

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**

**JOSÉ RICARDO SERATHIUK DA SILVEIRA**

**D-COLMEIA: EXTENSÃO DE MODERAÇÃO DISTRIBUÍDA E  
COLABORATIVA PARA O ACTIVITYPUB**

**CURITIBA**

**2026**

**JOSÉ RICARDO SERATHIUK DA SILVEIRA**

**D-COLMEIA: EXTENSÃO DE MODERAÇÃO DISTRIBUÍDA E  
COLABORATIVA PARA O ACTIVITYPUB**

**D-COLMEIA: Distributed and Collaborative Moderation Extension  
for ActivityPub**

Dissertação apresentado(a) como requisito  
para obtenção do título(grau) de Mestre em  
Computação Aplicada, do Programa de Pós-  
Graduação em Computação Aplicada, da  
Universidade Tecnológica Federal do Paraná  
(UTFPR).

Orientadora: Profa. Dra. Ana Cristina Bar-  
reiras Kochem Vendramin  
Coorientador: Prof. Dr. Daniel Fernando Pi-  
gatto

**CURITIBA**

**2026**



[4.0 Internacional](https://creativecommons.org/licenses/by/4.0/)

Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es).

Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.



**Ministério da Educação**  
**Universidade Tecnológica Federal do Paraná**  
**Campus Curitiba**



JOSE RICARDO SERATHIUK DA SILVEIRA

**D-COLMEIA: EXTENSÃO DE MODERAÇÃO DISTRIBUÍDA E COLABORATIVA PARA O ACTIVITYPUB**

Trabalho de pesquisa de mestrado apresentado como requisito para obtenção do título de Mestre Em Computação Aplicada da Universidade Tecnológica Federal do Paraná (UTFPR). Área de concentração: Engenharia De Sistemas Computacionais.

Data de aprovação: 19 de Junho de 2026

Dra. Ana Cristina Barreiras Kochem Vendramin, Doutorado - Universidade Tecnológica Federal do Paraná

Dr. Alexandre Reis Graeml, Doutorado - Universidade Tecnológica Federal do Paraná

Dr. Cesar Augusto Tacla, Doutorado - Universidade Tecnológica Federal do Paraná

Dr. Joao Paulo Orlando, Doutorado - Instituto Federal de Educação, Ciência e Tecnologia do Paraná (Ifpr)

Documento gerado pelo Sistema Acadêmico da UTFPR a partir dos dados da Ata de Defesa em 19/06/2026.

## RESUMO

As redes sociais descentralizadas vêm crescendo nos últimos anos, impulsionadas por preocupações com privacidade, controle de dados e moderação de conteúdo em plataformas centralizadas. O protocolo ActivityPub, base do fediverso, permite a comunicação federada entre diferentes instâncias, mas impõe desafios para uma moderação colaborativa e descentralizada. Este trabalho propõe a D-COLMEIA (*Distributed and Collaborative Moderation Extension for ActivityPub*), uma extensão ao protocolo ActivityPub, voltada à moderação de conteúdo distribuída, baseada no protocolo Kademlia. A proposta busca estabelecer um modelo descentralizado de moderação colaborativa, utilizando uma DHT (*Distributed Hash Table*) para armazenamento e recuperação de informações sobre moderação, evitando pontos únicos de falha. A arquitetura proposta permite que diferentes instâncias colaborem em um sistema distribuído, preservando a autonomia local e viabilizando a aplicação de conceitos de inteligência coletiva e *crowdsourcing* na moderação de conteúdo. Também é apresentada uma ontologia para representação dos dados de moderação, distribuída entre os nós da rede. A viabilidade da abordagem é demonstrada por meio de um protótipo funcional, validado em um ambiente controlado.

**Palavras-chave:** Fediverso. ActivityPub. Moderação de Conteúdo. Hash Table Distribuída. Kademlia.

## ABSTRACT

Decentralized social networks have gained traction in recent years, driven by concerns over privacy, data control, and content moderation in centralized platforms. The ActivityPub protocol, which underpins the fediverse, enables federated communication between instances but presents challenges for collaborative and decentralized moderation. This work proposes a Distributed and Collaborative Moderation Extension for ActivityPub (D-COLMEIA), an extension to the ActivityPub protocol focused on distributed content moderation based on the Kademlia protocol. The solution establishes a decentralized model for collaborative moderation, using a Distributed Hash Table (DHT) to store and retrieve moderation-related information while avoiding single points of failure. The proposed architecture enables different instances to participate in a distributed system, preserving local autonomy and applying concepts of collective intelligence and crowdsourcing to content moderation. An ontology is also introduced to represent moderation data, distributed among the network's nodes. The approach is validated through a functional prototype tested in a controlled environment.

**Keywords:** Fediverse. ActivityPub. Content Moderation. Distributed Hash Table. Kademlia.

## LISTA DE ILUSTRAÇÕES

|                                                                                                                                                                                                      |    |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Figura 1 – Representação da web centralizada . . . . .                                                                                                                                               | 17 |
| Figura 2 – Representação da web descentralizada . . . . .                                                                                                                                            | 17 |
| Figura 3 – Exemplo de Objeto em <i>ActivityStreams 2.0</i> . . . . .                                                                                                                                 | 19 |
| Figura 4 – Exemplo de Atividade em <i>ActivityStreams 2.0</i> . . . . .                                                                                                                              | 20 |
| Figura 5 – Exemplo de uso de múltiplos contextos em <i>ActivityStreams 2.0</i> . . . . .                                                                                                             | 20 |
| Figura 6 – Funcionamento básico da comunicação utilizando ActivityPub. . . . .                                                                                                                       | 22 |
| Figura 7 – Funcionamento da comunicação entre usuários no ActivityPub. . . . .                                                                                                                       | 22 |
| Figura 8 – Exemplo das informações do usuário no ActivityPub. . . . .                                                                                                                                | 23 |
| Figura 9 – Ações de moderação para um usuário no Mastodon. . . . .                                                                                                                                   | 25 |
| Figura 10 – Denúncia pelo usuário no Mastodon. . . . .                                                                                                                                               | 25 |
| Figura 11 – Seleção de postagens a serem denunciadas no Mastodon. . . . .                                                                                                                            | 26 |
| Figura 12 – Detalhamento da denúncia do usuário no Mastodon. . . . .                                                                                                                                 | 26 |
| Figura 13 – Ações disponíveis ao usuário após a denúncia no Mastodon. . . . .                                                                                                                        | 27 |
| Figura 14 – Denúncia recebida pelo moderador no Mastodon. . . . .                                                                                                                                    | 27 |
| Figura 15 – Ações de moderação disponíveis ao moderador no Mastodon. . . . .                                                                                                                         | 28 |
| Figura 16 – Moderação de domínio no Mastodon. . . . .                                                                                                                                                | 29 |
| Figura 17 – Densidade de palavras-chave relacionadas com <i>distributed hash table</i> na base<br><i>Scopus</i> , no período de 2015 até 2025. . . . .                                               | 31 |
| Figura 18 – Árvore binária de 4 níveis com rótulos binários . . . . .                                                                                                                                | 33 |
| Figura 19 – Representação de uma tabela de roteamento <i>k-buckets</i> de um nó em rede<br><i>Kademlia</i> de 4 bits. . . . .                                                                        | 34 |
| Figura 20 – Arquitetura do D-COLMEIA . . . . .                                                                                                                                                       | 43 |
| Figura 21 – Exemplo do Objeto <i>ModerationNode</i> do <i>D-COLMEIA</i> . . . . .                                                                                                                    | 47 |
| Figura 22 – Exemplo do Objeto <i>dcolmeia:CommunityNote</i> do <i>D-COLMEIA</i> . . . . .                                                                                                            | 48 |
| Figura 23 – Exemplo do Objeto <i>dcolmeia:CommunityNoteGroup</i> do <i>D-COLMEIA</i> , com<br>a lista de itens resumida. . . . .                                                                     | 49 |
| Figura 24 – Requisição <i>as:Follow</i> para um <i>dcolmeia:CommunityNoteGroup</i> . . . . .                                                                                                         | 50 |
| Figura 25 – Resposta <i>as:Accept</i> do <i>dcolmeia:CommunityNoteGroup</i> . . . . .                                                                                                                | 51 |
| Figura 26 – <i>as:Undo/as:Follow</i> para deixar de seguir um <i>dcolmeia:CommunityNoteGroup</i> . . . . .                                                                                           | 51 |
| Figura 27 – <i>as:Announce</i> de nova <i>dcolmeia:CommunityNote</i> para seguidores. . . . .                                                                                                        | 52 |
| Figura 28 – <i>as:Like</i> em uma <i>dcolmeia:CommunityNote</i> . . . . .                                                                                                                            | 52 |
| Figura 29 – <i>as:Dislike</i> em uma <i>dcolmeia:CommunityNote</i> . . . . .                                                                                                                         | 53 |
| Figura 30 – Exemplo do retorno WebFinger para um ator do D-COLMEIA. . . . .                                                                                                                          | 57 |
| Figura 31 – Adaptação do FIND_VALUE do Kademlia para o ActivityPub. . . . .                                                                                                                          | 59 |
| Figura 32 – Adaptação do FIND_NODE do Kademlia para o ActivityPub, com foco no<br>comportamento adotado quando o valor não é encontrado. . . . .                                                     | 61 |
| Figura 33 – Adaptação do STORE do Kademlia para o ActivityPub. . . . .                                                                                                                               | 62 |
| Figura 34 – Atividade <i>as:Follow</i> para um <i>dcolmeia:CommunityNoteGroup</i><br>enviada para a <i>outbox</i> do usuário. . . . .                                                                | 68 |
| Figura 35 – <i>Log</i> da primeira operação <i>Follow</i> , com criação do<br><i>dcolmeia:CommunityNoteGroup</i> sob demanda. . . . .                                                                | 69 |
| Figura 36 – <i>Log</i> da segunda operação <i>Follow</i> , com grupo já existente. . . . .                                                                                                           | 71 |
| Figura 37 – Mensagem <i>POST</i> para a <i>outbox</i> ActivityPub <a href="http://social.vivaldi.net.local/users/admin/outbox">http://social.vivaldi.net.local/<br/>users/admin/outbox</a> . . . . . | 72 |

|                                                                                                                           |    |
|---------------------------------------------------------------------------------------------------------------------------|----|
| Figura 38 – <i>Log</i> resumido da operação STORE no D-COLMEIA, com múltiplos saltos e notificação de seguidores. . . . . | 73 |
| Figura 39 – Atividade <code>dcolmeia:FindValue</code> enviada para a <i>outbox</i> do usuário. . . .                      | 75 |
| Figura 40 – <i>Log</i> da operação FIND_VALUE iterativa no D-COLMEIA, com múltiplos saltos. . . . .                       | 76 |
| Figura 41 – Representação gráfica da ontologia do D-Colmeia. . . . .                                                      | 89 |

## **LISTA DE QUADROS**

|                                                                     |    |
|---------------------------------------------------------------------|----|
| Quadro 1 – Características dos trabalhos relacionados . . . . .     | 40 |
| Quadro 2 – Descrição das Classes da Ontologia do D-COLMEIA. . . . . | 46 |
| Quadro 3 – Tipologia de avisos de conteúdo NEON. . . . .            | 90 |

## LISTA DE TABELAS

|                                                                                                                                                                                                                |    |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Tabela 1 – Distribuição dos nós participantes no espaço de endereçamento <i>Secure Hash Algorithm 1</i> (SHA-1) da rede D-COLMEIA . . . . .                                                                    | 67 |
| Tabela 2 – Os 10 nós D-COLMEIA com menor distância XOR ao recurso <a href="https://worstcontent.social/posts/abcd-1234">https://worstcontent.social/posts/abcd-1234</a> (hash $1,11 \times 10^{48}$ ). . . . . | 68 |

## LISTA DE SIGLAS E ACRÔNIMOS

|           |                                                                           |
|-----------|---------------------------------------------------------------------------|
| API       | <i>Application Programming Interface</i>                                  |
| D-COLMEIA | <i>Distributed and Collaborative Moderation Extension for ActivityPub</i> |
| DHT       | <i>Distributed Hash Table</i>                                             |
| DID       | <i>Decentralized Identifier</i>                                           |
| DOSN      | <i>Decentralized Open Social Network</i>                                  |
| FEP       | <i>Fediverse Enhancement Proposal</i>                                     |
| HTTP      | <i>Hypertext Transfer Protocol</i>                                        |
| IA        | Inteligência Artificial                                                   |
| ID        | Identificador                                                             |
| IETF      | <i>Internet Engineering Task Force</i>                                    |
| IMAP      | <i>Internet Message Access Protocol</i>                                   |
| IPFS      | <i>InterPlanetary File System</i>                                         |
| JSON      | <i>JavaScript Object Notation</i>                                         |
| JSON-LD   | <i>JavaScript Object Notation for Linked Data</i>                         |
| OWA       | <i>Open World Assumption</i>                                              |
| OWL       | <i>Ontology Web Language</i>                                              |
| P2P       | <i>Peer-to-Peer</i>                                                       |
| POP3      | <i>Post Office Protocol version 3</i>                                     |
| RDF       | <i>Resource Description Framework</i>                                     |
| RFC       | <i>Request for Comments</i>                                               |
| RFC 7033  | <i>RFC 7033: WebFinger Specification</i>                                  |
| RPC       | <i>Remote Procedure Call</i>                                              |
| SHA-1     | <i>Secure Hash Algorithm 1</i>                                            |
| SMTP      | <i>Simple Mail Transfer Protocol</i>                                      |
| Turtle    | <i>Terse RDF Triple Language</i>                                          |
| URI       | <i>Uniform Resource Identifier</i>                                        |
| URL       | <i>Uniform Resource Locator</i>                                           |
| W3C       | <i>World Wide Web Consortium</i>                                          |
| XML       | <i>Extensible Markup Language</i>                                         |
| XOR       | <i>Exclusive OR (Ou Exclusivo)</i>                                        |

## SUMÁRIO

|          |                                                                                                       |           |
|----------|-------------------------------------------------------------------------------------------------------|-----------|
| <b>1</b> | <b>INTRODUÇÃO . . . . .</b>                                                                           | <b>11</b> |
| 1.1      | OBJETIVO GERAL E OBJETIVOS ESPECÍFICOS . . . . .                                                      | 12        |
| 1.2      | ESTRUTURA DO DOCUMENTO . . . . .                                                                      | 13        |
| <b>2</b> | <b>FUNDAMENTAÇÃO TEÓRICA . . . . .</b>                                                                | <b>14</b> |
| 2.1      | REDES SOCIAIS . . . . .                                                                               | 14        |
| 2.2      | FEDIVERSO . . . . .                                                                                   | 16        |
| 2.3      | WEB DE DADOS . . . . .                                                                                | 18        |
| 2.4      | ACTIVITYPUB . . . . .                                                                                 | 21        |
| 2.5      | MODERAÇÃO DE CONTEÚDO NO FEDIVERSO . . . . .                                                          | 24        |
| 2.6      | INTELIGÊNCIA COLETIVA E MODERAÇÃO DE CONTEÚDO CO-<br>LABORATIVA . . . . .                             | 26        |
| 2.7      | SISTEMAS PEER-TO-PEER (P2P) . . . . .                                                                 | 29        |
| 2.8      | TABELAS <i>HASH</i> DISTRIBUÍDAS E O KADEMLIA . . . . .                                               | 32        |
| 2.9      | TRABALHOS RELACIONADOS . . . . .                                                                      | 35        |
| <b>3</b> | <b>D-COLMEIA: EXTENSÃO DE MODERAÇÃO DISTRIBUÍDA E CO-<br/>LABORATIVA PARA O ACTIVITYPUB . . . . .</b> | <b>42</b> |
| 3.1      | METODOLOGIA DE DESENVOLVIMENTO . . . . .                                                              | 44        |
| 3.2      | ONTOLOGIA DO D-COLMEIA . . . . .                                                                      | 44        |
| 3.3      | COMUNICAÇÃO ENTRE OS NÓS D-COLMEIA . . . . .                                                          | 56        |
| <b>4</b> | <b>RESULTADOS . . . . .</b>                                                                           | <b>65</b> |
| 4.1      | PROTÓTIPO . . . . .                                                                                   | 65        |
| 4.2      | CRIAÇÃO DE NOTAS DE MODERAÇÃO . . . . .                                                               | 66        |
| 4.3      | BUSCA DE UM <i>DCOLMEIA:COMMUNITYNOTEGROUP</i> . . . . .                                              | 74        |
| 4.4      | DISCUSSÃO DOS RESULTADOS . . . . .                                                                    | 77        |
| <b>5</b> | <b>CONCLUSÃO . . . . .</b>                                                                            | <b>79</b> |
| 5.1      | TRABALHOS FUTUROS . . . . .                                                                           | 80        |
|          | <b>REFERÊNCIAS . . . . .</b>                                                                          | <b>82</b> |
|          | <b>APÊNDICE A – REPRESENTAÇÃO GRÁFICA DA ONTOLOGIA<br/>DO D-COLMEIA. . . . .</b>                      | <b>89</b> |
|          | <b>ANEXO A – TIPOLOGIA DE AVISOS DE CONTEÚDO NEON. . . . .</b>                                        | <b>90</b> |

## 1 INTRODUÇÃO

Com o advento do protocolo ActivityPub, reconhecido como especificação oficial pela *World Wide Web Consortium* (W3C) em 2018, diversas implementações de redes sociais federadas de código aberto baseadas nesse protocolo foram desenvolvidas, como o *Mastodon* (MASTODON, 2021), o *Pixelfed* (PIXELFED, 2022) e o *Peertube* (PEERTUBE, 2023). Além disso, também foram lançadas iniciativas comerciais, como o *Threads* (META, 2023), da Meta.

Com o crescimento das redes sociais descentralizadas, alguns problemas já observados em redes sociais convencionais também passaram a ser identificados em redes sociais federadas baseadas no protocolo ActivityPub. Thiel *et al.* (2023) apontam, por exemplo, questões relacionadas à exploração sexual infantil e à segurança de menores nessas plataformas. Já Anaobi *et al.* (2023) destacam dificuldades na moderação de conteúdo, como o caráter repetitivo do trabalho dos moderadores, o que torna o processo de remoção de conteúdo indesejado lento e exaustivo.

Zia *et al.* (2022), Wiegmann *et al.* (2024), Anaobi *et al.* (2023) apontam que os problemas relacionados à moderação de conteúdo no fediverso decorrem da ausência de ferramentas adequadas e do fato de cada instância (servidor/nó no fediverso) ser responsável pela moderação de forma independente. Os autores também apontam que quando a instância *gab.social* ingressou no fediverso, ela propagou rapidamente discursos de ódio para outras instâncias, cabendo a cada uma lidar de forma autônoma com a questão, com bloqueio da mesma. Os autores também argumentam que a criação de um serviço centralizado de moderação, como tentativa de mitigar esse problema, contraria os princípios da Web Descentralizada. Além disso, os autores relatam que a definição do que constitui conteúdo “tóxico” varia de acordo com cada instância e suas respectivas políticas de moderação.

Existem instâncias voltadas à disseminação de conteúdos antissociais, tóxicos e nocivos, os quais são enfrentados por meio de rigorosa moderação local (ABBING *et al.*, 2023; SABO *et al.*, 2024). Cada instância é livre para definir suas próprias regras, e não há uma autoridade central capaz de moderar ou banir usuários e instâncias em escala global (WIEGMANN *et al.*, 2024).

Alguns estudos apontam que o bloqueio indiscriminado pode resultar em bloqueios injustos de instâncias, ou até mesmo transformar uma instância em uma câmara de eco, onde todos os participantes tendem a concordar entre si, geralmente sobre aspectos políticos e ideológicos, e onde não há diversidade de opiniões e ideias, o que contraria os princípios de uma rede social

saudável (CAVA *et al.*, 2024; BONO *et al.*, 2024; FRIEDL; MORGAN, 2024; MELDER *et al.*, 2025). Também foi observado que o bloqueio em massa tende a isolar completamente instâncias que permitem comportamentos nocivos (CAVA *et al.*, 2024; ABBING *et al.*, 2023; FRIEDL; MORGAN, 2024). Diante disso, os administradores enfrentam o desafio de equilibrar a moderação de forma que sua instância não se torne isolada, mas que também evite a comunicação com instâncias consideradas prejudiciais (MELDER *et al.*, 2025).

Para tentar equilibrar a moderação, torna-se necessário um modelo mais uniforme e colaborativo. Existem experiências em redes sociais centralizadas que utilizam conceitos de inteligência coletiva, como o *crowdsourcing*, e que podem ser adaptadas ao contexto do fediverso. No entanto, considerando os pontos expostos anteriormente, a natureza descentralizada e federada dessas redes torna desafiadora a definição de um modelo de moderação que seja, simultaneamente, colaborativo, descentralizado e capaz de manter interações de forma saudável, evitando a polarização de pensamento no fediverso (CAVA *et al.*, 2024).

## 1.1 OBJETIVO GERAL E OBJETIVOS ESPECÍFICOS

O objetivo geral deste trabalho é propor o D-COLMEIA, uma extensão ao protocolo ActivityPub voltada à moderação de conteúdo distribuída e colaborativa, que organiza uma rede de sobreposição baseada nos princípios da *Distributed Hash Table* (DHT) *Kademlia* para o armazenamento e a recuperação distribuída de informações de moderação no fediverso, sem depender de uma autoridade central. Essa extensão é chamada de *Distributed and Collaborative Moderation Extension for ActivityPub* (D-COLMEIA).

O trabalho possui a seguinte lista de objetivos específicos, essenciais para o alcance do objetivo geral:

- Definir um padrão semântico de mensagens baseado no protocolo ActivityPub, sustentado por uma ontologia *Ontology Web Language* (OWL) que estende o ActivityStreams 2.0, viabilizando a representação interoperável de dados de moderação;
- Desenvolver e documentar os algoritmos de distribuição e busca, fundamentados no protocolo *Kademlia* e adaptados ao modelo de comunicação assíncrona do ActivityPub, permitindo a disseminação eficiente e determinística de dados de moderação sem controle centralizado;

- Projetar e implementar um protótipo funcional que integre uma rede de sobreposição baseada em DHT com plataformas sociais federadas;
- Analisar o funcionamento da rede DHT voltada à moderação e sua integração com o fediverso;
- Realizar a validação experimental da abordagem proposta, demonstrando a correção das operações de protocolo, a viabilidade de integração com infraestruturas existentes e a efetividade da combinação de mecanismos de disseminação baseados em *pull* e *push*.

