do paciente associado e descrição da operação realizada. De modo complementar, outros campos caracterizam o arquivo envolvido: nome original, local de armazenamento e identificador de conteúdo. Com base nessa tabela de *logs*, a base de dados da camada de auditoria passou a receber informações provenientes da API desenvolvida, conforme ilustrado na Figura 7.

Figura 7. Tabela de *Logs* do *Supabase*



id int8	created_at timestamptz	identificador varchar	requisicao varchar	path varchar	nome varchar
11	2025-05-02 18:55:05.457078+00	456789	Upload	medico-456789/paciente-456/curriculo-henrique.pdf	curriculo-henrique
12	2025-05-02 18:55:23.082631+00	456789	Upload	medico-456789/paciente-456/2025-1 - UX Design - G1-1.pdf	2025-1 - UX Design
13	2025-05-02 18:56:11.25041+00	789	Upload	medico-789/paciente-456/curriculo-henrique.pdf	curriculo-henrique
14	2025-05-10 03:23:02.976649+00	medico-01	Upload	/medicos/medico-01/pacientes/paciente-01/relatorio	1079-10142-3-PB.pc
15	2025-05-10 04:00:53.642274+00	medico-01	Upload	/medicos/medico-01/pacientes/paciente-01/relatorio	applsci-14-08550.p
16	2025-05-10 04:15:59.296831+00	medico-01	Upload	/medicos/medico-01/pacientes/paciente-01/relatorio	TCC - Page 2.png
17	2025-05-10 04:33:56.749655+00	medico-01	Upload	/medicos/medico-01/pacientes/paciente-01/chat	curriculo-henrique
18	2025-05-10 14:15:23.545999+00	medico-06	Upload	/medicos/medico-06/pacientes/medico-01/anamnese	async uploadFile(fi
19	2025-05-13 22:48:11.310866+00	medico-10	Upload	/medicos/medico-10/pacientes/paciente-01/video	13-+241050.pdf
20	2025-05-15 21:40:45.866374+00	medico -22	Upload	/medicos/medico -22/pacientes/paciente-01/anamnese	async uploadFile(fi
21	2025-05-15 21:40:56.456854+00	medico -22	Upload	/medicos/medico -22/pacientes/paciente-01/anamnese	13-+241050.pdf
22	2025-06-06 19:32:08.315496+00	10	Upload	/medicos/10/pacientes/11/anamnese	CÃ³pia de Persona
23	2025-06-06 20:06:40.233821+00	med123	Upload	/medicos/med123/pacientes/pac456/anamnese	respostas_ABC123
24	2025-06-06 20:15:13.966935+00	med123	Upload	/medicos/med123/pacientes/pac456/anamnese	respostas_ABC123

A Figura 7 ilustra um recorte da tabela de *logs* armazenada no *Supabase*, utilizada como camada de auditoria da aplicação. Essa tabela registra cada operação realizada no sistema e permite a rastreabilidade das ações executadas pelos usuários. Cada registro corresponde a um evento, identificado pelo valor da coluna `requisicao`, enquanto a coluna `identificador` armazena o identificador do profissional responsável pela operação. O nome lógico do arquivo submetido é registrado na coluna `name`, e o caminho hierárquico de armazenamento no IPFS é descrito na coluna `path`, o que reflete a organização dos dados por profissional e paciente. A coluna `created_at` armazena o instante de criação do registro, o que possibilita a ordenação cronológica dos eventos e a reconstrução do histórico de submissões. Em conjunto, essas colunas compõem um modelo de dados voltado à auditoria, assegurando integridade, rastreabilidade e verificabilidade das operações realizadas no sistema.

4.4. Aplicação de Testes

Por fim, a aplicação de testes abrangeu o conjunto de *scripts* localizado no diretório *test/benchmark*, responsável por validar a disponibilidade, a latência e o *throughput* (taxa de transferência) do IPFS. Esses testes verificam o comportamento da API durante operações reais de armazenamento e recuperação, asseguram a consistência das respostas e confirmam a integração correta entre API, IPFS e camada de auditoria. Dessa forma, contribuem para os princípios da tríade CIA (Confidencialidade, Integridade e Disponibilidade), especialmente no que se refere à integridade dos dados e à disponibilidade dos serviços. A seguir, apresentam-se os resultados obtidos nessas avaliações, com início na tabela de latência.