Cabe destacar que análises de desempenho e escalabilidade estão fora do escopo deste trabalho e são deixadas para investigação futura.

A extensão proposta ao protocolo deverá permitir que, de forma totalmente descentralizada, os modelos de moderação de conteúdo possam ser mantidos colaborativamente, com múltiplos mantenedores, sem depender da infraestrutura de um único servidor ou instância. Com isso, busca-se viabilizar que usuários de diferentes instâncias do fediverso, por meio do uso de *crowdsourcing*, realizem a moderação da rede social de maneira mais efetiva, aproveitando-se do conhecimento coletivo gerado a partir das decisões individuais de cada usuário.

## 1.2 ESTRUTURA DO DOCUMENTO

Este trabalho está estruturado em cinco capítulos. O Capítulo 1 apresenta a introdução do trabalho, bem como suas motivações e objetivos. O Capítulo 2 dedica-se à fundamentação teórica, explicando conceitos relevantes para o desenvolvimento da pesquisa, como os princípios do fediverso e seu protocolo ActivityPub, inteligência coletiva, *crowdsourcing* e suas aplicações na moderação de conteúdo. Também são abordadas as bases teóricas da moderação de conteúdo, além dos fundamentos sobre sistemas *Peer-to-Peer* (P2P) e baseados em DHT, com ênfase no protocolo *Kademlia*. O capítulo é finalizado com a descrição dos trabalhos relacionados aos temas abordados. O Capítulo 3 detalha a arquitetura desenvolvida para a moderação descentralizada no fediverso. O Capítulo 4 apresenta os resultados obtidos. Finalmente, o Capítulo 5 traz as considerações finais.

## 2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo tem como objetivo apresentar e descrever conceitos de fundamental importância para o entendimento da extensão proposta neste trabalho.

Inicialmente, são discutidas as características das redes sociais (Seção 2.1), do Fediverso (Seção 2.2), da Web de Dados (Seção 2.3) e do protocolo ActivityPub (Seção 2.4), além dos mecanismos utilizados hoje para moderação de conteúdo (Seção 2.5). Em seguida, abordam-se os conceitos de inteligência coletiva e *crowdsourcing*, explorando sua relação com a moderação de conteúdo em redes sociais (Seção 2.6). Também são explorados conceitos relacionados a sistemas P2P (Seção 2.7) e tabelas *hash* distribuídas e o Kademlia (Seção 2.8). Por fim, são descritos os trabalhos relacionados ao tema desta pesquisa (Seção 2.9), os quais contribuem para o aprofundamento dos tópicos abordados e evidenciam como eles vêm sendo tratados na literatura.

### 2.1 REDES SOCIAIS

Segundo Abbing *et al.* (2023), redes sociais centralizadas são aquelas amplamente conhecidas e utilizadas, como o *X* (antigo *Twitter*), *Facebook* e *Instagram*. Esse tipo de rede social é controlado por uma única empresa, o que acarreta problemas de privacidade e censura que, conforme destacado por Marx e Cheong (2023), são questões que as redes sociais descentralizadas buscam solucionar.

Prodromou (2024) ressalta que “*uma rede social é um grupo de pessoas e as conexões que essas pessoas formam entre elas*”. O autor também afirma que “*assim como as redes de computadores ou as redes de televisão, a informação circula por essas conexões*”. Ele descreve que redes sociais podem apresentar os seguintes elementos:

- *Identidade*: uma identidade única para as pessoas usuárias, como um nome de usuário ou identificador;
- *Perfil*: as informações de uma pessoa na rede social, como *hobbies*, localização, idade etc.;
- *Relacionamentos*: as conexões entre usuários. Algumas redes sociais definem essa relação como amizade, com vínculo bidirecional; outras utilizam a dinâmica de seguidor/seguido,

uma relação unidirecional; e há também redes com grupos, nos quais a relação se dá entre muitas pessoas em torno de um tema específico;

- *Postagem de conteúdo*: o conteúdo gerado por usuários, que pode incluir textos de diferentes tamanhos, uma ou mais imagens, vídeos, áudios etc.;
- *Timeline do usuário*: seção da rede social onde são exibidos os conteúdos provenientes dos relacionamentos estabelecidos na rede;
- *Reações*: ações em resposta às postagens de conteúdo. Normalmente, o usuário pode reagir com um botão de “curtir”, com *emojis* ou respondendo à postagem com outro conteúdo.
- *Repostagem*: o ato de compartilhar a postagem de outro usuário com aqueles com quem se mantém relacionamento na rede, como amigos ou seguidores.

Prodromou (2024) também aponta ações que não são possíveis em redes sociais convencionais ou centralizadas:

- Conectar-se com alguém de outra rede social;
- Visualizar conteúdo de uma rede social em outra;
- Migrar a conta de usuário de uma rede social para outra;
- Ter controle total sobre o conteúdo visualizado, podendo escolher o que ver ou não ver;
- Escolher o modelo de negócios, como optar entre ver anúncios, pagar uma assinatura ou utilizar a rede sem contrapartidas financeiras;
- Buscar informações da forma desejada, saber como estão sendo utilizadas e exportá-las para evitar perda de dados em caso de encerramento da rede.

Nos últimos anos, redes sociais como o *Twitter* passaram por mudanças radicais em seu funcionamento, gerando insatisfação entre muitos usuários diante de políticas impostas unilateralmente (BONO *et al.*, 2024). Em resposta, surgiram novas redes sociais, como o *Mastodon*, o *Threads* e o *Bluesky*. As três podem ser consideradas redes sociais descentralizadas: o *Threads* e o *Mastodon* são baseados no protocolo ActivityPub, enquanto o *Bluesky* utiliza o protocolo *AT Proto* (WIEGMANN *et al.*, 2024).

De acordo com Guidi *et al.* (2022), “*sistemas descentralizados são aqueles em que um conjunto de entidades interage e coopera para alcançar um objetivo comum ou compartilhado*”. Os autores ressaltam que as *Decentralized Open Social Network* (DOSN) são implementadas sobre plataformas de gerenciamento de informações distribuídas, eliminando a dependência de um único provedor de serviços e oferecendo maior controle sobre os dados privados. A descentralização busca retirar o controle da informação das mãos de uma única organização.

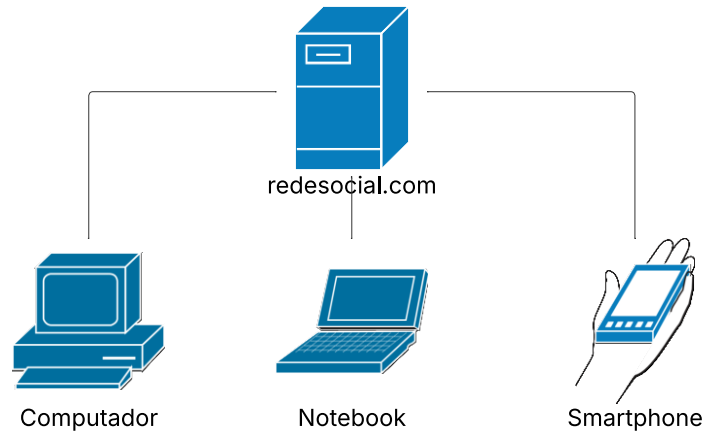
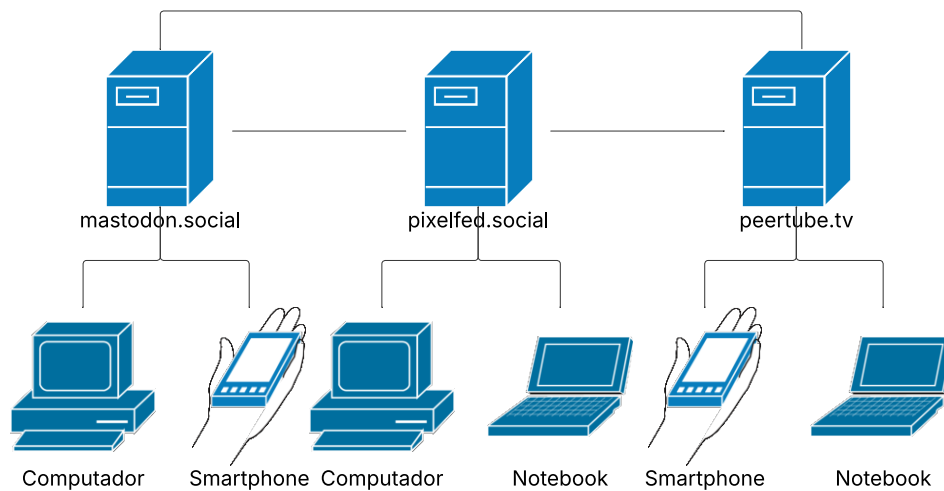
## 2.2 FEDIVERSO

O fediverso, ou “universo federado” é um ecossistema de redes sociais descentralizadas, composto por um conjunto de instâncias (servidores) compatíveis entre si, que utilizam um protocolo comum para comunicação (ABBING *et al.*, 2023). Cada instância opera de forma independente, com administração e infraestrutura próprias, e comunica-se com as demais por meio de protocolos abertos, como o ActivityPub. Como cada instância possui sua própria moderação local, a administração é voltada exclusivamente aos usuários hospedados naquela instância, resultando em uma gestão mais próxima e contextualizada. Essa independência é considerada uma qualidade essencial das redes sociais descentralizadas, em contraste com as redes centralizadas, nas quais os usuários estão sujeitos a uma moderação única, muitas vezes ineficiente, sem a possibilidade de escolher uma comunidade alinhada a seus valores (FRIEDL; MORGAN, 2024).

Segundo Prodromou (2024), o fediverso pode ser descrito como uma *web social*, conceito que aplica a lógica da interconexão de redes ao contexto das redes sociais. Trata-se de um conjunto de plataformas sociais que permite a interação entre usuários, mesmo quando pertencem a instâncias ou aplicações distintas. Na Figura 1, o autor apresenta uma arquitetura centralizada, na qual todos os clientes se comunicam com um único servidor. Em contraste, a Figura 2 ilustra uma arquitetura descentralizada, na qual os clientes podem se conectar a diferentes servidores, e esses servidores interagem entre si utilizando um protocolo aberto.

Como software, as instâncias do fediverso são, em sua maioria, aplicações de código aberto, como o *Mastodon*, *Peertube* e *Pixelfed*. Esses softwares implementam o protocolo ActivityPub para fornecer serviços de rede social a um número potencialmente amplo de usuários. A principal exceção é o *Threads*, cuja base de código é fechada, embora ainda mantenha interoperabilidade com o fediverso.

Os problemas associados ao X (*antigo Twitter*) e a outras plataformas centralizadas,

**Figura 1 – Representação da web centralizada****Fonte: Autoria Própria.****Figura 2 – Representação da web descentralizada****Fonte: Autoria Própria.**

nos últimos anos, estimularam o crescimento do número de usuários que optaram por migrar para o *Mastodon* e demais instâncias do fediverso (BONO *et al.*, 2024; CAVA *et al.*, 2024). Brembs *et al.* (2023) discutem a migração da comunidade acadêmica para o *Mastodon*, motivada pela aquisição do *Twitter* pelo empresário *Elon Musk*. Segundo os autores, a saída em massa da plataforma teve início antes mesmo da concretização da compra, impulsionada por uma série de ameaças direcionadas a minorias na rede. Os autores também destacam que o fediverso pode ser considerado um repositório aberto, característica que se alinha aos interesses da academia.

Wiegmann *et al.* (2024) e Anaobi *et al.* (2023) também observam que o modelo de administração adotado pelo *Twitter/X*, após sua aquisição por *Elon Musk*, intensificou o descontentamento e acelerou a adoção de redes descentralizadas. O número de usuários no fediverso

mais que dobrou após a aquisição (SABO *et al.*, 2024). A descentralização do poder sobre a rede é apontada como uma das principais razões para a migração em direção às redes sociais descentralizadas (ABBING *et al.*, 2023).

### 2.3 WEB DE DADOS

A *Web de Dados* é descrita por Isotani e Bittencourt (2022) como um conjunto de boas práticas para publicar e conectar conjuntos de dados estruturados na *web*. Ela está fortemente relacionada ao conceito de Dados Conectados (*Linked Data*), que permite, por meio do padrão *Resource Description Framework* (RDF), a definição de recursos para a *web*. O RDF funciona como uma linguagem de representação de informações na *web*, permitindo que recursos sejam descritos formalmente e interpretados por máquinas.

Sendo o RDF um modelo para descrever recursos na *web*, cada recurso deve possuir um identificador único e global, conhecido como *Uniform Resource Identifier* (URI). Esse identificador permite que um recurso — seja ele um texto, uma imagem ou outro tipo de documento — possa ser reconhecido globalmente. O RDF pode ser representado em diferentes formatos, como *Extensible Markup Language* (XML), *Terse RDF Triple Language* (Turtle) e *JavaScript Object Notation for Linked Data* (JSON-LD).

O JSON-LD é um formato baseado em JSON para serialização de dados conectados (*linked data*) (W3C, 2020), compatível com o modelo de dados do RDF (ISOTANI; BITTENCOURT, 2022). Todo documento JSON-LD possui os atributos `@context` e Identificador (ID). O atributo `@context` indica o “contexto da comunicação”, e o atributo ID representa seu identificador único dentro desse contexto.

O *ActivityStreams 2.0*, sobre o qual o protocolo ActivityPub é construído, é uma especificação que, segundo W3C (2017), define um modelo para representar atividades potenciais e concluídas utilizando o formato JSON-LD. Essa especificação foi projetada para ser utilizada com vocabulários que descrevem a estrutura de diferentes atividades e definem tipos específicos de ações.

De acordo com W3C (2017), o *ActivityStreams 2.0* é dividido nos seguintes tipos de dados/recursos:

- *Actor (Ator)*: pode ser uma pessoa, grupo, organização, serviço ou aplicação. Representa qualquer entidade capaz de executar uma ação;

- *Object (Objeto)*: são itens que podem receber ou compor uma ação. Incluem imagens, vídeos, textos ou mesmo objetos mais abstratos, como localizações, eventos e relações. São elementos que podem ser criados, compartilhados, atualizados ou removidos, entre outras ações;
- *Link*: uma referência indireta a outro recurso;
- *Activity (Atividade)*: representa ações que podem ser realizadas, como criar, ler, atualizar, remover, enviar, entre outras. Uma atividade pode ser transitiva ou intransitiva, isto é, pode ou não estar associada a um objeto;
- *Collection (Coleção)*: um agrupamento de objetos ou links.

A Figura 3 apresenta um exemplo de objeto *ActivityStreams 2.0* em formato *JavaScript Object Notation (JSON)*. A notação `@context`, com o valor `https://www.w3.org/ns/activitystreams`, indica que se trata de um documento JSON-LD estruturado conforme a especificação do *ActivityStreams 2.0*. O atributo `id` define o identificador único do objeto. O atributo `type` especifica que o tipo do objeto é `Note` (uma nota). O campo `name` define o título da nota, enquanto o atributo `published` indica a data de publicação. Por fim, o atributo `attributedTo` referencia outro objeto do tipo `Person`, indicando a qual *Ator* a nota está atribuída.

**Figura 3 – Exemplo de Objeto em *ActivityStreams 2.0*.**

```

1  {
2    "@context": "https://www.w3.org/ns/activitystreams",
3    "id": "http://example.org/foo",
4    "type": "Note",
5    "name": "My favourite stew recipe",
6    "attributedTo": {
7      "id": "http://joe.website.example/",
8      "type": "Person",
9      "name": "Joe Smith"
10   },
11   "published": "2014-08-21T12:34:56Z"
12 }

```

**Fonte: W3C (2017)**

A Figura 4 apresenta um exemplo de uma *Atividade*. O atributo `type` define o tipo do objeto como `Like`, indicando que se trata de uma ação de "curtir". O atributo `actor` especifica

a URI da `Person` que executou a atividade. Já o atributo `object` referencia o *Objeto* vinculado à atividade.

**Figura 4 – Exemplo de Atividade em *ActivityStreams 2.0*.**

```

1 {
2   "@context": "https://www.w3.org/ns/activitystreams",
3   "summary": "Joe liked a note",
4   "type": "Like",
5   "id": "http://www.test.example/activity/1",
6   "actor": "http://example.org/profiles/joe",
7   "object": "http://example.com/notes/1",
8   "published": "2014-09-30T12:34:56Z"
9 }

```

**Fonte: W3C (2017)**

A Figura 5 apresenta um exemplo de como utilizar múltiplos contextos em um mesmo documento. No objeto `@context`, é definido o atributo `@vocab`, que especifica o vocabulário padrão, além da definição de um contexto adicional, denominado *ext*, associado à `https://canine-extension.example/terms/`.

**Figura 5 – Exemplo de uso de múltiplos contextos em *ActivityStreams 2.0*.**

```

1 {
2   "@context": {
3     "@vocab": "https://www.w3.org/ns/activitystreams",
4     "ext": "https://canine-extension.example/terms/",
5     "@language": "en"
6   },
7   "summary": "A note",
8   "type": "Note",
9   "content": "My dog has fleas.",
10  "ext:nose": 0,
11  "ext:smell": "terrible"
12 }

```

**Fonte: W3C (2017)**

## 2.4 ACTIVITYPUB

O protocolo ActivityPub é a base de comunicação do fediverso. De acordo com W3C (2018), o ActivityPub é um protocolo de rede social descentralizado baseado no formato de dados *ActivityStreams 2.0*, descrito na Seção 2.3.

O ActivityPub fornece uma *Application Programming Interface* (API) cliente-servidor para criar, atualizar e excluir conteúdo, além de uma API federada servidor-servidor para entregar notificações e conteúdo. Este trabalho não abordará a comunicação cliente-servidor, que trata apenas da interação entre clientes e seus respectivos servidores, por não ser relevante para os objetivos aqui propostos. Sempre que o protocolo ActivityPub for mencionado ao longo do texto, estará se referindo à comunicação do tipo servidor-servidor.

O ActivityPub foi aceito, em 2018, como especificação da W3C, resultado do trabalho colaborativo de desenvolvedores com experiência prévia em projetos com objetivos semelhantes, como o *OStatus* e a *Pump API* (W3C, 2018).

De forma simplificada, o funcionamento do ActivityPub pode ser comparado ao serviço de e-mail (PRODROMOU, 2024). Em um serviço de e-mail, cada domínio, como *gmail.com*, é gerenciado por um servidor que hospeda diversos usuários, como *usuario1@gmail.com*. Esses usuários podem enviar e receber mensagens utilizando protocolos padronizados — como o *Simple Mail Transfer Protocol* (SMTP) para envio e *Post Office Protocol version 3* (POP3) ou *Internet Message Access Protocol* (IMAP) para recebimento. Assim, um usuário de um servidor consegue se comunicar com outro usuário de um servidor diferente, pois ambos utilizam o mesmo protocolo, mesmo mantendo administrações e políticas locais de moderação distintas (PRODROMOU, 2024).

No protocolo ActivityPub, cada usuário possui uma caixa de entrada (*inbox*) e uma caixa de saída (*outbox*). A *inbox* é utilizada para o recebimento de mensagens enviadas por outros servidores, enquanto a *outbox* é usada para o envio de mensagens geradas pelo próprio usuário (W3C, 2018). A Figura 6 ilustra esse funcionamento.

Quando um servidor se comunica com outro no contexto do protocolo ActivityPub, são, na prática, as caixas *inbox* e *outbox* dos usuários que realizam a troca de mensagens. A Figura 7 ilustra esse conceito. Quando Alyssa envia uma mensagem para Ben, ela a envia para sua própria *outbox*, que, por sua vez, encaminha a mensagem à *inbox* de Ben (W3C, 2018).

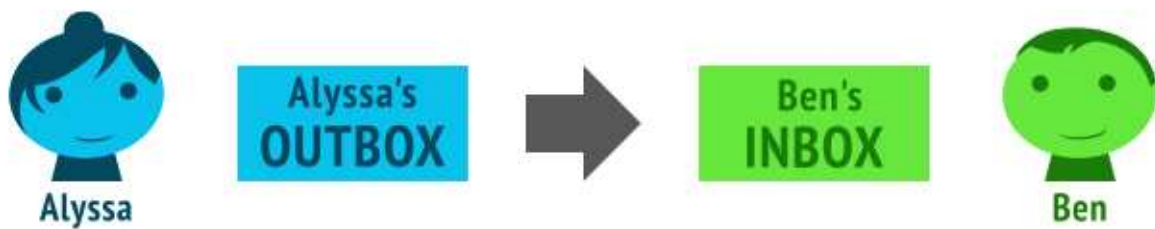
A documentação do protocolo ActivityPub recomenda a utilização do protocolo Hy-

**Figura 6 – Funcionamento básico da comunicação utilizando ActivityPub.**



Fonte: Adaptado de W3C (2018)

**Figura 7 – Funcionamento da comunicação entre usuários no ActivityPub.**



Fonte: W3C (2018)

*per*text Transfer Protocol (HTTP); no entanto, os endereços dos *endpoints* HTTP não fazem parte da especificação formal, ficando sob responsabilidade das implementações defini-los (PRODROMOU, 2024).

Nesse contexto, existe o *Actor* do tipo *Person*, como ilustrado na Figura 8, que detalha as informações associadas ao usuário, incluindo seus *endpoints* (W3C, 2018). Por meio desse objeto, é possível identificar, por exemplo, os *endpoints* da *inbox* e da *outbox* do usuário.

Como mecanismo de descoberta de atores (usuários) em uma implementação do protocolo ActivityPub, é recomendada a utilização da especificação *WebFinger* (PRODROMOU, 2024). O *WebFinger* é um padrão para descoberta de informações associadas a identidades na web, padronizado pela *Internet Engineering Task Force* (IETF) como a RFC 7033: *WebFinger Specification* (RFC 7033), publicada em 2013.