Tabela 1. Latência de Transferência de Arquivos

Tamanho	Nó	Operação	Min (ms)	Avg (ms)	Max (ms)
10KB	ipfs0	Upload	13.00	15.60	19.00
10KB	ipfs0	Download	1.00	1.20	2.00
10KB	ipfs1	Upload	13.00	14.80	18.00
10KB	ipfs1	Download	1.00	1.40	2.00
10KB	ipfs2	Upload	14.00	15.40	16.00
10KB	ipfs2	Download	1.00	1.80	3.00

1MB	ipfs0	Upload	19.00	34.40	72.00
1MB	ipfs0	Download	5.00	6.40	8.00
1MB	ipfs1	Upload	21.00	60.40	84.00
1MB	ipfs1	Download	4.00	13.00	33.00
1MB	ipfs2	Upload	19.00	37.80	66.00
1MB	ipfs2	Download	4.00	4.40	5.00
10MB	ipfs0	Upload	90.00	100.60	106.00
10MB	ipfs0	Download	35.00	36.20	37.00
10MB	ipfs1	Upload	100.00	103.40	105.00
10MB	ipfs1	Download	28.00	34.40	42.00
10MB	ipfs2	Upload	76.00	98.80	118.00
10MB	ipfs2	Download	34.00	37.60	40.00

A Tabela 1 ilustra os resultados de latência para operações de *upload* e *download* de arquivos no *cluster* IPFS, com testes que abrangem diferentes tamanhos de arquivos (10KB a 10MB) executados em três nós distintos (ipfs0, ipfs1 e ipfs2). Os valores, expressos em milissegundos (ms), incluem as métricas mínima, média e máxima obtidas em 5 iterações.

Os dados demonstram que a latência de *download* é significativamente menor que a de *upload* em todos os cenários. Para arquivos de 1MB, a latência média de *upload* variou entre 34.40ms e 60.40ms, ao passo que o *download* apresentou latências entre 4.40ms e 13.00ms. Este comportamento é esperado em sistemas IPFS, visto que o processo de *upload* envolve etapas adicionais como *chunking*, *hashing* e propagação pela rede DHT. A latência aumenta de forma não-linear com o crescimento do tamanho do arquivo, o que evidencia maior eficiência proporcional no processamento de arquivos maiores. A variação entre os nós manteve-se dentro de uma faixa aceitável (diferença máxima de 26ms), o que indica que o cluster está balanceado.

Além da análise de operações de transferência de arquivos, torna-se essencial avaliar também as operações realizadas diretamente no sistema de arquivos mutável do IPFS, conforme apresentado na próxima seção.

Tabela 2. Latência de Operações MFS

Operação	**Nó**	**Min (ms)**	**Avg (ms)**	**Max (ms)**
Write	ipfs0	12.00	13.60	18.00
Write	ipfs1	12.00	12.60	13.00
Write	ipfs2	11.00	12.80	15.00
Read	ipfs0	1.00	1.60	3.00
Read	ipfs1	1.00	1.60	3.00
Read	ipfs2	1.00	1.20	2.00
Stat	ipfs0	1.00	1.60	3.00

Stat	ipfs1	1.00	1.40	3.00
Stat	ipfs2	1.00	1.20	2.00

A Tabela 2 apresenta as latências das operações do MFS (*Mutable File System*), camada de abstração do IPFS que permite manipulação de arquivos de forma similar a sistemas de arquivos tradicionais. As operações testadas foram *Write* (escrita), *Read* (leitura) e *Stat* (obtenção de metadados), todas executadas nos três nós do *cluster*.

Quanto às operações de leitura, observa-se que *Read* e *Stat* apresentaram latências muito baixas (médias entre 1.20ms e 1.60ms), o que indica que o IPFS mantém metadados e blocos de dados em *cache* local, sem necessidade de comunicação extensa pela rede. Em contraste, a operação *Write* apresentou latências superiores (médias entre 12.60ms e 13.60ms), visto que exige atualização da estrutura de dados imutável, geração de novos CIDs e propagação para os *peers*. Apesar disso, a consistência entre os nós é notável, com diferença máxima de apenas 1ms entre as médias, o que reforça a estabilidade do *cluster*.

Diante desses resultados, a análise de latência do MFS mostra o comportamento das operações pontuais de arquivo, mas também exige a verificação do desempenho do sistema sob carga contínua, ao medir sua capacidade de processar volumes maiores de dados. Nesse sentido, a próxima tabela apresenta as métricas de *throughput*, que complementam a análise de latência.