**Figura 8 – Exemplo das informações do usuário no ActivityPub.**

```

1 {
2   "@context": "https://www.w3.org/ns/activitystreams",
3   "type": "Person",
4   "id": "https://social.example/alyssa/",
5   "name": "Alyssa P. Hacker",
6   "preferredUsername": "alyssa",
7   "summary": "Lisp enthusiast hailing from MIT",
8   "inbox": "https://social.example/alyssa/inbox/",
9   "outbox": "https://social.example/alyssa/outbox/",
10  "followers": "https://social.example/alyssa/followers/",
11  "following": "https://social.example/alyssa/following/",
12  "liked": "https://social.example/alyssa/liked/"
13 }

```

**Fonte: W3C (2018)**

Para implementar o *WebFinger*, utiliza-se o *endpoint* HTTP `/well-known/webfinger`, acessado por meio do método HTTP *GET*. Com o uso do parâmetro de consulta *resource*, é possível realizar a busca por um recurso específico. Por exemplo, para consultar os dados representados na Figura 8, poderia ser realizada uma requisição *GET* na seguinte URL: `https://social.example/.well-known/webfinger?resource=alyssa@social.example`.

Segundo W3C (2018), o protocolo ActivityPub define diversas atividades relacionadas a interações típicas de redes sociais, tais como:

- *Create Activity*: cria um objeto;
- *Update Activity*: atualiza um objeto;
- *Delete Activity*: remove um objeto;
- *Follow Activity*: permite seguir um *Actor*, um *Object* ou outro recurso;
- *Accept Activity*: aceita um *Object*. A especificação não define exatamente o que está sendo aceito; pode ser, por exemplo, uma solicitação de amizade ou uma postagem;
- *Reject Activity*: rejeita um *Object*. Assim como no caso do *Accept*, o conteúdo rejeitado pode variar conforme a implementação;
- *Add Activity*: adiciona um *Object* a uma coleção de destino;

- *Remove Activity*: remove um *Object* de uma coleção;
- *Like Activity*: sinaliza que um ator demonstrou apreciação por um *Object*;
- *Announce Activity*: compartilha ou republica um *Object*, funcionando de forma semelhante a uma postagem;
- *Undo Activity*: desfaz uma atividade previamente realizada.

Como descrito por Prodromou (2024), o ActivityPub, por ser baseado no *ActivityStreams*, permite a criação de novos recursos, como atores, objetos e links. Isso torna o protocolo flexível e adaptável a diferentes tipos de aplicações sociais, além de aberto a melhorias por meio das *Fediverse Enhancement Proposals* (FEPs).

As FEPs são propostas de melhoria para o *fediverso*, com o objetivo de documentar, padronizar e evoluir funcionalidades relacionadas ao protocolo ActivityPub e suas extensões. Inspirado em modelos colaborativos como o das *Request for Comments* (RFC) da IETF, o processo de desenvolvimento das FEPs é aberto, versionado e conduzido pela própria comunidade. As propostas são organizadas em um repositório público, onde podem ser discutidas, avaliadas e referenciadas por desenvolvedores e pesquisadores interessados na interoperabilidade de softwares federados (Fediverse Enhancement Proposals, 2024a).

## 2.5 MODERAÇÃO DE CONTEÚDO NO FEDIVERSO

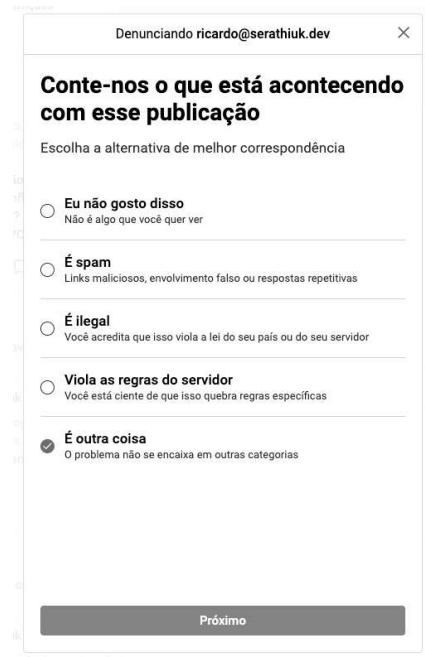
Como já citado anteriormente, existem várias implementações do ActivityPub, sendo a mais popular o Mastodon (ZHANG *et al.*, 2024). O Mastodon possui uma API própria para moderação (MASTODON, 2025a) a nível de postagem, porém pode, por escolha do usuário, notificar uma decisão de moderação para a instância onde está o usuário que está sendo moderado, utilizando a atividade *Flag* do ActivityStreams (MASTODON, 2025b).

Um usuário do Mastodon, para iniciar a moderação de conteúdo, pode utilizar funcionalidades implementadas pela rede social como mostrado na Figura 9. Nela, é possível visualizar no bloco 1 a postagem a ser moderada. Nos blocos 2 e 4 são mostradas opções de moderação como silenciar, filtrar, bloquear a comunicação com o usuário que fez a postagem ou até mesmo toda a instância do usuário autor da postagem. Essas decisões afetam apenas o usuário que as executa, não influenciando outros usuários ou instâncias. Caso o usuário selecione a opção de Denunciar, presente no bloco 3 da Figura 9, é exibida a tela mostrada na Figura 10.

**Figura 9 – Ações de moderação para um usuário no Mastodon.**



**Figura 10 – Denúncia pelo usuário no Mastodon.**



**Fonte: Autoria Própria.**

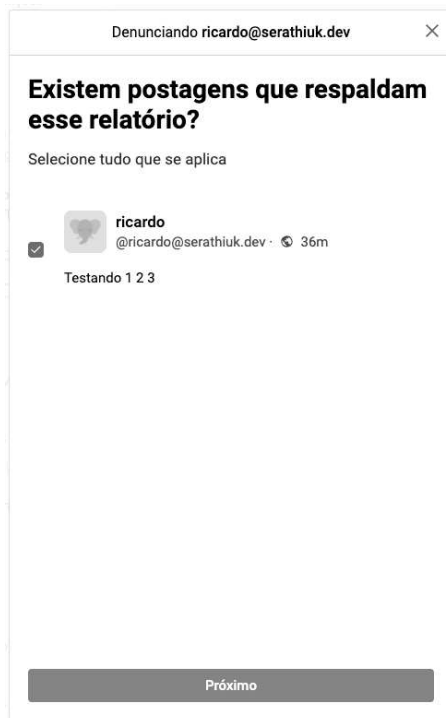
A seguir, caso sejam confirmadas as opções na tela da Figura 10, é exibida a tela da Figura 11 para que o usuário possa confirmar as postagens que estão sendo moderadas. Caso o usuário confirme, é exibida a tela presente na Figura 12. Nesta, o usuário pode, como mostrado no bloco 1, detalhar sua denúncia, e no bloco 2 escolher se a denúncia será enviada apenas para sua instância, o comportamento padrão, ou também para a instância do usuário que está sendo moderado.

Com a confirmação da denúncia, como mostrada na 13, ela é enviada para a instância do usuário e do usuário da postagem, caso a opção de encaminhamento tenha sido selecionada. Os moderadores das instâncias irão receber uma denúncia conforme exibida na Figura 14.

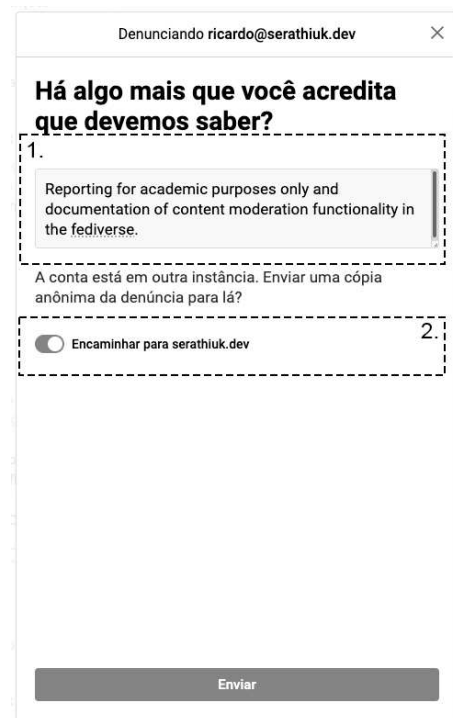
O moderador, conforme exibido na Figura 15, poderá executar diversas ações, como “Marcar como Resolvido”, que significa não tomar nenhuma ação punitiva, excluir a publicação de sua instância (ela é mantida nas outras instâncias), silenciar o usuário ou suspender o usuário impedindo sua comunicação com a instância. Também estão disponíveis opções personalizadas, como enviar apenas uma mensagem para o usuário, congelar a conta, marcar como sensível ou excluir a conta e todo seu conteúdo da instância.

Outro modelo de moderação amplamente utilizado no fediverso é a moderação por bloqueio ou limitação de domínio (CAVA *et al.*, 2024; MELDER *et al.*, 2025; GEHL, 2025). Nele, um moderador bloqueia ou limita a comunicação com outra instância. Na Figura 16 é

**Figura 11 – Seleção de postagens a serem denunciadas no Mastodon.**



**Figura 12 – Detalhamento da denúncia do usuário no Mastodon.**



**Fonte: Autoria Própria.**

exibida a tela do Mastodon onde essa moderação é realizada. O moderador informa o domínio, o tipo de moderação (Limitação ou Banimento), algumas configurações específicas da moderação e comentários públicos e privados sobre a ação. Na prática, listas de bloqueio compartilhadas entre instâncias emergiram como ferramentas de governança comunitária, com iniciativas como *The Bad Space*, *Oliphant's Lists* e *IFTAS CARIAD*, que oferecem bloqueio baseado em consenso entre administradores, com limiares configuráveis e procedimentos formais de apelação (GEHL, 2025).

## 2.6 INTELIGÊNCIA COLETIVA E MODERAÇÃO DE CONTEÚDO COLABORATIVA

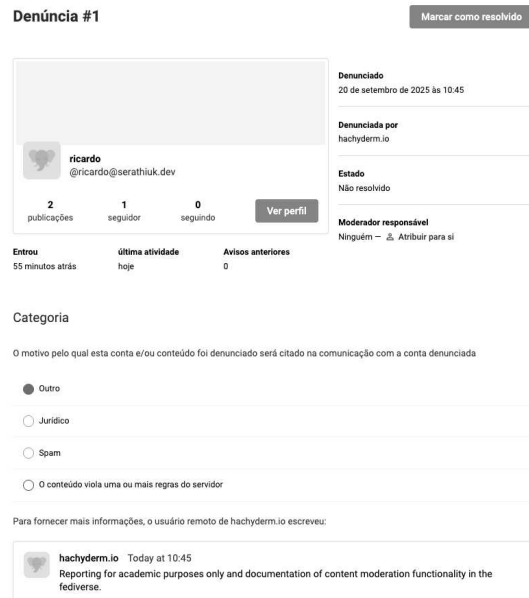
Inteligência coletiva pode ser definida como uma forma de inteligência universal e distribuída, que emerge da colaboração e competição entre muitos indivíduos, sendo capaz de permitir que um grupo execute uma variedade de tarefas (SALMINEN, 2012).

Atlee (2008) descreve diferentes escalas da inteligência coletiva. Ela pode se manifestar desde o nível individual ou de pequenos grupos até escalas maiores, como um país, uma sociedade ou mesmo a humanidade. O autor também classifica diferentes formas de inteligência coletiva, como aquelas baseadas em comunicação, aprendizado e estrutura organizacional.

**Figura 13 – Ações disponíveis ao usuário após a denúncia no Mastodon.**



**Figura 14 – Denúncia recebida pelo moderador no Mastodon.**



**Fonte: Autoria Própria.**

Vianna *et al.* (2019) definem o *crowdsourcing* como uma abordagem que permite a organizações obter ideias, soluções ou tarefas realizadas por um grande número de pessoas, geralmente por meio da internet. Essa prática amplia o modelo tradicional de terceirização ao envolver uma rede aberta de colaboradores externos, denominados *crowdsources*. As atividades podem variar desde microtarefas simples até desafios criativos ou complexos, sendo os participantes motivados por fatores diversos, como recompensas financeiras, reconhecimento, interesse pessoal ou o prazer de contribuir. Plataformas de *crowdsourcing* facilitam essa interação, conectando organizações a uma comunidade diversificada e global.

Algumas técnicas de moderação de conteúdo utilizam o *crowdsourcing* como abordagem para coletar e gerar conhecimento de forma colaborativa. De acordo com Pröllochs (2022), o *Birdwatch* foi uma dessas iniciativas. Desenvolvido pelo *Twitter* para combater a desinformação, o projeto permitia que usuários marcassem publicações enganosas e adicionassem notas explicativas. O funcionamento era colaborativo, incentivando a comunidade a avaliar e contextualizar informações, com o objetivo de promover maior credibilidade. A iniciativa reflete uma tendência crescente entre redes sociais de envolver os próprios usuários no processo de verificação de fatos, utilizando a inteligência coletiva para reduzir a desinformação e aumentar a confiança no conteúdo online.

Chuai *et al.* (2024) analisaram dados de todas as publicações verificadas desde o lançamento do recurso, em 2021. Os resultados indicam que o *Community Notes* (novo nome

**Figura 15 – Ações de moderação disponíveis ao moderador no Mastodon.**

Domínios de e-mail bloqueados

Conteúdo ofensivo será citado em comunicação com a conta denunciada — + Adicionar mais à denúncia

☐ ☐ X Remover da denúncia

☐ Testando 1 2 3

Web · 20 de setembro de 2025 às 10:05 · Público · Ver publicamente

Decida qual ação tomar para responder a essa denúncia. Se você tomar uma ação punitiva contra a conta denunciada, uma notificação por e-mail será enviada ao usuário, exceto quando a categoria **Spam** for selecionada.

|                        |                                                                                                                                                                                                           |
|------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Marcar como resolvi... | Nenhuma ação será tomada contra a conta denunciada, nenhuma violação será guardada e a denúncia será encerrada.                                                                                           |
| Excluir publicações    | As publicações denunciadas serão apagadas e um aviso de violação será mantido para te informar sobre o agravamento caso essa mesma conta cometa infrações no futuro.                                      |
| Silenciar              | A conta ficará visível apenas para aqueles que já a seguem ou que a procuram manualmente, limitando severamente seu alcance. Pode ser revertido a qualquer momento. Fecha todas as denúncias desta conta. |
| Suspender              | A conta e todo o seu conteúdo ficará inacessível e, eventualmente, excluído e interagir com ela será impossível. Reversível dentro de 30 dias. Encerra todas as denúncias contra esta conta.              |
| Personalizar           | Veja mais opções para controlar o comportamento da conta e personalizar a comunicação com a conta denunciada.                                                                                             |

Notas

Visualize e deixe anotações para outros moderadores e para você mesmo no futuro

Descreva quais ações foram tomadas ou quaisquer outras atualizações relacionadas...

Resolver com nota Adicionar nota

**Fonte: Autoria Própria.**

do *Birdwatch*) contribuiu para uma redução no engajamento — como curtidas e *retweets* — em publicações identificadas como enganosas. Isso sugere que a sinalização de imprecisões pode desencorajar a interação dos usuários com esse tipo de conteúdo. A pesquisa reforça a relevância de iniciativas de verificação de fatos conduzidas pela comunidade no combate à desinformação, oferecendo evidências empíricas da eficácia de ferramentas colaborativas nesse contexto.

Recentemente, outras redes sociais centralizadas também adotaram esse modelo de moderação colaborativa. A Meta incorporou o *Community Notes* em suas plataformas, como o *Instagram*, *Facebook* e *Threads* (META, 2025). O *TikTok* está testando uma funcionalidade semelhante, denominada *FootNotes*, inspirada no *Community Notes* (TIKTOK, 2025).

Um exemplo concreto de moderação colaborativa no fediverso é o uso da *hashtag FediBlock*. Por meio dela, usuários publicam sugestões de instâncias a serem bloqueadas, fun-

**Figura 16 – Moderação de domínio no Mastodon.**

## Novo bloqueio de domínio

**Domínio \***  
  

O bloqueio de domínio não vai prevenir a criação de entradas de contas na base de dados, mas vai retroativamente e automaticamente aplicar métodos específicos de moderação nessas contas.

**Gravidade**  

☒ Limitar  
☐ Banir  
☐ Nenhum

Quando selecionado, o domínio será bloqueado para a criação de mídia e dados de perfil da conta. Use **Nenhum** se você só quer rejeitar arquivos de mídia.

☐ **Rejeitar arquivos de mídia**  
 Remove arquivos de mídia armazenados localmente e recusa fazer download de qualquer um no futuro. Irrelevante para suspensões

☐ **Rejeitar denúncias**  
 Ignora todas as denúncias vindo deste domínio. Irrelevante para suspensões

☐ **Ofuscar nome de domínio**  
 Ofuscar parcialmente o domínio na lista se a exibição da lista de domínios limitados estiver habilitada

**Comentário privado**  
  
 Comente sobre essa restrição ao domínio para uso interno dos moderadores.

**Comentário público**  
  
 Comente sobre essa restrição ao domínio para o público geral, caso a divulgação da lista de bloqueio esteja ativada.

Criar bloqueio

**Fonte: Autoria Própria.**

cionando como um mecanismo informal de *crowdsourcing* de moderação. Administradores acompanham a *hashtag* para identificar instâncias problemáticas reportadas pela comunidade e decidem manualmente se aplicam os bloqueios sugeridos em suas próprias instâncias (MELDER *et al.*, 2025). Embora demonstre a disposição da comunidade do fediverso em colaborar na moderação, esse modelo depende inteiramente de esforço manual e não possui padronização ou infraestrutura técnica de suporte.

## 2.7 SISTEMAS PEER-TO-PEER (P2P)

Steen e Tanenbaum (2023) descrevem sistemas P2P como um conjunto de processos que operam de forma colaborativa e equivalente. Toda comunicação entre esses processos é simétrica, o que significa que não há hierarquia ou divisão de responsabilidades. Cada processo

atua simultaneamente como cliente e servidor. Os sistemas P2P podem ser classificados em estruturados ou não estruturados (STEEN; TANENBAUM, 2023).

Nos sistemas P2P estruturados, os nós (ou servidores) são organizados em uma topologia determinística, como um anel, uma árvore binária ou uma grade (*grid*). Este trabalho tem como foco esse tipo de sistema estruturado.

Sistemas P2P não estruturados, por outro lado, utilizam listas aleatórias de nós para estabelecer comunicação, sem uma organização pré-definida (STEEN; TANENBAUM, 2023). Quando um nó entra na rede, ele normalmente copia a lista de nós de um outro nó já existente. Para manter ou expandir essa lista, novas cópias podem ser obtidas de diferentes nós. Devido à ausência de uma topologia fixa para busca de informações, sistemas não estruturados costumam utilizar estratégias como *flooding*, em que o nó encaminha a requisição para todos os seus vizinhos, ou *random walks*, em que a requisição é enviada aleatoriamente a um único vizinho. Em ambas as abordagens, se o nó vizinho não possui a informação, ele repete o processo com seus próprios vizinhos.

Segundo Coulouris *et al.* (2013), sistemas P2P são projetados para disponibilizar serviços e aplicações distribuídas, aproveitando a capacidade computacional de computadores pessoais e estações de trabalho. Eles se baseiam em algoritmos descentralizados de distribuição e recuperação de informações entre computadores conectados por uma rede, seja ela corporativa ou a própria internet. Esses sistemas permitem que todos os usuários contribuam com recursos, sem distinção de responsabilidades entre os nós (COULOURIS *et al.*, 2013).

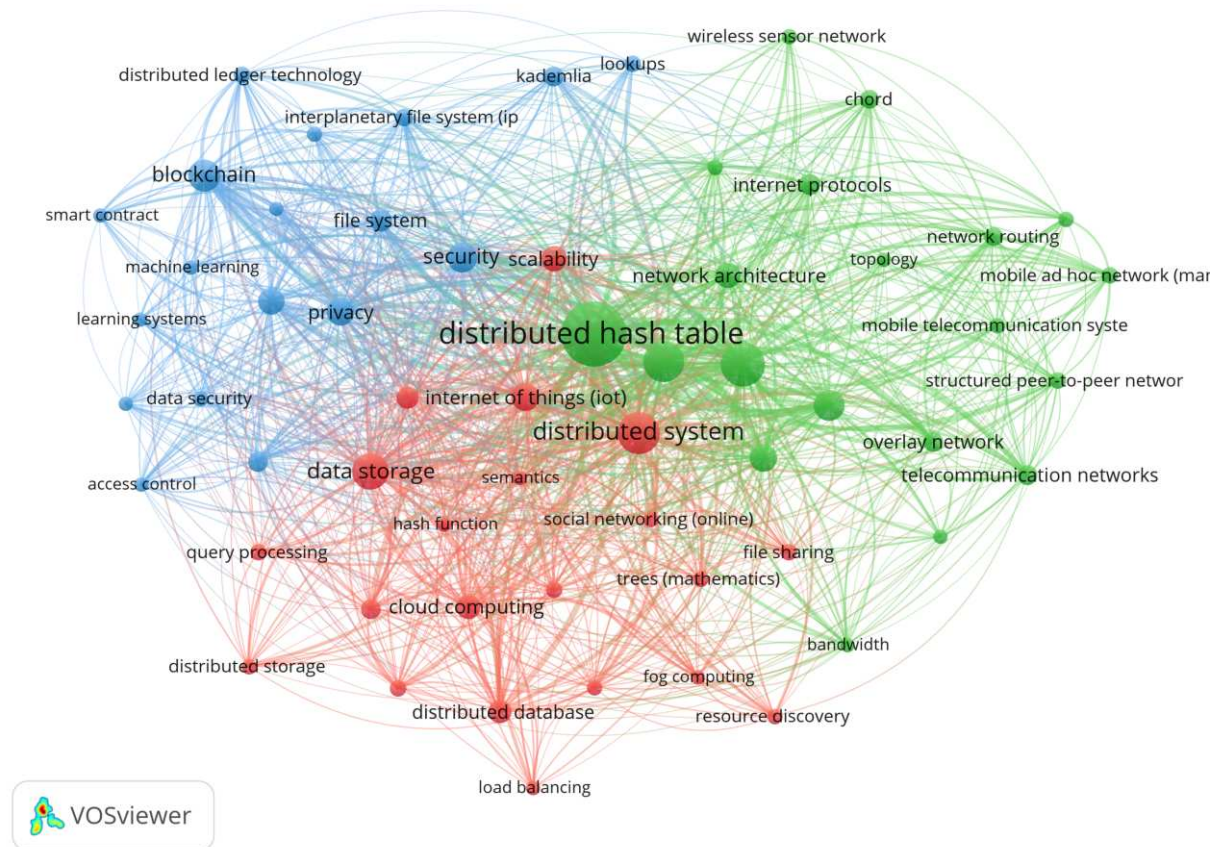
É possível identificar três gerações de serviços e aplicações P2P. A primeira geração é representada por aplicações de compartilhamento de músicas, como o *Napster*. A segunda geração inclui aplicações de compartilhamento de arquivos, como o *Gnutella*, *Kazaa* e *BitTorrent*. Já a terceira geração consiste em *middlewares* P2P, *Pastry* (ROWSTRON; DRUSCHEL, 2001), *Chord* (STOICA *et al.*, 2001) e *Kademlia* (MAYMOUNKOV; MAZIERES, 2002). Esses *middlewares* são baseados em estruturas de DHT.

Além desses *middlewares* descritos acima, outras aplicações populares baseadas em DHT incluem o *Mainline DHT*, que serve como base para a rede *BitTorrent* (LOEWENSTERN, 2008), o protocolo *Tor* (Tor Project, 2024), utilizado para navegação anônima; plataformas baseadas em *blockchain*, como o *Ethereum* (ETHEREUM, 2020); e sistemas de distribuição de arquivos voltados à *Web3*, como o *InterPlanetary File System* (IPFS) (IPFS, 2024). As DHTs também são utilizadas no desenvolvimento parcial ou completo de redes sociais P2P (MARX;

CHEONG, 2023; SALVE *et al.*, 2023; ZHANG *et al.*, 2023; MOUALKIA *et al.*, 2022; GUIDI *et al.*, 2022; GUIDI *et al.*, 2021).

Diante da variedade de *middlewares* baseados em DHT, buscou-se identificar quais permanecem relevantes na literatura recente. Em uma pesquisa na base de dados *Scopus* <<http://www.scopus.com>>, no período de 2015 até 2025, utilizando a palavra-chave “*distributed hash table*”, foram retornados 659 documentos. A visualização de densidade de palavras-chave, gerada com o *VOSviewer* <<https://www.vosviewer.com>> e apresentada na Figura 17, revela que apenas o *Kademlia* e o *Chord* aparecem com mais de 15 ocorrências. O *Chord* concentra-se em um *cluster* associado a comunicação e roteamento, enquanto o *Kademlia* integra um *cluster* relacionado a armazenamento, descentralização e *blockchain*.

**Figura 17 – Densidade de palavras-chave relacionadas com *distributed hash table* na base *Scopus*, no período de 2015 até 2025.**



Fonte: Informações da base Scopus e gerada via VOSViewer.

## 2.8 TABELAS *HASH* DISTRIBUÍDAS E O KADEMLIA

DHTs podem ser descritas como tabelas *hash* distribuídas que não necessitam de um servidor central. Uma DHT define um espaço global de chaves e valores, distribuído entre os nós da rede, que operam com as mesmas funções e responsabilidades (ZHANG *et al.*, 2013). Cada unidade de informação é representada por uma tupla  $(K, V)$ , sendo  $K$  a chave e  $V$  o valor correspondente. Uma DHT oferece a operação  $put(K, V)$  para armazenar um valor e a operação  $get(K)$  para recuperar o valor associado à chave.

Cada nó possui seu próprio ID, que é fundamental para determinar como ele será posicionado na rede e quais chaves será responsável por armazenar (ZHANG *et al.*, 2013). Tanto no *Chord* (STEEN; TANENBAUM, 2023) quanto no *Kademlia* (MAYMOUNKOV; MAZIERES, 2002), o ID exerce papel central no funcionamento da rede.

O *Kademlia* foi inicialmente proposto por Maymounkov e Mazieres (2002) como um protocolo específico de DHT, que utiliza a operação *Exclusive OR* (Ou Exclusivo) (XOR) para calcular a distância entre os nós. No *Kademlia*, os identificadores de nós e chaves são definidos por um valor *hash* de 160 *bits*.

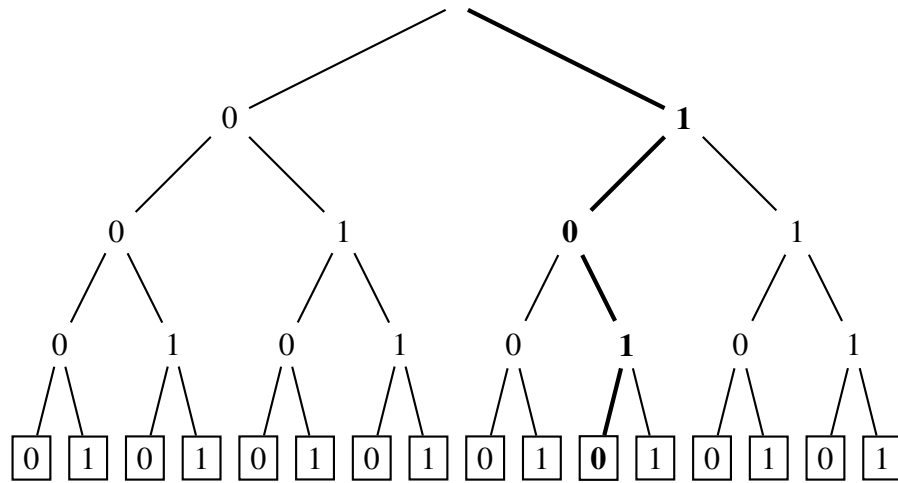
A Figura 18 apresenta uma representação simplificada de uma rede *Kademlia* de 16 nós (ou servidores). Os IDs desses nós possuem apenas 4 *bits* para facilitar a explicação do funcionamento do protocolo. Um exemplo de um ID de um nó é o **1010** que está destacado em negrito na Figura 18.

Esses IDs são gerados a partir de uma função de *hash*, como o algoritmo SHA-1, que por padrão gera valores de 160 *bits* (ZHANG *et al.*, 2013). A mesma estratégia é aplicada às chaves armazenadas na DHT, utilizando o mesmo algoritmo para gerar o *hash* correspondente à chave-valor (MAYMOUNKOV; MAZIERES, 2002).

Como também descrito por Maymounkov e Mazieres (2002), a proximidade entre os 2 *hashes* é calculada utilizando operação XOR entre os valores. A distância sempre será maior que zero para *hashes* diferentes. Já a distância será zero somente ao fazer a comparação utilizando operação XOR para o mesmo *hash*, o que significa que zero se trata do *hash* do mesmo nó.

O XOR como métrica faz com que a distância seja simétrica. Por exemplo, tanto com a operação  $1101 \oplus 0011$  quanto  $0011 \oplus 1101$ , a distância será 1110. Então, considerando  $d$  como a função que calcula a distância entre dois *hashes*, e que  $x$  e  $y$  são dois *hashes*, pode-se considerar que  $d(x, x) = 0$  e  $d(y, y) = 0$ . Também se verifica que  $d(x, y) = d(y, x)$  e  $d(x, y) > 0$  para  $x \neq y$ .

Figura 18 – Árvore binária de 4 níveis com rótulos binários



Fonte: Autoria própria.

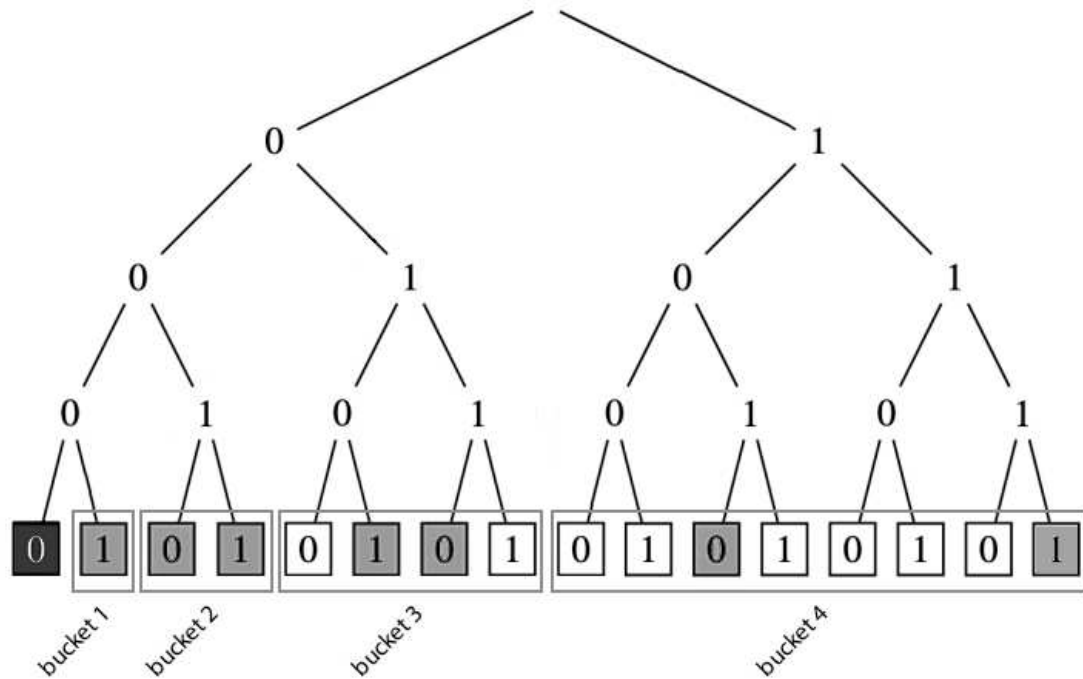
(MAYMOUNKOV; MAZIERES, 2002).

O entendimento dessa regra é importante para duas operações no *Kademlia* (MAYMOUNKOV; MAZIERES, 2002): encontrar em qual nó um par chave-valor será armazenado e construir a tabela de roteamento. Para localizar ou armazenar um novo par chave-valor, aplica-se primeiro a mesma função de *hash* utilizada para gerar o ID dos nós à chave do valor a ser armazenado. Com o *hash* resultante, busca-se o nó cujo *hash* seja o mais próximo, aplicando a função  $d(x,y)$  sobre os nós conhecidos até encontrar o menor valor retornado por  $d$ , que representa a menor distância.

Em uma rede *Kademlia* com milhares ou até milhões de nós, essa busca seria custosa, pois exigiria a verificação de todos os nós. Para otimizar esse processo, utiliza-se uma tabela de roteamento chamada *k-buckets*, que facilita a localização do nó com o identificador mais próximo, seja ele de outro nó ou de uma chave-valor (MAYMOUNKOV; MAZIERES, 2002).

O conjunto de *K-buckets* forma uma tabela de roteamento que pode ser representada conceitualmente como uma árvore binária, estruturada com base na distância entre os IDs dos nós da rede *Kademlia* e o ID do nó local, que é o nó que armazena a tabela de roteamento (ZHANG *et al.*, 2013). A Figura 19 exibe um exemplo de uma tabela de roteamento de um nó, considerando uma rede de 4 *bits*. O nó 0, apresentado em preto na figura, representa o nó local, ou seja, o nó no qual a tabela de roteamento *k-buckets* está armazenada. Embora a estrutura de dados do *Kademlia* não seja uma árvore, essa representação visual é útil para demonstrar a distância lógica entre os nós e, consequentemente, a forma como os *k-buckets* particionam a rede (MAYMOUNKOV; MAZIERES, 2002).

Figura 19 – Representação de uma tabela de roteamento *k-buckets* de um nó em rede *Kademlia* de 4 bits.



Fonte: Autoria própria.

A estrutura da tabela de roteamento é dividida em  $n$  listas chamadas de *k-buckets*, sendo  $n$  o número de *bits* da chave (MAYMOUNKOV; MAZIERES, 2002). Para uma chave de 4 *bits*, haverá 4 listas. Para uma rede de 160 *bits*, haverá 160 listas. As faixas de distância correspondentes a cada lista são ilustradas pelos círculos vermelhos na Figura 19. O nó 0000, conhece  $k$  nós de cada *bucket*, sendo  $k$  um número fixo definido pela implementação do protocolo.

Na Figura 19, são exibidos em cinza os nós de cada *bucket* cuja referência está armazenada na tabela de roteamento, considerando  $k = 2$ , que é a quantidade máxima de referências que um *bucket* pode armazenar. Os nós em branco são aqueles que não possuem referência registrada na tabela de roteamento do nó ou não existem na rede Kademlia. É possível observar que a quantidade possível de nós em um *k-bucket* cresce exponencialmente com a distância, enquanto a quantidade de referências armazenadas permanece constante, sempre limitada a  $k$  (MAYMOUNKOV; MAZIERES, 2002). Um *k-bucket*  $i$  pode referenciar nós cuja distância esteja no intervalo  $[2^i, 2^{i+1} - 1]$ .

O processo de busca no Kademlia é iterativo e paralelo. Quando um nó, conhecido como o iniciador, precisa localizar um ID de destino, ele primeiro seleciona em sua própria tabela um pequeno número dos contatos mais próximos do alvo que ele conhece, sendo este o

$\alpha$ , um parâmetro de concorrência geralmente igual a 3 (MAYMOUNKOV; MAZIERES, 2002). Em seguida, o iniciador envia requisições *FIND\_NODE* em paralelo a esses  $\alpha$  contatos. Cada nó consultado responde diretamente ao iniciador, informando sobre outros nós que ele conhece que estão ainda mais próximos do alvo. O iniciador coleta essas respostas, atualiza sua lista de candidatos e repete este ciclo, consultando o novo grupo dos  $\alpha$  contatos mais promissores (MAYMOUNKOV; MAZIERES, 2002). O processo continua em rodadas até que os  $k$  nós mais próximos do *ID* desejado sejam encontrados.

No caso de armazenamento ou busca de uma chave-valor, se o nó com o *hash* exato não existir, a chave-valor será armazenada ou recuperada no nó mais próximo ao *hash* calculado (MAYMOUNKOV; MAZIERES, 2002).

Todas as operações descritas são executadas utilizando o protocolo *Kademlia*, que define quatro operações *Remote Procedure Call* (RPC) (MAYMOUNKOV; MAZIERES, 2002):

- *FIND\_NODE* é utilizada para buscar um nó por sua ID. A operação recebe a ID de um nó como argumento e retorna como resposta uma lista contendo os  $k$  contatos mais próximos ao ID desejado que o nó consultado conhece;
- *PING* é utilizada para verificar se um nó está ativo. Um ID aleatório de 160 *bits* é enviado ao nó, que deve responder devolvendo o mesmo identificador, confirmando sua disponibilidade. Caso não haja resposta adequada, o nó é considerado inativo;
- *STORE* é utilizada para armazenar um par chave-valor em um nó. A informação armazenada deve expirar em 24 horas, sendo responsabilidade de quem publicou a chave-valor reenviá-la periodicamente por meio da operação *STORE*, caso deseje manter o valor na DHT. Essa prática evita que os nós da rede mantenham informações obsoletas;
- *FIND\_VALUE* é utilizada para buscar o valor associado a uma chave. Se a chave for encontrada no nó onde a operação foi iniciada, o valor correspondente é retornado. Caso contrário, a operação retorna os  $k$  contatos mais próximos da chave que o nó conhece, de forma análoga à operação *FIND\_NODE*.

## 2.9 TRABALHOS RELACIONADOS

Esta seção apresenta os trabalhos relacionados aos temas de moderação de conteúdo em redes sociais, sejam elas centralizadas ou descentralizadas, discutindo técnicas, descobertas e

implementações relevantes.

Zia *et al.* (2022) propõem abordagens para o compartilhamento de modelos parcialmente treinados com o objetivo de detectar toxicidade em redes sociais descentralizadas e auxiliar na moderação distribuída. No entanto, o estudo identifica desafios importantes, como a definição do conceito de “toxicidade”, que pode variar de acordo com a visão de cada administrador de instância. O trabalho se limita a instâncias que utilizam a rede Pleroma, uma rede de *microblogging* que implementa o ActivityPub, e sua API, e não de forma genérica pensando no ActivityPub e no fediverso como um todo. O trabalho não detalha como o modelo seria compartilhado nem como se daria a comunicação entre instâncias não-Pleroma, concentrando-se apenas na utilidade do modelo para rotular automaticamente conteúdos tóxicos para instâncias do Pleroma.

Anaobi *et al.* (2023) realizam uma análise aprofundada sobre o perfil de administração de conteúdo em redes sociais federadas. Os autores relatam que a maioria dos administradores de instâncias utilizam configurações padrões de moderação, e que demoram a tomar medidas de moderação, por estarem sobrecarregados com o número de *posts*. Eles também citam que muitas das instâncias possuem apenas um administrador. Entretanto, a escolha de apenas a plataforma *Pleroma* e em modelos específicos limita o escopo da proposta a uma fração da rede federada que implementa o protocolo ActivityPub.

Thiel *et al.* (2023) descrevem as dificuldades na moderação de conteúdo em redes sociais federadas baseadas em ActivityPub, com foco específico em casos de exploração infantil. São sugeridas abordagens para detecção e tratamento desses casos, com destaque para a integração do *Mastodon* com a abordagem centralizada baseada no Microsoft PhotoDNA, tecnologia que identifica imagens semelhantes com base em seus *hashes*. A tecnologia Microsoft PhotoDNA é utilizada para detectar imagens que contenham conteúdos relacionados à exploração infantil. Porém, essa análise teria que ser feita individualmente para cada instância, o que pode reduzir a eficiência da moderação, pois cada instância faria as mesmas requisições para a mesma imagem. O trabalho propõe uma extensão para o ActivityPub para resolver esse problema. Nessa extensão, o servidor que originou o *post* original ficaria a cargo de fazer a análise da imagem com o Microsoft PhotoDNA e adicionaria uma assinatura do *hash* da imagem junto com a mensagem do ActivityPub, atestando que ela já foi verificada.

Jhaver *et al.* (2018) analisam o uso de listas de bloqueio no *Twitter* como estratégia para combater o assédio online. O estudo explora como essas listas funcionam, por que os usuários as criam e quais são seus pontos fortes e limitações. Os autores destacam a importância de

compreender as experiências dos usuários para o desenvolvimento de estratégias de moderação mais eficazes. O trabalho destaca a importância de focar na moderação em grupos vulneráveis a assédios online, com o desenvolvimento de ferramentas e sistemas que atendam às necessidades destes grupos. O trabalho também sugere mecanismos híbridos de moderação de conteúdo, como aqueles que combinam algoritmos com curadoria social, bem como meios de tornar as listas de bloqueios mais transparentes, deixando claro o motivo de cada bloqueio e oferecendo aos bloqueados a possibilidade de contestarem, caso achem o bloqueio injusto.

Morrow *et al.* (2022) revisam a ciência emergente da rotulagem de conteúdo em plataformas de mídia social, propondo uma taxonomia dos tipos de rótulos e discutindo seus efeitos sobre a percepção e o comportamento dos usuários. São discutidos os tipos de rótulos, que podem ser de veracidade, para ajudar na checagem de fatos, ou de contexto, para explicar melhor algum conteúdo. O estudo reforça que a rotulagem de conteúdo é uma ferramenta promissora para moderação eficaz em ambientes digitais. Os autores apresentam o *crowdsourcing* como uma alternativa à rotulagem utilizando Inteligência Artificial (IA) ou totalmente manual. É citado o Birdwatch, do Twitter, como uma ferramenta utilizada para rotulagem via *crowdsourcing* com intuito de atestar a veracidade das publicações. Porém, deixam claro que esse tipo de abordagem deve ser planejada pensando em possíveis usos coordenados via *bots* ou *trolls*, com intuito de distorcer a rotulagem.

Cava *et al.* (2024) examinam como a arquitetura descentralizada de redes sociais como o *Mastodon* pode favorecer a formação de comunidades polarizadas. Para isso, os autores modelam as relações de seguimento entre instâncias como arestas positivas e os bloqueios de domínio como arestas negativas em um grafo assinado, aplicando técnicas de análise de redes para identificar grupos polarizados no nível de instância. A ausência de uma moderação centralizada facilita a criação de grupos ideologicamente homogêneos, promovendo *echo chambers*. Embora a descentralização promova autonomia e diversidade, os autores sugerem a adoção de estratégias que incentivem o contato com perspectivas diversas para mitigar os efeitos da polarização.

Bono *et al.* (2024) analisam a moderação descentralizada no *Mastodon*, com foco no uso de listas de bloqueios, que permitem a administradores restringirem ou bloquearem interações com outras instâncias. O estudo destaca que, embora eficaz para proteger comunidades, o uso indiscriminado dessas listas ou a importação automática de listas públicas pode levar ao isolamento excessivo de instâncias legítimas e à fragmentação da federação. Os autores ressaltam a importância de práticas de moderação criteriosas e transparentes.

Badis *et al.* (2021) apresentam o P2PCF, um framework baseado em filtragem colaborativa para recomendação de conteúdo em redes P2P, com preservação da privacidade dos usuários. Cada peer armazena uma matriz local de ratings e consulta seus contatos sociais diretamente para gerar recomendações, sem necessidade de um servidor central. O trabalho também aborda o problema do cold start, relacionado à recomendação para novos usuários ou novos conteúdos.

Guidi *et al.* (2022) desenvolveram o DISCO, uma ferramenta para detecção de comunidades em DOSN, de forma distribuída e auto-organizada. Os autores descrevem redes sociais como grafos de amizade, nos quais os nós representam pessoas e as conexões simbolizam interações e destacam que em redes descentralizadas, as técnicas de análise devem considerar sua natureza dinâmica e distribuída, propondo protocolos descentralizados para gerenciar a rede e extrair conhecimento em tempo real. Os autores organizaram os nós da rede em uma DHT, utilizando o protocolo Pastry DHT, para possibilitar a descoberta e a comunicação descentralizadas. A proposta utiliza uma arquitetura de três camadas, sendo a camada superior o usuário (um nó da rede). A camada intermediária é responsável pelos serviços de gerenciamento, *lookup*, entre outros, e a camada inferior é a de comunicação, responsável pela comunicação dos nós na DOSN. Quando o usuário/nó está *offline*, a comunicação é delegada para um *super-peer*, que é chamado pelos autores de *moderator*.