Tabela 3. *Throughput* (Vazão de Dados)

Nó	Operação	Vazão (MB/s)	Total	Taxa Sucesso
ipfs0	Upload	3.36	16.11 MB	100.0%
ipfs0	Download	309.76	16.11 MB	100.0%
ipfs1	Upload	3.38	16.11 MB	100.0%
ipfs1	Download	298.29	16.11 MB	100.0%
ipfs2	Upload	3.43	16.11 MB	100.0%
ipfs2	Download	201.34	16.11 MB	100.0%

A Tabela 3 quantifica o *throughput* (vazão) do *cluster* IPFS, métrica que expressa a taxa de transferência de dados em megabytes por segundo (MB/s) durante operações contínuas. Ao contrário da latência, que avalia o tempo de resposta de ações isoladas, essa métrica indica a capacidade agregada de processamento do sistema.

No que se refere ao desempenho, as taxas de *download* variaram entre 201,34 MB/s e 309,76 MB/s. Em contraste, as taxas de *upload* ficaram entre 3,36 MB/s e 3,43 MB/s, diferença explicada pelas etapas adicionais envolvidas na ingestão de dados, como *chunking*, geração de hashes SHA-256 e construção de *Merkle DAGs*. Apesar dessa assimetria, os valores permanecem adequados para documentos médicos usuais, que normalmente possuem tamanho reduzido. Um arquivo de 10MB, por exemplo, pode ser processado em cerca de 3 segundos. Em complemento, todos os testes registraram 100% de sucesso, sem qualquer falha durante a execução dos *benchmarks*.

Com base nesses dados, o *cluster* apresenta desempenho adequado para operações de escrita e leitura, com superioridade clara no processo de recuperação de arquivos. A Tabela 4 amplia essa análise ao apresentar informações referentes à disponibilidade dos nós e ao comportamento da replicação no ambiente distribuído, que são importantes para verificar a

capacidade do sistema de manter o acesso aos dados e garantir redundância mesmo diante de falhas parciais na rede.

Tabela 4. Disponibilidade e Replicação do *Cluster*

Nó	Métrica	Resultado	Detalhe
ipfs0	Status	Online	0.34.1
ipfs1	Status	Online	0.34.1
ipfs2	Status	Online	0.34.1
ipfs0	Recuperação de Dados	Sucesso	5.00 ms
ipfs1	Recuperação de Dados	Sucesso	9.00 ms
ipfs2	Recuperação de Dados	Sucesso	4.00 ms
Cluster	Disponibilidade Total	100.00%	-
Cluster	Fator de Replicação	1.00x	-

A Tabela 4 apresenta os resultados referentes à disponibilidade dos nós e ao processo de replicação do conteúdo no *cluster* IPFS. Todos os nós permaneceram *online* durante os testes, cada um executando a versão 0.34.1 do *daemon* Kubo, o que assegura uniformidade no ambiente. As operações de recuperação de dados registraram latências entre 4ms e 9ms, o que indica acesso imediato aos arquivos e resposta adequada da rede às solicitações.

No mesmo sentido, a disponibilidade total atingiu 100%, o que mostra ausência de falhas ao longo dos testes. O fator de replicação de 1,00x indica a manutenção de ao menos uma cópia válida dos dados, em conformidade com a política definida para o cenário de avaliação. Esses resultados demonstram que a estrutura distribuída preserva a continuidade do serviço e reduz o impacto de eventuais falhas isoladas.

Assim, os resultados de disponibilidade confirmam o funcionamento estável do *cluster* e sustentam a viabilidade de sua utilização em sistemas que tratam dados sensíveis. Com isso, conclui-se a etapa de avaliação dos testes, que evidenciou integração adequada entre a API, o IPFS e a camada de auditoria.

5. Considerações Finais

Este trabalho teve como objetivo desenvolver um sistema de armazenamento seguro para dados de teleatendimento psicológico, estruturado sobre IPFS e organizado com MFS, com serviços de aplicação em *NestJS*, autenticação por JWT, criptografia assimétrica para proteção de conteúdo e trilhas de auditoria persistidas. A arquitetura contemplou a orquestração de nós, o isolamento por contêineres, a definição de diretórios por profissional e paciente, a vinculação de arquivos a CIDs para identificação imutável e a exposição de *endpoints* documentados. Desse modo, o projeto oferece indícios de redução do ponto único de falha, aumento de disponibilidade via replicação e melhoria na rastreabilidade do ciclo de vida dos dados.