Friedl e Morgan (2024) analisam infraestruturas de moderação de conteúdo descentralizadas, nas quais a autoridade e a responsabilidade são distribuídas entre múltiplos atores ou instituições. Exemplos como *Reddit* e *Mastodon* são utilizados para ilustrar formas de moderação baseadas em comunidades locais. Apesar de promoverem governança participativa, esses modelos enfrentam desafios como a fragmentação discursiva, a criação de *echo chambers* e a sobrecarga de moderadores voluntários. Os autores destacam a necessidade de ferramentas e estruturas adequadas para apoiar a moderação descentralizada e sugerem que políticas públicas considerem essas particularidades ao regular plataformas. Com relação ao *fediverso*, o artigo destaca que cada instância é livre para definir suas próprias regras locais e pode optar por bloquear outras instâncias, tornando seu conteúdo invisível para os próprios usuários. No entanto, não há uma autoridade central com poder para banir ou impor políticas globais entre instâncias.

Ermoshina e Musiani (2025) analisam como a arquitetura e a infraestrutura de plataformas de mídia social federadas influenciam a moderação de conteúdo. O trabalho argumenta que a federação pode criar novas formas de governança para a moderação, combinando organização comunitária, disseminação de informações e ferramentas para combater o assédio, o discurso de

ódio e a desinformação. No entanto, o artigo também explora os desafios e as armadilhas potenciais desse modelo, questionando se é possível criar espaços seguros desde a sua concepção e qual o papel dos moderadores humanos nesses ambientes descentralizados. O artigo também explica como os administradores do Mastodon tentam otimizar a moderação, por exemplo, através da desfederação de instâncias (bloqueio em massa de forma coordenada), e aponta que a moderação automatizada, utilizando *machine-learning* ou de outra natureza, é vista com ceticismo pela comunidade do Fediverso, ao mesmo tempo que os administradores se sentem sobrecarregados com a tarefa de moderar suas instâncias.

Melder *et al.* (2025) investigam as tensões entre moderação comunitária e ajuda mútua no fediverso, com foco no uso de listas de bloqueio compartilhadas entre instâncias como mecanismo emergente de proteção coletiva. Os autores identificam que as ferramentas nativas de moderação do protocolo ActivityPub, limitadas a duas ações binárias aplicadas no nível de instância (silenciar ou desfederar), são insuficientes para lidar com a complexidade do ambiente federado, resultando em efeitos colaterais como falsos positivos propagados em cascata e perda irreversível de conexões entre comunidades. O trabalho destaca ainda que listas de bloqueio baseadas em consenso entre instâncias podem distribuir o trabalho de identificação de conteúdo indesejado entre múltiplos moderadores, reduzindo a sobrecarga individual, e propõe que instâncias possam configurar limiares de consenso para determinar quais bloqueios adotar. No entanto, o estudo não propõe mecanismos técnicos concretos para a distribuição dessas listas nem para o estabelecimento desse consenso, concentrando-se na análise qualitativa das tensões sociais decorrentes do modelo de moderação atual do fediverso.

Gehl (2025) investigou as práticas de moderação de conteúdo no fediverso, com foco no uso de listas de bloqueio compartilhadas entre instâncias. O autor documenta três problemas fundamentais dessas listas: a ausência de critérios de seleção embutidos, sua natureza binária e a tendência à proliferação sem controle. O trabalho também documenta iniciativas comunitárias que buscam mitigar esses problemas, como o *The Bad Space*, o *Oliphant's Lists* e o *IFTAS CARIAD*, que adotam sistemas de consenso entre administradores e limiares configuráveis pelo importador. No entanto, essas soluções dependem de infraestrutura externa ao protocolo ActivityPub e de esforço manual, sem padronização técnica.

Para contextualizar esta pesquisa, o Quadro 1 resume as principais características dos trabalhos relacionados discutidos anteriormente. O quadro distingue entre os estudos de natureza teórica e aqueles que propõem abordagens práticas. Os trabalhos teóricos são aqueles que

apresentam ou discutem problemas relacionados ao fediverso ou às *DOSN*.

**Quadro 1 – Características dos trabalhos relacionados**

| Trabalho                    | Teórico                                                                                                 | Prático                                                                                     |                                                                                          |
|-----------------------------|---------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------|
|                             | Problemas Apontados                                                                                     | Moderação de Conteúdo                                                                       | Protocolos, técnicas e ferramentas                                                       |
| Anaobi <i>et al.</i> (2023) | Sobrecarga dos administradores de instâncias.                                                           | Modelo para auxiliar na moderação de conteúdo de uma única instância e restrito ao Pleroma. | API do Pleroma.                                                                          |
| Badis <i>et al.</i> (2021)  | Distribuição de recomendação de conteúdo em arquiteturas <i>P2P</i> .                                   | Genérico e não focado em moderação de conteúdo.                                             | Filtragem colaborativa <i>P2P</i> .                                                      |
| Bono <i>et al.</i> (2024)   | Isolamento de instância.                                                                                | Uso de listas de bloqueios.                                                                 | API do Mastodon                                                                          |
| Cava <i>et al.</i> (2024)   | Polarização entre instâncias.                                                                           | -                                                                                           | API do Mastodon; análise de grafos assinados.                                            |
| Ermoshina e Musiani (2025)  | Dificuldades relacionadas à moderação de conteúdo de redes federadas.                                   | Desfederação de instâncias                                                                  | Mastodon                                                                                 |
| Friedl e Morgan (2024)      | Dificuldades na moderação de conteúdo em <i>DOSN</i> , câmaras de eco e falta de ferramentas adequadas. | -                                                                                           | -                                                                                        |
| Gehl (2025)                 | Listas de bloqueio não carregam critérios de seleção, são binárias e tendem a proliferar sem controle.  | Listas de bloqueio compartilhadas com consenso entre administradores.                       | Mastodon, <i>The Bad Space</i> , <i>Oliphant's Lists</i> , <i>IFTAS</i> <i>CA-RIAD</i> . |
| Guidi <i>et al.</i> (2022)  | Dificuldade de detecção de comunidades de forma descentralizada em <i>DOSN</i> .                        | Detecção de comunidades para auxiliar na moderação, separando usuários em grupos.           | Pastry DHT.                                                                              |
| Jhaver <i>et al.</i> (2018) | Uso de listas de bloqueio contra assédio online.                                                        | Discussão sobre possíveis melhorias em mecanismos de listas de bloqueio.                    | Twitter                                                                                  |
| Melder <i>et al.</i> (2025) | Ferramentas de moderação binárias geram falsos positivos e sobrecarga dos moderadores.                  | Listas de bloqueio baseadas em consenso entre instâncias.                                   | ActivityPub, Mastodon.                                                                   |
| Morrow <i>et al.</i> (2022) | Rotulagem de conteúdo em plataformas de mídia social.                                                   | Rótulos de conteúdo                                                                         | Birdwatch, Twitter                                                                       |
| Thiel <i>et al.</i> (2023)  | Falta de ferramentas adequadas para combater a exploração infantil no fediverso.                        | Extensão do ActivityPub para detecção de abuso infantil.                                    | Mastodon, Microsoft PhotoDNA, ActivityPub                                                |
| Zia <i>et al.</i> (2022)    | O significado de conteúdo tóxico pode mudar de uma instância para outra.                                | Modelo parcialmente compartilhável e restrito ao Pleroma.                                   | API do Pleroma, <i>Google Jigsaw's Perspective API</i> .                                 |

**Fonte: Autoria própria.**

Com base na análise dos trabalhos relacionados, é possível analisar os desafios relacionados à moderação de conteúdo em *DOSN*. A sobrecarga dos administradores, a dificuldade em definir políticas de moderação consistentes entre instâncias, a tendência à formação de câmaras de eco e a fragmentação da rede são temas recorrentes que mostram as limitações dos

mecanismos atuais. Embora abordagens pontuais tenham sido propostas, muitas se restringem a plataformas específicas ou não abordam o problema de forma padronizada e interoperável, pensando no fediverso como um todo e no protocolo ActivityPub. Considerando o cenário apresentado, o próximo capítulo apresenta a arquitetura de moderação de conteúdo proposta neste trabalho como uma extensão ao protocolo ActivityPub.

### 3 D-COLMEIA: EXTENSÃO DE MODERAÇÃO DISTRIBUÍDA E COLABORATIVA PARA O *ACTIVITYPUB*

No Fediverso, não existe um ponto central responsável pelo armazenamento de todas as informações transmitidas entre os usuários. Cada instância é responsável por armazenar os dados enviados e recebidos por seus próprios usuários (ABBING *et al.*, 2023). Dessa forma, em um sistema que se propõe a criar um repositório colaborativo entre as instâncias, a própria estrutura federada pode representar um obstáculo. O fluxo de comunicação entre as instâncias pode ser elevado, e há o risco de as instâncias manterem estados desatualizados sobre a moderação, ou incerteza sobre onde as informações devem ser armazenadas. Como observam Friedl e Morgan (2024), a moderação descentralizada pode criar fragmentação discursiva e dificuldades de coordenação, especialmente quando múltiplos nós de governança possuem jurisdição sobreposta sobre o mesmo conteúdo, gerando incerteza sobre onde residem as responsabilidades de moderação. Melder *et al.* (2025), em pesquisa empírica com usuários e administradores do fediverso, identificaram que a ausência de mecanismos padronizados para distribuição e consenso de informações de moderação leva administradores a recorrer a listas de bloqueio distribuídas de forma improvisada entre instâncias.

Por exemplo, a instância que originou um *post* alvo de denúncia pode não ser um repositório confiável dessa informação, já que pode agir de forma parcial ou rejeitar avisos vindos de outras instâncias (FRIEDL; MORGAN, 2024; CAVA *et al.*, 2024). A criação de um servidor central para armazenar tais dados também seria problemática, pois representaria um ponto único de falha e concentraria o controle em uma única entidade, o que contraria os princípios do *fediverso* (ABBING *et al.*, 2023).

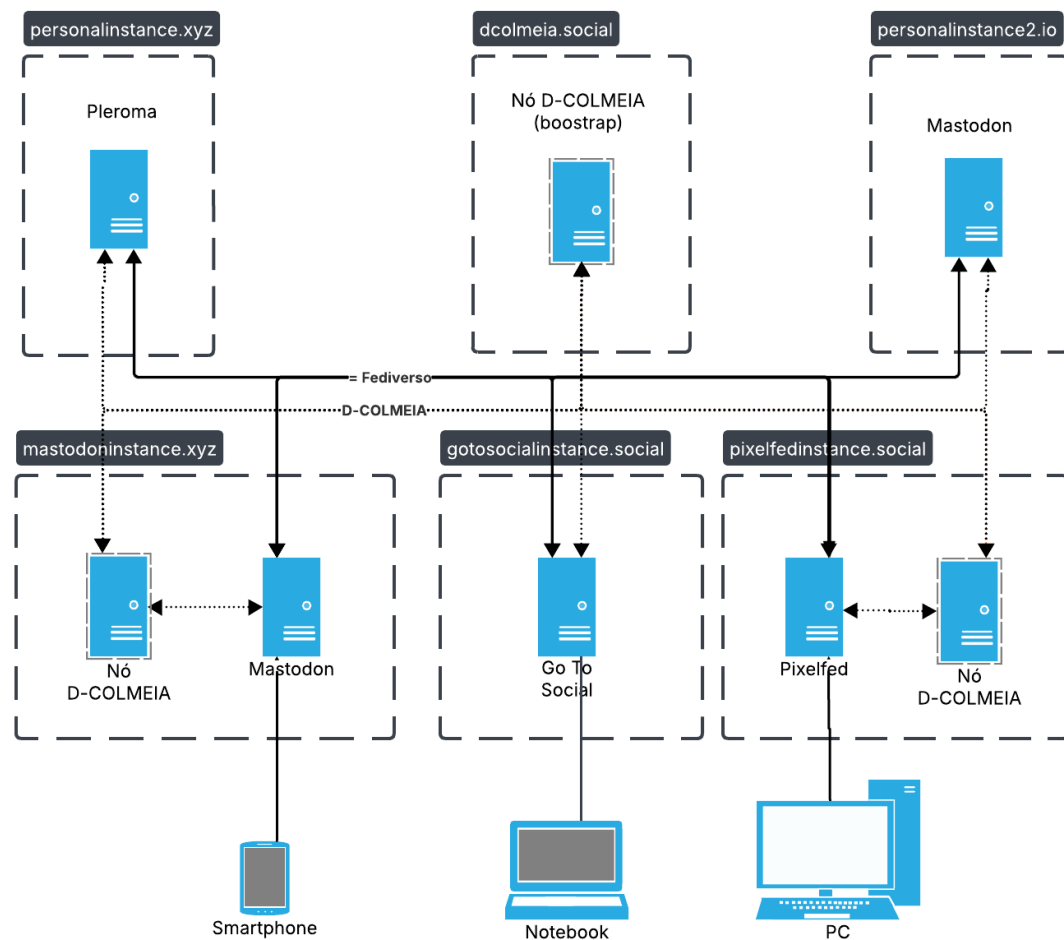
Uma abordagem para esse problema é a adoção de uma estrutura de rede baseada em DHT, com o objetivo de distribuir, armazenar e localizar informações relacionadas à moderação, além de viabilizar mecanismos de consenso entre instâncias sobre o conteúdo de moderação. Essa proposta, apresentada neste trabalho, é chamada de D-COLMEIA (*Distributed and Collaborative Moderation Extension for ActivityPub*). Através do D-COLMEIA, a informação passa a pertencer à rede, e não a uma instância ou usuário isolado.

Conforme apresentado na Seção 2.7, entre os *middlewares* de DHT com maior relevância na literatura recente, o *Kademlia* está associado ao armazenamento e recuperação de dados, enquanto o *Chord* se concentra em comunicação e roteamento. Como o objetivo da D-COLMEIA

envolve o armazenamento e a recuperação distribuída de dados de moderação, o *Kademlia* apresenta maior alinhamento com o problema tratado, sendo adotado como base para a arquitetura proposta.

A arquitetura do D-COLMEIA é apresentada na Figura 20, na qual tem-se uma rede Fediverso composta por cinco instâncias e a rede DHT de moderação composta por três nós DHT responsáveis pela moderação.

**Figura 20 – Arquitetura do D-COLMEIA**



**Fonte: Autoria própria**

Uma instância pode optar por criar um novo nó na DHT para colaborar e estabelecer seu ponto de comunicação com a rede de moderação, ou pode utilizar um nó mantido por terceiros, como outra instância ou organização. Também é possível que um nó seja hospedado de forma independente, sem vínculo direto com uma instância específica, apenas com a finalidade de compor a rede DHT.

### 3.1 METODOLOGIA DE DESENVOLVIMENTO

O desenvolvimento do D-COLMEIA seguiu um processo incremental e iterativo, conduzido ao longo da pesquisa de mestrado. A proposta foi sendo refinada de forma progressiva à medida que o aprofundamento teórico em protocolos de federação, tabelas *hash* distribuídas e modelagem de ontologias permitia revisar e amadurecer as decisões de modelagem e de arquitetura. A cada ciclo, novos conceitos estudados eram incorporados ao modelo, que era então reavaliado e ajustado.

Nesse processo, o protótipo funcional, apresentado no Capítulo 4, teve papel central. Mais do que uma etapa final de validação, ele atuou como instrumento de investigação. Hipóteses de projeto eram implementadas, confrontadas com as restrições reais do *ActivityPub* e do fediverso e, então, reincorporadas ao modelo conceitual. Diversas decisões de modelagem emergiram desse confronto entre a especificação teórica e a viabilidade prática de implementação.

Em consequência dessa abordagem, parte das decisões de modelagem apresentadas neste capítulo, como a estrutura do `dcolmeia:CommunityNoteGroup` e o reaproveitamento de propriedades do *ActivityStreams 2.0*, privilegia a aderência às práticas correntes da rede e a viabilidade de implementação, em vez de uma formalização estritamente ortodoxa do vocabulário. Essa orientação, centrada na construção e na avaliação de um artefato como meio de produção de conhecimento, é convergente com os princípios da *Design Science Research* (HEVNER *et al.*, 2004; PEFFERS *et al.*, 2007).

### 3.2 ONTOLOGIA DO D-COLMEIA

Para que instâncias independentes do fediverso possam colaborar na moderação de conteúdo, é necessário que os dados compartilhados possuam uma estrutura semântica comum. Com esse objetivo, foi criada uma ontologia que define o domínio da moderação de conteúdo no D-COLMEIA. As instâncias dessa ontologia, como `dcolmeia:CommunityNote` e `dcolmeia:CommunityNoteGroup`, são distribuídas entre os nós da rede por meio da DHT. Cada instância possui um identificador único (sua URI), cujo *hash* determina em qual nó da DHT ela será armazenada. Como o *Kademlia* e outros *middlewares* baseados em DHT distribuem os dados por meio do *hash* de um identificador, é possível distribuir de forma uniforme os dados de moderação entre os nós da rede descentralizada (ZHANG *et al.*, 2013).

O modelo de dados baseia-se e estende a ontologia *ActivityStreams 2.0* (prefixo *as*;

*namespace* <https://www.w3.org/ns/activitystreams>). As classes e propriedades específicas do D-COLMEIA são definidas em um *namespace* próprio, <https://dcolmeia.social/ns>. Tanto o vocabulário base quanto a extensão proposta são formalmente descritos com a OWL. O Quadro 2 lista as classes definidas pelo D-COLMEIA e como estas se relacionam com o *ActivityStreams 2.0*.

A partir do domínio definido no OWL é possível descrever as mensagens do D-COLMEIA utilizando JSON-LD. A escolha da OWL como linguagem de representação da ontologia é justificada por dois fatores principais. Primeiro, a OWL adota a *Open World Assumption* (OWA), segundo a qual a ausência de uma informação não implica sua falsidade (HITZLER *et al.*, 2012). Esta característica é particularmente adequada a um sistema distribuído como o D-COLMEIA, no qual nenhum nó detém uma visão global e completa da rede: o fato de uma nota de moderação não ser localizada em um determinado nó, ou ao final de uma busca, não autoriza concluir sua inexistência, mas apenas indica que o conhecimento daquele nó é parcial. Tal interpretação é coerente com a natureza de uma DHT, na qual a ausência de um valor pode decorrer da indisponibilidade temporária de nós ou da ainda não convergência das tabelas de roteamento, e não de uma negação efetiva. Segundo, a OWL é construída sobre o RDF, que por sua vez pode ser serializado em JSON-LD (W3C, 2020). Além de viabilizar a serialização, o modelo de grafos do RDF confere flexibilidade de esquema à ontologia: novas tipologias de moderação podem ser incorporadas e novas instâncias podem participar da rede sem invalidar o conhecimento previamente modelado. Essa compatibilidade é essencial, uma vez que o protocolo ActivityPub utiliza JSON-LD como formato de representação de suas mensagens, permitindo uma integração semântica direta entre a ontologia do D-COLMEIA e o protocolo base do fediverso.

A Figura 21 exemplifica um objeto do tipo `dcolmeia:ModerationNode`, que representa um nó de moderação. Este objeto descreve tanto informações padrão do *ActivityStreams 2.0*, como os endereços de *inbox* e *outbox*, quanto a lista de *buckets* do nó, definida na propriedade específica `dcolmeia:bucketItems`.