Os resultados indicam integração entre a camada de rede descentralizada e a aplicação, com controle de acesso, registro de eventos relevantes e organização de evidências para auditoria. A solução demonstrou viabilidade técnica para armazenar, recuperar e auditar artefatos sensíveis do atendimento, com preservação dos princípios de confidencialidade, integridade e disponibilidade, além de conformidade com práticas de governança de dados compatíveis com ambientes clínicos.

Por outro lado, verificam-se limitações que orientam aperfeiçoamentos. IPFS não cifra conteúdo por padrão. Assim, a segurança depende de políticas de criptografia de ponta a ponta e de gestão segregada de chaves. A dependência de *gateways* públicos pode introduzir latências e janelas de indisponibilidade. Para próximos trabalhos, recomenda-se múltiplos pontos de acesso e estratégias de *pin*. Além disso, a avaliação concentrou-se em validações técnicas.

Diante desse quadro, propõe-se como trabalhos futuros testes controlados de estresse e de disponibilidade, evoluir para uma estrutura com *cluster* ampliado e utilizar nuvens públicas para a distribuição dos nós. Em síntese, o trabalho consolida fundamentos técnicos e operacionais para o armazenamento descentralizado de dados de telepsicologia, abre caminho para a validação em campo e estrutura a solução para uso em escala institucional.

6. Referências

AZBEG, Kebira; OUCHETTO, Ouail; ANDALOUSSI, Said Jai. BlockMedCare: A healthcare system based on IoT, Blockchain and IPFS for data management security. **Egyptian informatics journal**, v. 23, n. 2, p. 329-343, 2022. Disponível em: https://www.sciencedirect.com/science/article/pii/S1110866522000160.

BENET, Juan. **IPFS – Content Addressed, Versioned, P2P File System.** arXiv preprint arXiv:1407.3561, 2014. Disponível em: https://arxiv.org/abs/1407.3561.

BRASIL. Lei nº 13.709, de 14 de agosto de 2018. Dispõe sobre a proteção de dados pessoais e altera a Lei nº 12.965, de 23 de abril de 2014 (Marco Civil da Internet). Brasília, DF, 2018. Disponível em: https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/l13709.htm.

CONSELHO FEDERAL DE PSICOLOGIA. **Resolução nº 11, de 11 de maio de 2018**. Regulamenta a prestação de serviços psicológicos realizados por meios de tecnologias da informação e da comunicação e revoga a Resolução CFP nº 11/2012. Brasília, 2018.

CONSELHO FEDERAL DE PSICOLOGIA. **Resolução nº 4, de 26 de março de 2020**. Dispõe sobre regulamentação de serviços psicológicos prestados por meio de Tecnologia da Informação e da Comunicação durante a pandemia do COVID-19. Brasília, 2020. Disponível em:https://atosoficiais.com.br/cfp/resolucao-do-exercicio-profissional-n-4-2020-dispoe-sobre-regulamentacao-de-servicos-psicologicos-prestados-por-meio-de-tecnologia-da-informacao-e-da-comunicacao-durante-a-pandemia-do-covid-19.

CONSELHO FEDERAL DE PSICOLOGIA. **Resolução nº 9, de 18 de julho de 2024**. Regulamenta o exercício profissional da Psicologia mediado por Tecnologias Digitais da Informação e da Comunicação (TDICs) em território nacional e revoga as Resoluções CFP nº 11/2018 e nº 04/2020. Brasília, 2024. Disponível em:https://atosoficiais.com.br/cfp/resolucao-do-exercicio-profissional-n-9-2024-regulamenta-o-exercicio-profissional-da-psicologia-mediado-por-tecnologias-digitais-da-informacao-e-da-comunicacao-tdics-em-territorio-nacional-e-revoga-as-resolucao-cfp-n%C2%BA-11-de-11-de-maio-de-2018-e-resolucao-cfp-n%C2%BA-04-de-26-de-marco-de-2020.

CONSELHO FEDERAL DE PSICOLOGIA. **Código de Ética Profissional do Psicólogo.** Brasília, 2005. Disponível em: https://site.cfp.org.br/wp-content/uploads/2012/07/codigo-de-etica-psicologia.pdf.