Já a Figura 22 descreve uma nota da comunidade, que é baseada em um artigo (`as:Article`) do *ActivityStreams 2.0*. São apresentadas informações relevantes ao *ActivityStreams 2.0*, como `as:name`, `as:content`, `as:published`, `as:attributedTo` e `as:inReplyTo`. Também é apresentada a propriedade `dcolmeia:category`, que dá contexto à nota da comunidade, a partir de categorias definidas na tipologia Neon (CHARLES

**Quadro 2 – Descrição das Classes da Ontologia do D-COLMEIA.**

| <b>Classe e Herança</b>                                                              | <b>Descrição</b>                                                                                                                                                                                                                                                        | <b>Propriedades</b>                                                                                                                                                                                                                                |
|--------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>CommunityNoteGroup</i><br>herda: <i>as:Group</i> ,<br><i>as:OrderedCollection</i> | Grupo de notas públicas <i>CommunityNote</i> sobre um mesmo recurso/URL. Por ser baseado em um ator ( <i>as:Group</i> ), possui <i>as:inbox</i> e <i>as:outbox</i> . Por herdar de <i>as:OrderedCollection</i> , possui <i>as:totalItems</i> e <i>as:orderedItems</i> . | <i>as:inbox</i><br><i>as:outbox</i><br><i>as:totalItems</i><br><i>as:orderedItems</i><br><i>hasCommunityNotes</i> ( $\rightarrow$ <i>CommunityNote</i> , 1:n)                                                                                      |
| <i>ModerationCategory</i><br>herda: <i>as:Object</i>                                 | Categoria de moderação. Utilizada para dar contexto padronizado ao <i>CommunityNote</i> , com tipologia inspirada na tipologia Neon (CHARLES <i>et al.</i> , 2022).                                                                                                     | <i>hasSubcategory</i> ( $\rightarrow$ <i>ModerationCategory</i> , 1:n)                                                                                                                                                                             |
| <i>CommunityNote</i><br>herda: <i>as:Article</i>                                     | Nota pública e colaborativa que adiciona contexto, esclarecimento ou verificação de fatos a um recurso externo.                                                                                                                                                         | <i>as:name</i><br><i>as:content</i><br><i>as:attributedTo</i><br><i>as:to</i><br><i>inCommunityGroup</i> ( $\rightarrow$ <i>CommunityNoteGroup</i> , n:1)<br><i>category</i> ( $\rightarrow$ <i>ModerationCategory</i> , n:1)                      |
| <i>Transaction</i><br>herda:<br><i>as:OrderedCollection</i>                          | Um conjunto de atividades ( <i>as:Activity</i> ) que compõem uma transação distribuída.                                                                                                                                                                                 | <i>as:orderedItems</i> ( $\rightarrow$ <i>FindValue</i> , <i>FindNode</i> e respostas, 1:n)                                                                                                                                                        |
| <i>FindValue</i><br>herda: <i>as:Activity</i>                                        | Atividade responsável por solicitar a busca de um <i>CommunityNoteGroup</i> a partir do recurso/URL fornecido em <i>as:target</i> .                                                                                                                                     | <i>as:origin</i> ( $\rightarrow$ <i>as:Object</i> , n:1)<br><i>as:target</i> ( $\rightarrow$ <i>as:Object</i> , n:1)<br><i>inTransaction</i> ( $\rightarrow$ <i>Transaction</i> , n:1)                                                             |
| <i>FindValueResponse</i><br>herda: <i>as:Activity</i>                                | Atividade com a resposta à atividade <i>FindValue</i> .                                                                                                                                                                                                                 | <i>as:origin</i> ( $\rightarrow$ <i>as:Object</i> , n:1)<br><i>as:target</i> ( $\rightarrow$ <i>as:Object</i> , n:1)<br><i>as:result</i> ( $\rightarrow$ <i>as:Object</i> , n:1)<br><i>inTransaction</i> ( $\rightarrow$ <i>Transaction</i> , n:1) |
| <i>FindNode</i><br>herda: <i>as:Activity</i>                                         | Atividade responsável por solicitar a busca do nó responsável pelo recurso/URL preenchido na propriedade <i>as:target</i> .                                                                                                                                             | <i>as:origin</i> ( $\rightarrow$ <i>as:Object</i> , n:1)<br><i>as:target</i> ( $\rightarrow$ <i>as:Object</i> , n:1)<br><i>inTransaction</i> ( $\rightarrow$ <i>Transaction</i> , n:1)                                                             |
| <i>FindNodeResponse</i><br>herda: <i>as:Activity</i>                                 | Atividade com a resposta à atividade <i>FindNode</i> .                                                                                                                                                                                                                  | <i>as:origin</i> ( $\rightarrow$ <i>as:Object</i> , n:1)<br><i>as:target</i> ( $\rightarrow$ <i>as:Object</i> , n:1)<br><i>as:result</i> ( $\rightarrow$ <i>as:Object</i> , n:1)<br><i>inTransaction</i> ( $\rightarrow$ <i>Transaction</i> , n:1) |
| <i>NextNode</i><br>herda: <i>as:Object</i>                                           | Objeto com as informações do próximo nó onde a busca pelo recurso deve continuar. <i>as:origin</i> indica o recurso buscado e <i>as:target</i> indica a nova instância.                                                                                                 | <i>as:origin</i> ( $\rightarrow$ <i>as:Object</i> , n:1)<br><i>as:target</i> ( $\rightarrow$ <i>as:Object</i> , n:1)<br><i>inTransaction</i> ( $\rightarrow$ <i>Transaction</i> , n:1)                                                             |
| <i>NotFound</i><br>herda: <i>as:Object</i>                                           | Sinaliza que uma busca terminou sem encontrar um recurso.                                                                                                                                                                                                               | <i>as:origin</i> ( $\rightarrow$ <i>as:Object</i> , n:1)<br><i>as:target</i> ( $\rightarrow$ <i>as:Object</i> , n:1)                                                                                                                               |
| <i>ModerationNode</i><br>herda: <i>as:Service</i>                                    | Nó da rede P2P que armazena notas e participa do roteamento (DHT). Funciona como um ator ActivityPub para receber comandos. Utiliza o atributo <i>bucketItems</i> para vincular aos <i>buckets</i> de roteamento.                                                       | <i>as:name</i><br><i>as:summary</i><br><i>as:inbox</i><br><i>as:outbox</i><br><i>bucketItems</i> ( $\rightarrow$ <i>KBucket</i> , 1:n)                                                                                                             |
| <i>KBucket</i><br>herda: <i>as:Collection</i>                                        | <i>Bucket</i> na tabela de roteamento que agrupa referências a outros nós ( <i>KBucketNode</i> ) a uma distância Kademlia específica. Possui um tamanho máximo fixo ('k') de 20.                                                                                        | <i>as:items</i> ( $\rightarrow$ <i>KBucketNode</i> , 1:n)                                                                                                                                                                                          |
| <i>KBucketNode</i><br>herda: <i>as:Object</i>                                        | Representa outro nó na rede no <i>bucket</i> , contendo as informações necessárias para cálculo de distância e contato.                                                                                                                                                 | <i>as:target</i> ( $\rightarrow$ <i>as:Object</i> , n:1)                                                                                                                                                                                           |

**Fonte: Autoria própria.**

**Figura 21 – Exemplo do Objeto *ModerationNode* do D-COLMEIA.**

```

1  {
2    "@context": [
3      "https://www.w3.org/ns/activitystreams",
4      {
5        "dcolmeia": "https://dcolmeia.social/ns#",
6        "xsd": "http://www.w3.org/2001/XMLSchema#"
7      }
8    ],
9    "id": "https://dcolmeia.social/users/dcolmeia",
10   "type": "dcolmeia:ModerationNode",
11   "name": "Nó principal D-COLMEIA",
12   "summary": "Nó principal e ponto de entrada da rede D-COLMEIA.",
13   "inbox": "https://dcolmeia.social/users/dcolmeia/inbox",
14   "outbox": "https://dcolmeia.social/users/dcolmeia/outbox",
15   "dcolmeia:bucketItems": [
16     {
17       "type": "dcolmeia:KBucket",
18       "items": [
19         {
20           "id": "https://anotherhost.social/users/dcolmeia",
21           "type": "dcolmeia:KBucketNode",
22           "target": "https://another.social/users/dcolmeia"
23         }
24       ]
25     }
26   ]
27 }

```

**Fonte: Autoria própria.**

*et al.*, 2022). Charles *et al.* (2022) desenvolveram uma tipologia, apresentada no Anexo A, de alertas de conteúdo que foi utilizada como base para um padrão de categorias e subcategorias no D-COLMEIA. Segundo os autores, essa estrutura foi desenvolvida a partir de uma revisão sistemática da literatura e pode auxiliar na marcação de conteúdo de forma consistente e padronizada. Foi adicionada à ontologia do D-COLMEIA como uma estrutura de categorias para a `dcolmeia:CommunityNote`.

A Figura 23 descreve um grupo de notas de comunidade, que estende as `:Group` e as `:OrderedCollection` do *ActivityStreams 2.0*. Por herdar de `as:Group`, o objeto possui os atributos de ator `as:inbox` e `as:outbox`. Por herdar de `as:OrderedCollection`, possui os atributos `as:totalItems` e `as:orderedItems`, que descrevem a coleção ordenada de `dcolmeia:CommunityNote` associadas ao grupo. Também são utilizados os atributos `as:published`, `as:updated` e `as:inReplyTo`.

**Figura 22 – Exemplo do Objeto *dcolmeia:CommunityNote* do D-COLMEIA.**

```

1 {
2   "@context": [
3     "https://www.w3.org/ns/activitystreams",
4     {
5       "dcolmeia": "https://dcolmeia.social/ns#",
6       "xsd": "http://www.w3.org/2001/XMLSchema#"
7     }
8   ],
9   "id": "https://dcolmeia.social/note/xyz-789",
10  "type": "dcolmeia:CommunityNote",
11  "name": "Nome da nota",
12  "dcolmeia:category": "dcolmeia:injustice",
13  "content": "Conteúdo da nota",
14  "published": "2025-06-25T23:05:00Z",
15  "attributedTo": "https://another.social/users/dcolmeia",
16  "inReplyTo": "https://worstcontent.social/posts/abc-123"
17 }

```

**Fonte: Autoria própria.**

A ontologia do D-COLMEIA, como apresentada no Quadro 2, e visualmente no Apêndice A, foi projetada para representar semanticamente um sistema de moderação distribuída e colaborativa, aproveitando ao máximo as primitivas já estabelecidas nos padrões *ActivityPub* (W3C, 2018) e *ActivityStreams 2.0* (W3C, 2017). As classes definidas estendem o vocabulário base do *ActivityStreams* (prefixo *as:*), adicionando elementos específicos necessários para implementar a funcionalidade de moderação distribuída baseada em DHT Kademlia.

Para facilitar a leitura e identificação da origem de cada classe e propriedade ao longo do texto, adota-se a convenção de prefixos utilizada em ontologias: o prefixo *as:* refere-se ao vocabulário do *ActivityStreams 2.0* (*namespace* <https://www.w3.org/ns/activitystreams>), enquanto o prefixo *dcolmeia:* refere-se ao vocabulário definido pelo D-COLMEIA (*namespace* <https://dcolmeia.social/ns>).

Por ser um ator *ActivityPub*, o *dcolmeia:CommunityNoteGroup* possui *as:inbox* e *as:outbox* próprias, o que possibilita que usuários da rede federada interajam com o grupo de notas comunitárias da mesma forma que interagem com outros atores no fediverso. O endereço do ator segue o padrão [https://{instância}/communitynotes/{SHA1\(URL do recurso\)}](https://{instância}/communitynotes/{SHA1(URL do recurso)}), permitindo que qualquer instância construa deterministicamente a URL de um

**Figura 23 – Exemplo do Objeto *dcolmeia:CommunityNoteGroup* do D-COLMEIA, com a lista de itens resumida.**

```

1  {
2    "@context": [
3      "https://www.w3.org/ns/activitystreams",
4      {
5        "dcolmeia": "https://dcolmeia.social/ns#",
6        "xsd": "http://www.w3.org/2001/XMLSchema#"
7      }
8    ],
9    "id": "https://dcolmeia.social/communitynotes/a34...w34",
10   "type": "dcolmeia:CommunityNoteGroup",
11   "published": "2025-06-25T23:05:00Z",
12   "updated": "2025-06-06T08:12:00Z",
13   "inReplyTo": "https://worstcontent.social/posts/abc-123",
14   "inbox": "https://dcolmeia.social/communitynotes/a34...w34/inbox",
15   "outbox": "https://dcolmeia.social/communitynotes/a34...w34/outbox",
16   "totalItems": 2,
17   "orderedItems": [
18     "https://dcolmeia.social/moderate/note-001",
19     "https://dcolmeia.social/moderate/note-002"
20   ]
21 }

```

**Fonte: Autoria própria.**

*dcolmeia:CommunityNoteGroup* a partir da URL do recurso moderado, sem necessidade de consulta prévia. Cada *dcolmeia:CommunityNote* é vinculada ao seu grupo correspondente pela propriedade *dcolmeia:inCommunityGroup*.

A modelagem do *dcolmeia:CommunityNoteGroup* como subclasse simultânea de *as:Group* e *as:OrderedCollection* constitui uma decisão de projeto que prioriza a simplicidade estrutural sobre a separação formal entre ator e coleção preconizada pelo *ActivityStreams 2.0*. Em uma leitura rigorosa da especificação, atores e coleções desempenham papéis distintos. O ator é a entidade que envia e recebe atividades por meio de *as:inbox* e *as:outbox*, enquanto a coleção é uma estrutura de agrupamento de objetos, de natureza consultiva. O papel de ator do grupo segue o padrão de grupo-como-agregador formalizado pela *FEP-1b12 Group Federation* (Fediverse Enhancement Proposals, 2024b) e adotado por sistemas como o Lemmy (Lemmy Project, 2026), no qual o grupo é um ator seguível que responde a *as:Follow* com *as:Accept* e encapsula em *as:Announce* o conteúdo validado antes de encaminhá-lo aos seguidores. A esse papel, o D-COLMEIA acrescenta o de coleção ordenada de notas, de modo que *as:totalItems* e *as:orderedItems* sejam expostos diretamente no

próprio grupo. A combinação dos dois papéis em um único objeto é válida no modelo formal do RDF, no qual uma classe pode ser subclasse de várias e um recurso pode possuir múltiplos tipos, e foi escolhida por simplificar a estrutura do objeto e o seu endereçamento determinístico na DHT.

Usuários podem seguir um `dcolmeia:CommunityNoteGroup` enviando uma atividade `as:Follow` para a *inbox* do grupo. A atividade utiliza a propriedade `as:context`, definida na especificação do ActivityStreams 2.0 como um mecanismo para agrupar objetos e atividades que compartilham um propósito comum, para indicar a URL do recurso sendo moderado. Este uso do `as:context` permite que o nó responsável crie o `dcolmeia:CommunityNoteGroup` sob demanda caso ele ainda não exista, uma vez que o *hash* contido na URL do grupo pode ser validado a partir da URL do recurso informada no `as:context`. Dessa forma, usuários podem manifestar interesse na moderação de um recurso antes mesmo da criação da primeira `dcolmeia:CommunityNote`.

Ao receber a atividade `as:Follow`, o grupo responde automaticamente com uma atividade `as:Accept`, adicionando o usuário à sua coleção de seguidores. A partir deste momento, todas as novas `dcolmeia:CommunityNote` adicionadas ao grupo serão notificadas ao seguidor. A Figura 24 ilustra um exemplo de mensagem para seguir um grupo.

**Figura 24 – Requisição `as:Follow` para um `dcolmeia:CommunityNoteGroup`.**

```

1 {
2   "@context": "https://www.w3.org/ns/activitystreams",
3   "type": "Follow",
4   "id": "https://sports.social/users/alice/follows/123",
5   "actor": "https://sports.social/users/alice",
6   "object": "https://dcolmeia.social/communitynotes/a94a...bbd3",
7   "context": "https://worstcontent.social/posts/abc-123"
8 }
```

**Fonte: Autoria própria.**

A Figura 25 apresenta a resposta do `dcolmeia:CommunityNoteGroup` ao aceitar o seguidor.

Para deixar de seguir o grupo, o usuário envia uma atividade `as:Undo` que envolve a atividade `as:Follow` original. O grupo processa esta requisição removendo o usuário de sua coleção de seguidores, conforme ilustrado na Figura 26.

**Figura 25 – Resposta *as:Accept* do *dcolmeia:CommunityNoteGroup*.**

```

1  {
2    "@context": "https://www.w3.org/ns/activitystreams",
3    "type": "Accept",
4    "id":
5    ↪ "https://dcolmeia.social/communitynotes/a94a...8fe5/accepts/e8f4...ae32",
6    "actor": "https://dcolmeia.social/communitynotes/a94a...bbd3",
7    "object": {
8      "type": "Follow",
9      "id": "https://sports.social/users/alice/follows/123",
10     "actor": "https://sports.social/users/alice",
11     "object": "https://dcolmeia.social/communitynotes/a94a...bbd3"
12   }
13 }

```

**Fonte: Autoria própria.**

**Figura 26 – *as:Undo/as:Follow* para deixar de seguir um *dcolmeia:CommunityNoteGroup*.**

```

1  {
2    "@context": "https://www.w3.org/ns/activitystreams",
3    "type": "Undo",
4    "id": "https://sports.social/users/alice/undo/789",
5    "actor": "https://sports.social/users/alice",
6    "object": {
7      "type": "Follow",
8      "id": "https://sports.social/users/alice/follows/123",
9      "actor": "https://sports.social/users/alice",
10     "object": "https://dcolmeia.social/communitynotes/a94a...bbd3"
11   }
12 }

```

**Fonte: Autoria própria.**

Sempre que uma nova *dcolmeia:CommunityNote* é adicionada ao *dcolmeia:CommunityNoteGroup* através de uma atividade *as:Add* aceita, o grupo notifica todos os seus seguidores enviando uma atividade *as:Announce* para suas inboxes. Esta notificação contém a atividade *as:Add* original, permitindo que os seguidores sejam informados em tempo real sobre novas contribuições de moderação. Este mecanismo é análogo ao funcionamento de comunidades no Lemmy (Lemmy Project, 2026) e outros sistemas federados baseados em grupos (Fediverse Enhancement Proposals, 2024b). A Figura 27 demonstra como o grupo anuncia uma nova nota para seus seguidores.

**Figura 27 – *as:Announce* de nova *dcolmeia:CommunityNote* para seguidores.**

```

1  {
2    "@context": "https://www.w3.org/ns/activitystreams",
3    "type": "Announce",
4    "id":
5    ↪ "https://dcolmeia.social/communitynotes/a94a...8fe5/announces/3450...c56a",
6    "actor": "https://dcolmeia.social/communitynotes/a94a...bbd3",
7    "to": ["https://www.w3.org/ns/activitystreams#Public"],
8    "object": {
9      "type": "dcolmeia:CommunityNote",
10     "id": "https://sports.social/notes/note456",
11     "content": "Este artigo contém informações desatualizadas sobre...",
12     "target": "https://news.example/article/12345"
13   }
14 }

```

**Fonte: Autoria própria.**

As *dcolmeia:CommunityNote* são objetos *ActivityPub* completos e, como tais, podem receber interações de outros usuários da rede. O D-COLMEIA suporta as atividades *as:Like* e *as:Dislike* como mecanismos de avaliação da qualidade e utilidade das notas comunitárias. Estas avaliações podem ser utilizadas por algoritmos de ordenação e classificação para determinar quais notas devem ser exibidas com maior proeminência. As Figuras 28 e 29 exemplificam estas interações. O campo *audience* é utilizado para indicar o *dcolmeia:CommunityNoteGroup* ao qual a nota avaliada pertence, seguindo o padrão estabelecido pela *FEP-1b12 Group Federation* (Fediverse Enhancement Proposals, 2024b) para associação de conteúdo a atores do tipo *as:Group*.

**Figura 28 – *as:Like* em uma *dcolmeia:CommunityNote*.**

```

1  {
2    "@context": "https://www.w3.org/ns/activitystreams",
3    "type": "Like",
4    "id": "https://games.social/users/charlie/likes/555",
5    "actor": "https://games.social/users/charlie",
6    "object": "https://sports.social/notes/note456",
7    "audience": "https://dcolmeia.social/communitynotes/a94a...bbd3"
8  }

```

**Fonte: Autoria própria.**

**Figura 29 – *as:Dislike* em uma *dcolmeia:CommunityNote*.**

```

1 {
2   "@context": "https://www.w3.org/ns/activitystreams",
3   "type": "Dislike",
4   "id": "https://anime.social/users/diana/dislikes/777",
5   "actor": "https://anime.social/users/diana",
6   "object": "https://sports.social/notes/note456",
7   "audience": "https://dcolmeia.social/communitynotes/a94a...bbd3"
8 }

```

**Fonte: Autoria própria.**

Para ilustrar como os dados colaborativos de moderação disponibilizados pelo D-COLMEIA podem ser utilizados na prática, são apresentados a seguir cenários hipotéticos de uso.

No primeiro cenário, uma implementação do ActivityPub utiliza as atividades *as:Like* e *as:Dislike* recebidas em cada *dcolmeia:CommunityNote* como métrica de avaliação comunitária. A proporção entre avaliações positivas e negativas serve como indicador de consenso sobre a relevância e a veracidade da nota. Notas com alta proporção de *as:Like* podem ser exibidas com maior destaque junto ao conteúdo original, enquanto notas com predominância de *as:Dislike* podem ser ocultadas ou rebaixadas na ordenação.

No segundo cenário, as *dcolmeia:CommunityNote* são utilizadas como notas de contexto exibidas diretamente aos usuários, de forma análoga às *Community Notes* da plataforma X. Nesse modelo, o conteúdo da nota é apresentado junto à publicação original, oferecendo esclarecimentos, verificações de fatos ou alertas sobre o conteúdo, sem que a publicação seja removida ou censurada.

No terceiro cenário, as *dcolmeia:CommunityNote* são direcionadas exclusivamente aos moderadores da instância, funcionando como um sistema colaborativo de apoio à decisão. Os moderadores recebem as notas como insumo para suas decisões de moderação local, podendo optar por aplicar advertências, restringir a visibilidade do conteúdo ou tomar outras medidas conforme as políticas da instância.

Esses cenários não são mutuamente exclusivos e podem ser combinados por uma mesma implementação. O protocolo D-COLMEIA não prescreve nenhum desses comportamentos, limitando-se a fornecer a infraestrutura de dados e comunicação sobre a qual cada plataforma

define suas próprias políticas de apresentação e moderação.

Além de permitir que usuários individuais acompanhem a moderação de recursos específicos, o D-COLMEIA também possibilita que instâncias colaborem de forma proativa entre si. Para isso, cada nó expõe um ator especial do tipo `dcolmeia:ModerationNode` denominado `dcolmeia`. Este ator representa o próprio nó na rede federada e pode ser seguido por outros nós através do mecanismo padrão de `as:Follow` do ActivityPub. Quando o usuário `dcolmeia` de uma instância deseja ser notificado sobre novos `dcolmeia:CommunityNoteGroup` criados em outra instância, ele envia uma atividade `as:Follow` para a *inbox* do ator `dcolmeia` da instância remota. Uma vez estabelecida a relação de seguidor, sempre que a instância remota criar um novo `dcolmeia:CommunityNoteGroup`, ela notifica automaticamente todos os seguidores de seu ator `dcolmeia` através de uma atividade `as:Announce`, tendo o `dcolmeia:CommunityNoteGroup` recém-criado como valor da propriedade `as:object`. Este mecanismo, assim como o mecanismo de seguir `dcolmeia:CommunityNoteGroup`, implementa um modelo *push* de disseminação de informações de moderação, complementando as consultas *pull* realizadas via DHT, e permitindo que nós interessados mantenham-se atualizados em tempo real sobre as atividades de moderação de seus pares na rede federada.

É importante ressaltar que o protocolo D-COLMEIA define exclusivamente a estrutura de dados e o fluxo de mensagens para moderação colaborativa distribuída. A forma como estas informações são apresentadas ao usuário final—incluindo a interface de usuário, a visualização das notas, os algoritmos de ordenação, e os critérios para destacar ou ocultar conteúdo—é de responsabilidade de cada implementação específica do ActivityPub que incorporar o D-COLMEIA. Esta separação entre protocolo e apresentação permite que diferentes plataformas adaptem o sistema às suas necessidades e contextos particulares, mantendo a interoperabilidade na camada de protocolo. Da mesma forma, políticas de expiração, remoção ou arquivamento de `dcolmeia:CommunityNote` e `dcolmeia:CommunityNoteGroup` não são definidas pelo protocolo, ficando a critério de cada implementação.

Para dar contexto padronizado às notas, a classe `dcolmeia:ModerationCategory` foi criada como extensão de `as:Object`, permitindo a classificação das notas segundo tipologias estabelecidas, como a tipologia NEON de avisos de conteúdo, que foi desenvolvida por Charles *et al.* (2022). Essa categorização facilita o processamento automatizado e a aplicação consistente de políticas de moderação entre diferentes instâncias.

A comunicação distribuída é orquestrada por um conjunto de atividades especializadas. As classes `dcolmeia:FindValue` e `dcolmeia:FindNode`, ambas estendendo `as:Activity`, representam as operações fundamentais de busca na DHT, sendo a primeira para localizar um `dcolmeia:CommunityNoteGroup` específico e a segunda para identificar o nó responsável por um recurso. Suas contrapartes, `dcolmeia:FindValueResponse` e `dcolmeia:FindNodeResponse`, encapsulam as respostas respectivas, todas utilizando as propriedades `as:origin` e `as:target`, além de `dcolmeia:inTransaction` para vinculação com transações distribuídas. A classe `dcolmeia:Transaction`, estendendo `as:OrderedCollection`, agrupa essas atividades em operações atômicas distribuídas através da propriedade `as:orderedItems`.

Para casos onde a busca deve continuar iterativamente pela rede, os objetos `dcolmeia:NextNode` indicam o próximo nó a ser consultado, enquanto `dcolmeia:NotFound` sinaliza o término infrutífero de uma busca, ambos utilizando `as:origin` e `as:target` para manter o contexto da operação.

As atividades de busca e roteamento do D-COLMEIA reutilizam as propriedades `as:origin` e `as:target` do *ActivityStreams 2.0*, estendendo ao contexto de roteamento na DHT a semântica conceitual definida pelo W3C (W3C, 2017), segundo a qual `as:origin` descreve o ponto de partida a partir do qual a atividade é conduzida e `as:target` descreve o seu destino. No D-COLMEIA, o ponto de referência de toda operação de busca é o recurso que está sendo moderado, de modo que `as:origin` identifica esse recurso e o mantém constante ao longo dos saltos de uma mesma busca, enquanto `as:target` indica, em geral, a instância de destino para a qual a mensagem é dirigida, como no redirecionamento via `dcolmeia:NextNode`. Nas requisições iniciais `dcolmeia:FindValue` e `dcolmeia:FindNode`, em que ainda não há uma instância de destino determinada, `as:target` indica o próprio recurso consultado, que define o alvo lógico da busca a ser resolvido pelo roteamento subsequente. Optou-se por reaproveitar essas propriedades nativas, em vez de definir novos termos no vocabulário `dcolmeia:`, de modo a maximizar a aderência ao vocabulário base e reduzir a superfície de extensão da ontologia.

A infraestrutura de rede é representada pela classe `dcolmeia:ModerationNode`, que utiliza a propriedade `dcolmeia:bucketItems` para referenciar sua tabela de roteamento Kademlia, composta por objetos `dcolmeia:KBucket` que agrupam até 20 referências a outros nós (`dcolmeia:KBucketNode`) a uma distância XOR específica na DHT.

### 3.3 COMUNICAÇÃO ENTRE OS NÓS D-COLMEIA

Para viabilizar a colaboração entre instâncias na moderação de conteúdo, o D-COLMEIA necessita de um mecanismo de comunicação que permita armazenar, localizar e disseminar informações de moderação de forma descentralizada. A arquitetura de comunicação e distribuição inspira-se no protocolo Kademlia, detalhado na Seção 2.8. Internamente, cada servidor D-COLMEIA mantém uma estrutura de roteamento baseada em *k-buckets*, análoga à do Kademlia. Para gerar o identificador dos nós nesta DHT, o sistema utiliza o algoritmo SHA-1 para gerar identificadores de 160 bits a partir do nome de domínio e da porta de comunicação de cada instância, o que difere do Kademlia clássico, que utiliza *hashes* aleatórios para identificar os nós. Por exemplo, o ID para `https://dcolmeia.social` é calculado como:

$$\text{sha1}(\text{dcolmeia.social}:443) = f78f\dots f8ce$$

Para integrar essa estrutura à web federada, cada nó D-COLMEIA é implementado como um ator especial do ActivityStreams 2.0. As informações deste ator são descobertas publicamente via WebFinger, um protocolo para descoberta de informações sobre identidades digitais (PRODROMOU, 2024).

O retorno do WebFinger, ilustrado na Figura 30, segue os padrões definidos nas RFCs 7033 e 7565 (JONES *et al.*, 2013; SAINT-ANDRE, 2015). A identificação de que um nó implementa o D-COLMEIA ocorre quando um objeto no array `links` possui simultaneamente o atributo `rel` com o valor `self` e um objeto `properties` contendo a chave `https://www.w3.org/ns/activitystreams#type` com o valor `https://dcolmeia.social/ns#ModerationNode`.

A consulta à *Uniform Resource Locator* (URL) presente no campo `href` retorna o objeto completo do ator, conforme o exemplo da Figura 21. Este objeto, do tipo `dcolmeia:ModerationNode`, contém os endereços de sua *inbox* e *outbox*, que são essenciais para a comunicação entre instâncias.

O processo de consulta utilizando WebFinger é idêntico à descoberta de outros atores do Fediverso, como Usuários e Grupos. Para isso, utiliza-se o identificador padronizado `dcolmeia@instancia`. Por exemplo:

- Para obter as informações do nó da instância `dcolmeia.social`, a requisição GET seria feita para o *endpoint*: `https://dcolmeia.social/.well-known/webfinger?resource=acct:dcolmeia@dcolmeia.social`;

**Figura 30 – Exemplo do retorno WebFinger para um ator do D-COLMEIA.**

```

1  {
2    "subject": "acct:dcolmeia@sua_instancia.social",
3    "aliases": [
4      "https://sua_instancia.social/users/dcolmeia"
5    ],
6    "links": [
7      {
8        "rel": "self",
9        "type": "application/activity+json",
10       "href": "https://sua_instancia.social/users/dcolmeia",
11       "properties": {
12         "https://www.w3.org/ns/activitystreams#type":
↪    "https://dcolmeia.social/ns#ModerationNode"
13       }
14     }
15   ]
16 }

```

**Fonte:** Autoria própria.

- Similarmente, para interagir com a instância *sports.social*, o *endpoint* seria: `https://sports.social/.well-known/webfinger?resource=acct:dcolmeia@sports.social`.

Sobre o armazenamento e distribuição de valores, uma diferença fundamental entre o D-COLMEIA e o Kademlia clássico está na aplicação do armazenamento chave-valor. Enquanto o Kademlia oferece uma API genérica para este fim, o D-COLMEIA emprega o conceito apenas internamente, como o mecanismo para distribuir os dados da moderação.

Para cada conteúdo moderado, o par chave-valor é definido da seguinte maneira:

- **Chave:** O *hash* SHA-1 do URI da mensagem *ActivityStreams 2.0*;
- **Valor:** O objeto completo da própria mensagem.

Como exemplo prático, para armazenar os dados de moderação da mensagem com URL `https://worstcontent.social/posts/abc-123`, a chave correspondente na DHT seria o seu *hash*:

$$\text{sha1}(\text{https://worstcontent.social/posts/abc-123}) = 0b4e...3138$$

Assim como no Kademlia clássico, o par chave-valor é armazenado na instância mais próxima do recurso. Essa proximidade é definida pela métrica de distância XOR aplicada nos *hashes* do identificador da instância e da mensagem a ser moderada. Desta forma, o D-COLMEIA

garante que o conteúdo de moderação seja armazenado e distribuído de forma determinística, permitindo a sua localização por qualquer instância interessada, participante ou não da rede D-COLMEIA.

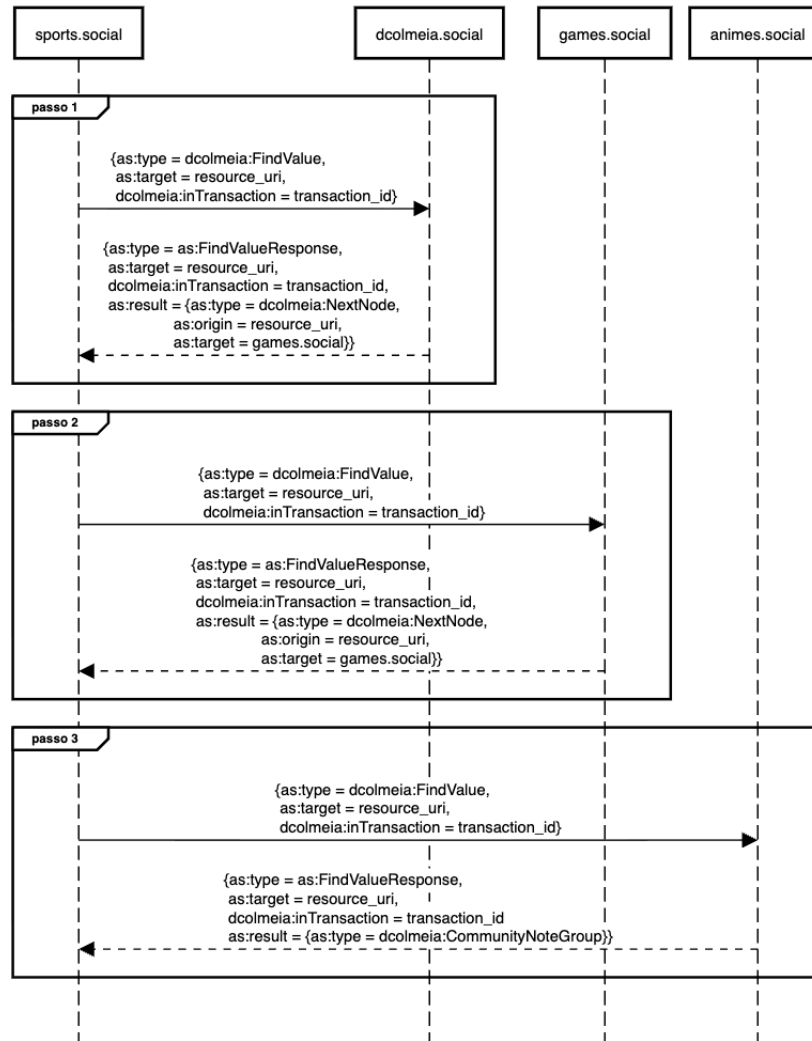
Para que os fluxos descritos a seguir operem corretamente, é necessário que a tabela de roteamento (*k-buckets*) de cada nó esteja populada com referências a outros nós da rede. O ingresso de um novo nó na rede D-COLMEIA requer o conhecimento prévio de ao menos um nó já participante, denominado nó de *bootstrap*. Qualquer instância que implemente o D-COLMEIA pode desempenhar esse papel, porém uma ou mais instâncias podem ser designadas como pontos de entrada padronizados, como a instância *dcolmeia.social*.

Ao ingressar na rede, o novo nó envia uma atividade *as:Announce* para a *inbox* do ator *dcolmeia* do nó de *bootstrap*, tendo como *as:object* o seu próprio ator *dcolmeia:ModerationNode*. O nó de *bootstrap* adiciona o novo nó à sua tabela de roteamento e responde com referências a nós presentes em seus *k-buckets*. Com essas referências, o novo nó popula sua tabela de roteamento e pode, opcionalmente, realizar uma operação *FIND\_NODE* sobre seu próprio identificador para descobrir nós vizinhos adicionais, de forma análoga ao *bootstrap* do Kademlia original (MAYMOUNKOV; MAZIERES, 2002).

A tabela de roteamento também é alimentada de forma contínua e oportunista. Durante a operação normal, ao receber mensagens *ActivityPub*, o nó pode extrair os domínios presentes em campos como *as:actor* e *as:attributedTo*, consultando o *WebFinger* de cada instância para verificar se esta implementa o D-COLMEIA. Caso possua um ator do tipo *dcolmeia:ModerationNode*, suas informações são adicionadas ao *k-bucket* correspondente, permitindo que a tabela de roteamento cresça organicamente com as interações no fediverso.

Diferentemente do Kademlia clássico, que opera sobre chamadas RPC síncronas, o D-COLMEIA foi projetado sobre o paradigma do *ActivityPub*, que diferente da comunicação RPC, é assíncrona. O padrão request/reply foi mantido para preservar a semântica das RPCs originais do Kademlia. Essa mudança fundamental na forma de comunicação levou à adaptação das quatro operações principais: *FIND\_NODE*, *FIND\_VALUE*, *STORE* e *PING*. A explicação dos fluxos a seguir assume um comportamento padrão dos *k-buckets*, conforme a especificação Kademlia. Nesse modelo, a busca por um nó inicia-se sempre localmente, e a requisição é subsequentemente enviada para a instância que possui o *hash* mais próximo ao do recurso de destino.

**Figura 31 – Adaptação do FIND\_VALUE do Kademlia para o ActivityPub.**



**Fonte: Autoria própria.**

A Figura 31 ilustra um diagrama de sequência que descreve o processo de busca por um `dcolmeia:CommunityNoteGroup` na rede D-COLMEIA, compatível com o `FIND_VALUE` do Kademlia clássico, considerando que a descoberta via `WebFinger` já tenha sido feita. A interação ocorre entre quatro instâncias distintas: *sports.social*, *dcolmeia.social*, *games.social* e *animes.social*. O processo é executado de forma iterativa, utiliza um identificador de transação (`dcolmeia:inTransaction`) para rastreamento de requisições, e é dividido em três passos principais:

- Passo 1**
- A busca é iniciada pela instância *sports.social*, que envia uma mensagem do tipo `dcolmeia:FindValue` para um nó de referência, *dcolmeia.social*. O alvo da busca (`as:target`) é um recurso genérico identificado por `resource_uri`, e a requisição inclui a propriedade `dcolmeia:inTransaction` com um identifica-

dor único (`transaction_id`) que permite rastrear todas as mensagens pertencentes a esta operação;

- A instância *dcolmeia.social*, ao não possuir o recurso, responde à requisição com uma mensagem `dcolmeia:FindValueResponse`. O campo `as:result` desta mensagem contém um objeto `dcolmeia:NextNode`, que atua como um ponteiro, indicando que o próximo nó a ser consultado (`as:target`) é *games.social*. A resposta mantém o mesmo `transaction_id` da requisição original.

**Passo 2** • Seguindo a indicação recebida, a instância *sports.social* envia uma nova requisição `dcolmeia:FindValue` para o próximo nó da rota, *games.social*, mantendo o mesmo `transaction_id`;

- De forma análoga ao passo anterior, *games.social* também não armazena o recurso e responde com uma mensagem `dcolmeia:FindValueResponse` contendo um objeto `dcolmeia:NextNode`, desta vez apontando para a instância *animex.social*.

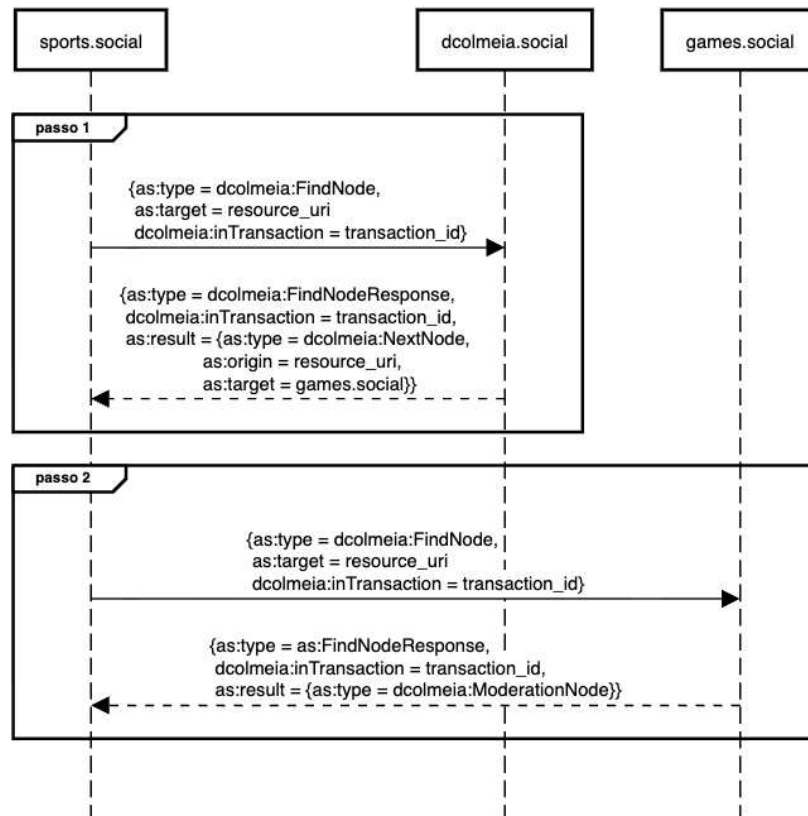
**Passo 3** • A instância *sports.social* realiza a consulta final, enviando a mensagem `dcolmeia:FindValue` para a instância *animex.social*, mantendo o mesmo `transaction_id`;

- A instância *animex.social* localiza a informação. A resposta é uma mensagem `dcolmeia:FindValueResponse` cujo resultado (`as:result`) é um objeto do tipo `dcolmeia:CommunityNoteGroup`. Este objeto representa o conjunto de notas comunitárias encontrado, sinalizando o término bem-sucedido da busca.

A Figura 32 ilustra um diagrama de sequência que descreve o processo de busca iterativa na rede D-COLMEIA. O fluxo é compatível com as operações `FIND_NODE` do Kademlia clássico. A interação ocorre entre três instâncias distintas: *sports.social*, *dcolmeia.social* e *games.social*. O processo é executado de forma iterativa, utiliza um identificador de transação (`dcolmeia:inTransaction`) para rastreamento de requisições, e é dividido em dois passos principais:

**Passo 1** • A busca é iniciada pela instância *sports.social*, que envia uma mensagem do tipo `dcolmeia:FindNode` para um nó de referência, *dcolmeia.social*. O alvo da busca (`as:target`) é um recurso genérico identificado por `resource_uri`, e a requisição inclui a propriedade `dcolmeia:inTransaction` com um identificador

**Figura 32 – Adaptação do FIND\_NODE do Kademlia para o ActivityPub, com foco no comportamento adotado quando o valor não é encontrado.**



Fonte: Autoria própria.

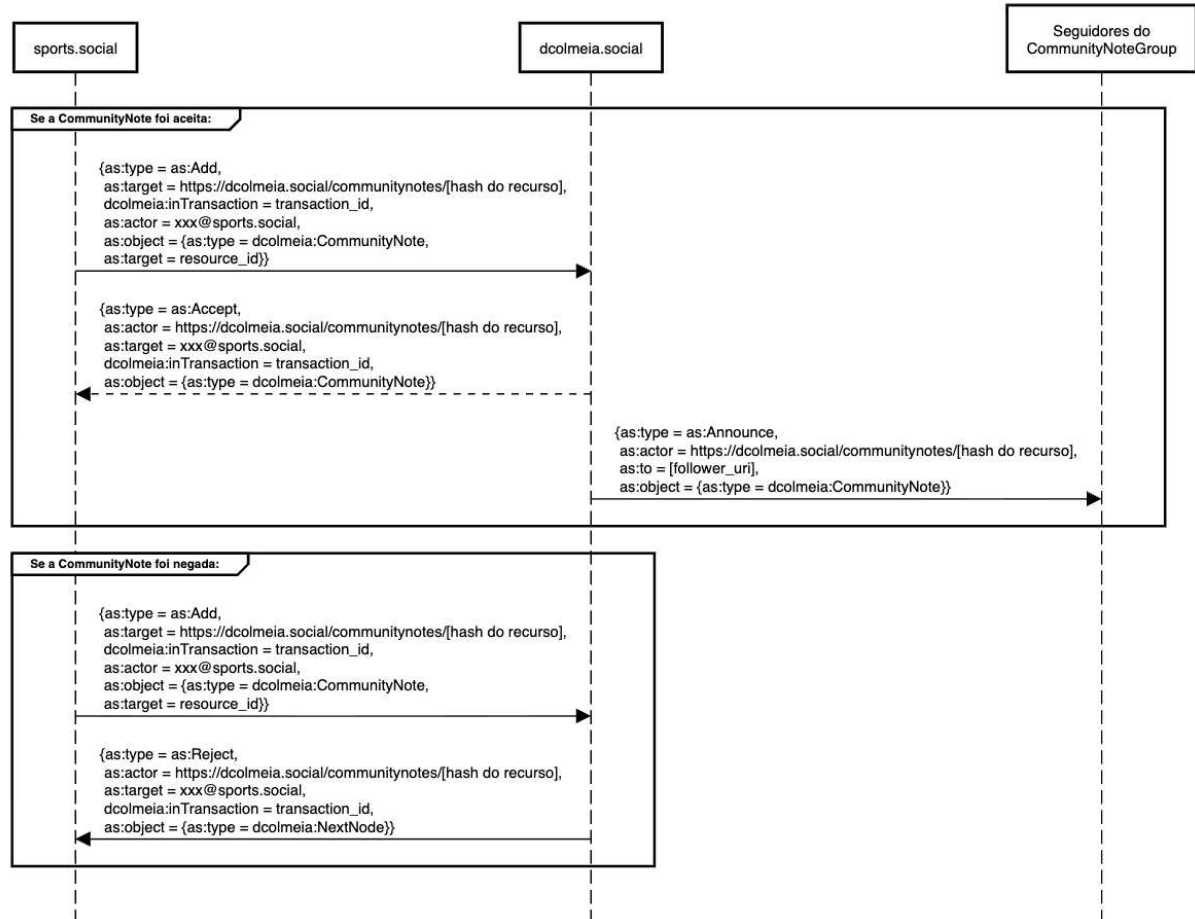
único (`transaction_id`) que permite rastrear todas as mensagens pertencentes a esta operação;

- A instância *dcolmeia.social*, ao não possuir o recurso, responde à requisição com uma mensagem `dcolmeia:FindNodeResponse`. O campo `as:result` desta mensagem contém um objeto `dcolmeia:NextNode`, que atua como um ponteiro, indicando que o próximo nó a ser consultado (`as:target`) é *games.social*. A resposta mantém o mesmo `transaction_id` da requisição original.

- Passo 2**
- Seguindo a indicação recebida, a instância *sports.social* envia uma nova requisição `dcolmeia:FindNode` para o próximo nó da rota, *games.social*, mantendo o mesmo `transaction_id`;
  - A instância *games.social* consulta sua tabela de roteamento e conclui que ela mesma é o nó mais próximo do recurso buscado. Como o valor não foi localizado, ela responde com uma mensagem `dcolmeia:FindNodeResponse` cujo resultado (`as:result`) é um objeto do tipo `dcolmeia:ModerationNode`. Este objeto

representa a própria instância *games.social*, sinalizando o término da busca com a identificação do nó responsável pela chave.