COULOURIS, George et al. **Sistemas Distribuídos-: Conceitos e Projeto**. Bookman Editora, 2013. Disponível em: https://books.google.com.br/books?hl=pt-BR&lr=&id=6WU3AgAAQBAJ&oi=fnd&pg=PR3&dq=COULOURIS,+George%3B+Dollimore,+Jean%3B+Kindberg,+Tim.+Sistemas+Distribu%C3%ADdos:+Conceitos+e+Projeto.+5+ed..+Editora:+Bookman,+2013&ots=Fj9biM_PiZ&sig=2v-kHIxMlxYU1uuor9t4zeE9Tzw&redir_esc=y#v=onepage&q&f=false.

CREMER, Frank; SHEEHAN, Barry; FORTMANN, Michael; KIA, Arash N.; MULLINS, Martin; MURPHY, Finbarr; MATERNE, Stefan. **Cyber risk and cybersecurity: a systematic review of data availability.** The Geneva Papers on Risk and Insurance – Issues and Practice, v. 47, p. 698–736, 2022. DOI: 10.1057/s41288-022-00266-6. Disponível em: https://link.springer.com/article/10.1057/s41288-022-00266-6.

DA SILVA, Kayla Niandra et al. TELEPSICOTERAPIA EM TEMPOS DE PANDEMIA. **Revista Jovens Pesquisadores**, v. 12, n. 1, 2022. Disponível em: https://seer.unisc.br/index.php/jovenspesquisadores/article/view/17456.

DE OLIVEIRA BRASIL, Anderson; BORBA, Jean Marlos Pinheiro. Aporte teórico e legal do atendimento psicológico on-line no Brasil. **Revista NUFEN: Phenomenology and Interdisciplinarity**, v. 15, n. 02, 2023. Disponível em: https://submission-pepsic.scielo.br/index.php/nufen/article/view/23859.

DOCKER. **What is Docker?** In: Docker Documentation — Get Started. Disponível em: https://docs.docker.com/get-started/docker-overview/.

DOMINGUES, Emily Bezerra; AUGUSTINI, Gustavo Henrique; OLIVEIRA, Wdson de. **A importância da segurança em banco de dados: garantindo a proteção de informações sensíveis.** In: IV Congresso de Segurança da Informação das Fatec – FatecSeg, 2024. Disponível em: https://fatecseg.fatecourinhos.edu.br/index.php/fatecseg/article/view/261.

FERNANDES, Marcos Rodrigues; GALLINDO, Erica de Lima; DAMASCENO, Alexsandro Lima. **Fortalecendo a segurança da informação em órgãos públicos: estudo e consolidação de modelos existentes**. Aracati: Instituto Federal de Educação, Ciência e Tecnologia do Ceará, 2024. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação). Disponível em: https://gestaoaracati.ifce.edu.br/attachments/download/2779/tcc.pdf.

GONÇALVES, Charlisson Mendes; NETO, João Leite Ferreira. O atendimento psicológico on-line: Uma revisão sistemática da literatura. **Revista Foco**, v. 16, n. 5, p. e1723-e1723, 2023. Disponível em: https://ojs.focopublicacoes.com.br/foco/article/view/1723.

HARVARD BUSINESS REVIEW. **Why Data Breaches Spiked in 2023**. 2024. Disponível em: https://hbr.org/2024/02/why-data-breaches-spiked-in-2023.

IBM. **Database Security: Best Practices and Threats**. 2024. Disponível em: https://www.ibm.com/br-pt/think/topics/database-security.

IPFS. **Docs – IPFS Documentation.** Disponível em: https://docs.ipfs.tech/.

IPFS. **How IPFS works – Subsystems overview**. Disponível em: https://docs.ipfs.tech/concepts/how-ipfs-works/#subsystems-overview.

JSON Web Token (JWT). **When to use JSON Web Tokens.** Disponível em: https://www.jwt.io/introduction#when-to-use-json-web-tokens.

KHALID, Muhammad Irfan; EHSAN, Ibtisam; AL-ANI, Ayman Khallel; IQBAL, Jawaid; HUSSAIN, Saddam; ULLAH, Syed Sajid; NAYAB. **A comprehensive survey on blockchain-based decentralized storage networks**. IEEE Access, 2023. DOI: 10.1109/ACCESS.2023.3240237. Disponível em: https://www.researchgate.net/publication/367203995_A_Comprehensive_Survey_on_Blockchain-Based_Decentralized_Storage_Networks.

LI, Chuanlei; XU, Minghui; ZHANG, Jiahao; GUO, Hechuan; CHENG, Xiuzhen. **SoK: Decentralized storage network.** High-Confidence Computing, v. 4, p. 100239, 2024. DOI: https://doi.org/10.1016/j.hcc.2024.100239.

LI, Xiaozhou; MORESCHINI, Sergio; ZHANG, Zheying; PALOMBA, Fabio; TAIBI, Davide. **The anatomy of a vulnerability database: a systematic mapping study**. Journal of Systems and Software, v. 201, e111679, 2023. DOI: 10.1016/j.jss.2023.111679. Disponível em: https://doi.org/10.1016/j.jss.2023.111679.

LINHARES, Matheus de Souza. **Uso do IPFS e Blockchain para descentralização do armazenamento de dados**. Serra: Instituto Federal do Espírito Santo, 2022. Trabalho de Conclusão de Curso (Tecnologia em Sistemas para Internet).

LINHARES, Milena Melo. **Uma aplicação para armazenamento de resultados de exames laboratoriais utilizando as tecnologias Blockchain e IPFS**. 2024. 74 f. Trabalho de Conclusão de Curso (Bacharelado em Sistemas de Informação) — Instituto Federal do Espírito Santo, Cachoeiro de Itapemirim, 2024. Disponível em: https://repositorio.ifes.edu.br/handle/123456789/5604.

NESTJS Documentation. **Docs – NestJS Documentation**. Disponível em: https://docs.nestjs.com/.

NUNES, Caroline; MA, Stephane; TEIXEIRA FILHO, Marcelo Silveira. **Armazenamento descentralizado no Sistema Único de Saúde brasileiro (SUS) usando InterPlanetary File System (IPFS) e Blockchain**. Revista de Direito, Viçosa, v. 13, n. 1, 2021. DOI: https://doi.org/10.32361/2021130111695.

PELINSON, Sandra Cristina. Os Desafios na troca de informação em Saúde (Interoperabilidade) em um ambiente organizacional de Cooperativas Médicas. **FGV Repositório Digital**, 2022. Disponível em: https://repositorio.fgv.br/server/api/core/bitstreams/0de01fce-e9ee-4595-9d15-efc941b4d698/content.

RSA SecurID. **RSA SecurID Documentation**. Disponível em:https://community.rsa.com/s/rsa-securid-documentation.

RUBIRA, Lucas Alves. **Bancos de dados NoSQL: vulnerabilidades de segurança encontrados no MongoDB**. 2025. 33 f. Trabalho de Conclusão de Curso (Tecnologia em Sistemas para Internet) — Universidade Tecnológica Federal do Paraná, Câmpus Toledo, 2025. Disponível em: http://repositorio.utfpr.edu.br/jspui/handle/1/36794.

SABLONE, Sara et al. Telepsychology revolution in the mental health care delivery: a global overview of emerging clinical and legal issues. **Forensic sciences research**, v. 9, n. 3, p. owae008, 2024. Disponível em: https://academic.oup.com/fsr/article/9/3/owae008/7597769.

SILVA, Thais Ribeiro da. **Uso das tecnologias da informação e comunicação por psicólogos no Sistema Único de Saúde (SUS)**. 2023. Tese de Doutorado. Universidade de São Paulo. Disponível em: https://www.teses.usp.br/teses/disponiveis/108/108131/tde-23062023-113621/publico/ThaisRibeiroDaSilvaVersaoCorrigida.pdf.

SUPABASE. **Docs – Supabase Documentation**. Disponível em: https://supabase.com/docs.

SWAGGER. **Swagger Documentation**. Disponível em: https://swagger.io/docs/.

TENAGLIA, Matheus Rodrigues. **Simulação de ataques cibernéticos nos dispositivos IoT em ambientes de saúde**. Goiânia: Pontifícia Universidade Católica de Goiás, 2023. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação). Disponível em: https://repositorio.pucgoias.edu.br/jspui/bitstream/123456789/7816/1/MATHEUS_TENAGLIA--TCC2pronto%20%282%29.pdf.

TRAUTWEIN, Dennis et al. Design and evaluation of IPFS: a storage layer for the decentralized web. In: **Proceedings of the ACM SIGCOMM 2022 Conference**. 2022. p. 739-752. Disponível em: https://dl.acm.org/doi/abs/10.1145/3544216.3544232.

ZHONG, Li. **A survey of prevent and detect access control vulnerabilities**. 2023. Disponível em: https://doi.org/10.48550/arXiv.2304.10600.