**Figura 33 – Adaptação do STORE do Kademlia para o ActivityPub.**



**Fonte: Autoria própria.**

A Figura 33 ilustra o diagrama de sequência que descreve a operação de armazenamento (STORE) do Kademlia clássico adaptada para o D-COLMEIA. O fluxo assume que o nó de destino já foi determinado por uma busca prévia (conforme o fluxo FIND\_NODE). A interação ocorre entre a instância *sports.social*, que deseja armazenar uma *dcolmeia:CommunityNote*, a instância *dcolmeia.social*, que hospeda o *dcolmeia:CommunityNoteGroup* de destino, e os seguidores desse grupo. O armazenamento é direcionado a um *dcolmeia:CommunityNoteGroup*, um ator ActivityPub dos tipos *as:Group* e *as:OrderedCollection* e identificado pelo hash SHA-1 do recurso moderado. O processo utiliza um identificador de transação (*dcolmeia:inTransaction*) para rastreamento e é dividido em dois cenários:

- Passo 1 (Requisição de armazenamento)** • A instância *sports.social* envia uma atividade do tipo `as:Add` para o `dcolmeia:CommunityNoteGroup` em *dcolmeia.social*. O campo `as:target` especifica o identificador do grupo (e.g., `https://dcolmeia.social/communitynotes/[hashdorecurso]`), que corresponde à chave na DHT. O campo `as:object` contém a `dcolmeia:CommunityNote` a ser armazenada, que por sua vez possui um campo `as:target` indicando o `resource_id` do conteúdo moderado. A transação é rastreada pelo campo `dcolmeia:inTransaction`;
- Ao receber a requisição, a instância *dcolmeia.social* verifica se o `dcolmeia:CommunityNoteGroup` correspondente já existe. Caso o grupo ainda não tenha sido criado, ele é automaticamente instanciado a partir do hash SHA-1 do recurso, garantindo que todas as notas sobre o mesmo conteúdo sejam agregadas no mesmo grupo.

- Passo 2a (Nota aceita)** • Se a `dcolmeia:CommunityNote` atende aos critérios definidos, o `dcolmeia:CommunityNoteGroup` responde com uma atividade `as:Accept`. O `as:actor` é o próprio grupo, o `as:target` é o usuário que enviou a requisição (*xxx@sports.social*), e o campo `as:object` contém a `dcolmeia:CommunityNote` aceita. O mesmo `transaction_id` é mantido para correlação;
- Após a aceitação, o grupo emite uma atividade `as:Announce` endereçada individualmente a cada seguidor do grupo (`as:to = [follower_uri]`). Este mecanismo de *push*, baseado no padrão Follow/Announce do ActivityPub, permite que instâncias interessadas recebam notificações sobre novas notas de moderação sem necessidade de consultas periódicas via `FIND_VALUE`.

- Passo 2b (Nota rejeitada)** • Caso a nota não atenda aos critérios de qualidade, viole políticas locais, ou o nó não seja o responsável pela chave, o `dcolmeia:CommunityNoteGroup` responde com uma atividade `as:Reject`. A estrutura da resposta é análoga ao cenário de aceitação, mantendo o mesmo `transaction_id` e referenciando a `dcolmeia:CommunityNote` rejeitada no campo `as:object`;
- Neste cenário, nenhuma atividade `as:Announce` é emitida, pois a nota não foi incorporada à coleção do grupo.

Para finalizar, o `PING` pode ser realizado utilizando o próprio `WebFinger` da instância, consultando o ator `dcolmeia@instância`. Se o nó estiver ativo, este irá retornar os dados para localizar o ator `dcolmeia:ModerationNode`, como descrito anteriormente.

O próximo capítulo valida a arquitetura D-COLMEIA através da análise de casos de uso e da apresentação de resultados obtidos em simulações de ambiente controlado.

## 4 RESULTADOS

Este capítulo apresenta os resultados obtidos a partir da validação funcional da arquitetura D-COLMEIA, detalhada no capítulo anterior. Para este fim, foi desenvolvido um protótipo funcional e instanciada uma rede de testes em um ambiente controlado, permitindo a observação e a análise do comportamento do protótipo proposto. Os resultados demonstram a viabilidade da arquitetura, validando a comunicação entre os nós de forma descentralizada e a execução das operações fundamentais do sistema. Aspectos relacionados a desempenho e escalabilidade não fazem parte do escopo desta pesquisa, sendo indicados como direções para trabalhos futuros.

A estrutura deste capítulo está organizada da seguinte forma. Inicialmente, a Seção 4.1 descreve a implementação do protótipo e o ambiente de testes utilizado. As seções subsequentes são dedicadas à demonstração prática das operações Kademlia adaptadas: a Seção 4.2 detalha o armazenamento de dados (`STORE`), junto com a busca da instância onde a informação será armazenada `FIND_NODE`; e a Seção 4.3 apresenta a busca por valores (`FIND_VALUE`). Por fim, a Seção 4.4 realiza uma discussão sobre os resultados observados, focando nos desafios e nas implicações do modelo de comunicação assíncrono do D-COLMEIA em comparação com as operações do Kademlia clássico, e também possíveis melhorias a serem efetuadas.

### 4.1 PROTÓTIPO

Para validar o funcionamento do D-COLMEIA, foi desenvolvido um protótipo da arquitetura com as tecnologias Java 25 e Spring Boot 4 compilando nativamente utilizando GraalVM. A compilação nativa por meio do GraalVM foi adotada para reduzir significativamente o consumo de memória de cada instância, viabilizando a execução simultânea de um maior número de nós no ambiente de testes. Para a verificação, foram criados testes de integração em Python 3 com a ferramenta *pytest*. O protótipo também implementa funcionalidades essenciais do ActivityPub, como as atividades de seguir (`as:Follow`) e criar notas (`as:Note`), além da estrutura de *inbox*.

A partir do protótipo, foi instanciada uma rede local com 64 nós D-COLMEIA utilizando Docker e Traefik. O Docker é uma plataforma de containerização que permite empacotar aplicações e suas dependências em unidades isoladas denominadas contêineres, possibilitando a execução simultânea de múltiplas instâncias independentes em um mesmo ambiente (Docker

Inc., 2025). O Traefik é um *proxy* reverso que atua como ponto de entrada da rede, realizando o roteamento de requisições com base no nome de *host* de cada contêiner (Traefik Labs, 2025). Com essa configuração, cada nó da rede pôde ser acessado por meio de seu próprio nome de domínio, simulando o comportamento de instâncias independentes em uma rede federada. O número de 64 nós foi definido em função das limitações de memória RAM do ambiente de execução disponível, representando o número máximo de instâncias simultâneas que puderam ser mantidas de forma estável durante os testes.

Para aproximar o ambiente de testes de um cenário realista, os nomes de *host* dos nós foram definidos a partir de instâncias reais do fediverso. Foi desenvolvido um *script* que consulta a API do serviço *instances.social* e seleciona as 63 instâncias com maior número de usuários ativos. A estas foi adicionado o *host* fictício *dcolmeia.social*, que atua como nó de *bootstrap* da rede, totalizando 64 nós. No ambiente local, cada nó recebeu o sufixo *.local* em seu nome de domínio (por exemplo, *mastodon.social.local*), permitindo que as tabelas *hash* geradas pela função SHA-1 sobre os nomes de *host* produzam distribuições semelhantes às que ocorreriam em um ambiente real do fediverso.

A Tabela 1 apresenta a lista das 64 instâncias D-COLMEIA utilizadas no ambiente de simulação, juntamente com seus respectivos identificadores de 160 bits gerados pelo algoritmo SHA-1. Para cada instância, a tabela exibe o *hash* SHA-1 correspondente em duas representações, sendo um formato hexadecimal reduzido para facilitar a leitura. As instâncias estão ordenadas de forma crescente com base nos *hash*, o que reflete suas posições relativas no espaço de identificadores da rede DHT.

## 4.2 CRIAÇÃO DE NOTAS DE MODERAÇÃO

Esta seção exemplifica a operação de criação de notas de moderação no D-COLMEIA, se baseando no (STORE) do Kademlia. O processo é realizado através da criação e da persistência distribuída de uma nova `dcolmeia:CommunityNote`, a associando a um `dcolmeia:CommunityNoteGroup` e detalhando o fluxo de mensagens do ActivityPub necessário para sua correta alocação na rede DHT.

Para demonstrar a correta implementação da métrica de distância XOR e do roteamento Kademlia no D-COLMEIA, foram calculadas distâncias fora do D-COLMEIA. O objetivo é verificar se, dado um recurso, o protocolo identifica corretamente o nó da rede que deve ser responsável por seu armazenamento, conforme a regra do nó mais próximo.

**Tabela 1 – Distribuição dos nós participantes no espaço de endereçamento SHA-1 da rede D-COLMEIA**

| #  | Instância                               | Hash SHA-1         | #  | Instância                           | Hash SHA-1         |
|----|-----------------------------------------|--------------------|----|-------------------------------------|--------------------|
| 1  | <i>techhub.social.local</i>             | <i>00f4...e797</i> | 33 | <i>mstdn.party.local</i>            | <i>94be...fac2</i> |
| 2  | <i>det.social.local</i>                 | <i>0495...d292</i> | 34 | <i>universeodon.com.local</i>       | <i>9597...6aad</i> |
| 3  | <i>mastodon.world.local</i>             | <i>0cfe...626c</i> | 35 | <i>pawoo.net.local</i>              | <i>9b9a...f97d</i> |
| 4  | <i>nrv.social.local</i>                 | <i>0e13...e0e8</i> | 36 | <i>troet.cafe.local</i>             | <i>a0ab...41a6</i> |
| 5  | <i>flipboard.social.local</i>           | <i>14ef...57d2</i> | 37 | <i>mathstodon.xyz.local</i>         | <i>a6ac...d21a</i> |
| 6  | <i>mastodon.nl.local</i>                | <i>192e...4625</i> | 38 | <i>infosec.exchange.local</i>       | <i>a977...2de8</i> |
| 7  | <i>alpha.argyle.social.local</i>        | <i>3306...a2f9</i> | 39 | <i>mastodontech.de.local</i>        | <i>aeda...ce83</i> |
| 8  | <i>c.im.local</i>                       | <i>4010...f10e</i> | 40 | <i>tech.lgbt.local</i>              | <i>af08...1e6d</i> |
| 9  | <i>hachyderm.io.local</i>               | <i>461d...d84e</i> | 41 | <i>mastodon.uno.local</i>           | <i>b3fc...7b97</i> |
| 10 | <i>social.vivaldi.net.local</i>         | <i>4990...d2c0</i> | 42 | <i>sfba.social.local</i>            | <i>b4eb...16e0</i> |
| 11 | <i>masto.es.local</i>                   | <i>4b92...e44a</i> | 43 | <i>indieweb.social.local</i>        | <i>b9d7...d1f5</i> |
| 12 | <i>mastodon.sdf.org.local</i>           | <i>4c0e...f099</i> | 44 | <i>dcolmeia.social.local</i>        | <i>baa2...d011</i> |
| 13 | <i>pouet.chapril.org.local</i>          | <i>522c...e996</i> | 45 | <i>link.baai.ac.cn.local</i>        | <i>bdcf...b4ca</i> |
| 14 | <i>social-md.wunderground.com.local</i> | <i>5580...f8c2</i> | 46 | <i>mastodonapp.uk.local</i>         | <i>c2cd...d9f4</i> |
| 15 | <i>norden.social.local</i>              | <i>56c3...a246</i> | 47 | <i>social.tchncs.de.local</i>       | <i>c5d7...c289</i> |
| 16 | <i>noauthority.social.local</i>         | <i>57ac...5268</i> | 48 | <i>occm.cc.local</i>                | <i>c610...0677</i> |
| 17 | <i>loforo.com.local</i>                 | <i>5ab0...4f3e</i> | 49 | <i>social.linux.pizza.local</i>     | <i>cb9e...80f0</i> |
| 18 | <i>kolektiva.social.local</i>           | <i>5e4d...f21f</i> | 50 | <i>mastodon.ie.local</i>            | <i>d124...e9ce</i> |
| 19 | <i>ohai.social.local</i>                | <i>69e4...8eba</i> | 51 | <i>planet.moe.local</i>             | <i>d3e5...60f4</i> |
| 20 | <i>mastodon.scot.local</i>              | <i>6d32...fda5</i> | 52 | <i>wxw.moe.local</i>                | <i>d6bd...6a38</i> |
| 21 | <i>aethy.com.local</i>                  | <i>6e0b...f307</i> | 53 | <i>mastodon.gamedev.place.local</i> | <i>d778...7837</i> |
| 22 | <i>defcon.social.local</i>              | <i>7013...76ae</i> | 54 | <i>mastodon.art.local</i>           | <i>da65...883f</i> |
| 23 | <i>livellosegreto.it.local</i>          | <i>7124...b431</i> | 55 | <i>mstdn.ca.local</i>               | <i>dcf2...7211</i> |
| 24 | <i>digitalcourage.social.local</i>      | <i>71f1...0237</i> | 56 | <i>alive.bar.local</i>              | <i>ddd2...b99a</i> |
| 25 | <i>wxw.ooo.local</i>                    | <i>773d...774e</i> | 57 | <i>mastodon.social.local</i>        | <i>de0f...db50</i> |
| 26 | <i>dice.camp.local</i>                  | <i>7946...0c23</i> | 58 | <i>beta.argyle.social.local</i>     | <i>dfa5...70b3</i> |
| 27 | <i>m.cmx.im.local</i>                   | <i>7b91...ff5a</i> | 59 | <i>piaille.fr.local</i>             | <i>e347...675c</i> |
| 28 | <i>toot.community.local</i>             | <i>813a...aaa5</i> | 60 | <i>ruhr.social.local</i>            | <i>e848...de55</i> |
| 29 | <i>mstdn.social.local</i>               | <i>821e...7719</i> | 61 | <i>relay.mastodon.nl.local</i>      | <i>ea4c...8094</i> |
| 30 | <i>sueden.social.local</i>              | <i>8480...c596</i> | 62 | <i>hessen.social.local</i>          | <i>f745...dafa</i> |
| 31 | <i>mastodon.cloud.local</i>             | <i>8bf1...b186</i> | 63 | <i>theforkiverse.com.local</i>      | <i>ffa9...48c8</i> |
| 32 | <i>thu.closed.social.local</i>          | <i>9472...64ec</i> | 64 | <i>mastodon.nu.local</i>            | <i>ffd7...c254</i> |

Fonte: Autoria própria.

O valor-alvo para a busca foi o URI <https://worstcontent.social/posts/abcd-1234>. O *hash* SHA-1 de 160 bits correspondente a este URI, que serve como a chave na DHT, é:

- **Valor:** <https://worstcontent.social/posts/abcd-1234>
- **Hash (hex):** *c29502fe2309211e4cc5064297970cf79e1d2c73*
- **Hash (int):**  $1,11 \times 10^{48}$

A Tabela 2 apresenta os 10 nós mais próximos da rede de simulação e a distância XOR calculada entre o *hash* de cada nó e o *hash* do valor-alvo. A tabela está ordenada pela distância, da menor para a maior. Conforme a especificação do Kademlia, o nó com a menor distância é o nó canonicamente responsável por armazenar o par (chave, valor) associado a este *hash*. Neste

caso, o *mastodonapp.uk.local* é o nó mais próximo, com uma distância de aproximadamente  $1,97 \times 10^{45}$ .

**Tabela 2 – Os 10 nós D-COLMEIA com menor distância XOR ao recurso <https://worstcontent.social/posts/abcd-1234> (hash  $1,11 \times 10^{48}$ ).**

| Nó                                  | Hash (Reduzido)    | Hash do Nó (Numérico) | Distância XOR ao Alvo |
|-------------------------------------|--------------------|-----------------------|-----------------------|
| <i>mastodonapp.uk.local</i>         | <i>c2cd...d9f4</i> | $1,11 \times 10^{48}$ | $1,97 \times 10^{45}$ |
| <i>occm.cc.local</i>                | <i>c610...0677</i> | $1,13 \times 10^{48}$ | $2,58 \times 10^{46}$ |
| <i>social.tchncs.de.local</i>       | <i>c5d7...c289</i> | $1,13 \times 10^{48}$ | $4,14 \times 10^{46}$ |
| <i>social.linux.pizza.local</i>     | <i>cb9e...80f0</i> | $1,16 \times 10^{48}$ | $5,16 \times 10^{46}$ |
| <i>planet.moe.local</i>             | <i>d3e5...60f4</i> | $1,21 \times 10^{48}$ | $9,96 \times 10^{46}$ |
| <i>mastodon.ie.local</i>            | <i>d124...e9ce</i> | $1,19 \times 10^{48}$ | $1,12 \times 10^{47}$ |
| <i>www.moe.local</i>                | <i>d6bd...6a38</i> | $1,23 \times 10^{48}$ | $1,15 \times 10^{47}$ |
| <i>mastodon.gamedev.place.local</i> | <i>d778...7837</i> | $1,23 \times 10^{48}$ | $1,25 \times 10^{47}$ |
| <i>mastodon.art.local</i>           | <i>da65...883f</i> | $1,25 \times 10^{48}$ | $1,42 \times 10^{47}$ |
| <i>mastodon.social.local</i>        | <i>de0f...db50</i> | $1,27 \times 10^{48}$ | $1,63 \times 10^{47}$ |

Fonte: Autoria própria.

Para validar o mecanismo de inscrição em um `dcolmeia:CommunityNoteGroup`, foram realizadas duas operações de `as:Follow` a partir de instâncias distintas. A atividade `as:Follow` é enviada para a *outbox* do usuário, utilizando a propriedade `as:context` para indicar a URL do recurso sendo moderado e a propriedade `as:object` para indicar a URL do `dcolmeia:CommunityNoteGroup` na instância de destino. A Figura 34 ilustra a atividade enviada.

**Figura 34 – Atividade `as:Follow` para um `dcolmeia:CommunityNoteGroup` enviada para a *outbox* do usuário.**

```

1 {
2   "@context": "https://www.w3.org/ns/activitystreams",
3   "type": "Follow",
4   "actor": "http://mstdn.party.local/users/admin",
5   "object": "http://mastodonapp.uk.local/communitynotes/c295...2c73",
6   "context": "https://worstcontent.social/posts/abcd-1234"
7 }

```

Fonte: Autoria própria.

Para poder simular o comportamento completo, o `dcolmeia:CommunityNoteGroup` é seguido 2 vezes.

A primeira operação de `Follow` é executada a partir de *mstdn.party.local*, seguindo o recurso <https://worstcontent.social/posts/abcd-1234>. Como o

`dcolmeia:CommunityNoteGroup` ainda não existe, ele é criado sob demanda pelo nó responsável. A Figura 35 apresenta o *log* desta operação.

**Figura 35 – Log da primeira operação Follow, com criação do `dcolmeia:CommunityNoteGroup` sob demanda.**

```

1 mstdn.party.local | INFO | FollowOutboxActivityProcessor: Processing Follow
  ↳ activity from user: http://mstdn.party.local/users/admin
2 mstdn.party.local | INFO | FollowOutboxActivityProcessor: Discovering target
  ↳ instance via WebFinger: mastodonapp.uk.local:80
3 mastodonapp.uk.local | INFO | WebfingerController: Found resource:
  ↳ acct:dcolmeia@mastodonapp.uk.local
4 mstdn.party.local | INFO | FollowOutboxActivityProcessor: Target instance is a
  ↳ valid dcolmeia node: http://mastodonapp.uk.local
5 mstdn.party.local | INFO | RoutingTable: Adding node http://mastodonapp.uk.local
  ↳ to bucket 158
6 mstdn.party.local | INFO | FollowOutboxActivityProcessor: Resolved DHT
  ↳ responsible node for Follow: https://worstcontent.social/posts/abcd-1234 ->
  ↳ http://mastodonapp.uk.local/communitynotes/c295...1d2c73
7 mstdn.party.local | INFO | ActivityPubService: Sending activity to group inbox:
  ↳ http://mastodonapp.uk.local/communitynotes/c295...1d2c73/inbox
8 mastodonapp.uk.local | INFO | GroupInboxService: Processing activity type
  ↳ 'as:Follow' for group 'c295...1d2c73'
9 mastodonapp.uk.local | INFO | CommunityNoteGroupService: Created new
  ↳ CommunityNoteGroup: c295...1d2c73 for target:
  ↳ https://worstcontent.social/posts/abcd-1234
10 mastodonapp.uk.local | INFO | GroupInboxService: Handling Follow for group:
  ↳ c295...1d2c73
11 mastodonapp.uk.local | INFO | CommunityNoteGroupService: Adding follower
  ↳ http://mstdn.party.local/users/admin to group c295...1d2c73
12 mastodonapp.uk.local | INFO | GroupInboxService: Sent Accept activity to
  ↳ http://mstdn.party.local/users/admin for group c295...1d2c73
13 mastodonapp.uk.local | INFO | GroupInboxService: User
  ↳ http://mstdn.party.local/users/admin is now following group c295...1d2c73
14 mstdn.party.local | INFO | AcceptInboxActivityProcessor: Processing Accept
  ↳ activity for user: http://mstdn.party.local/users/admin
15 mstdn.party.local | INFO | UserService: Confirm follower user:
  ↳ http://mstdn.party.local/users/admin

```

**Fonte: Autoria própria.**

O *log* da Figura 35 detalha a operação completa de Follow com criação do grupo sob demanda.

**Etapas 1: Descoberta da Instância de Destino** A instância *mstdn.party.local* recebe a atividade `as:Follow` na *outbox* do usuário. A partir da URL presente na propriedade `as:object`, o processador extrai o domínio da instância de destino (*mastodonapp.uk.local*) e realiza uma consulta WebFinger para verificar se esta implementa o D-COLMEIA. A resposta confirma a existência do ator `dcolmeia:ModerationNode`, e a instância é adicionada à tabela de roteamento local (*bucket 158*);

**Etapas 2: Resolução DHT e Envio do Follow** Com a instância de destino na tabela de rote-

amento, o processador consulta o nó mais próximo do *hash* do recurso moderado (<https://worstcontent.social/posts/abcd-1234>) e determina que *mastodonapp.uk.local* é o nó responsável. A atividade *as:Follow* é então enviada diretamente para a *inbox* do *dcolmeia:CommunityNoteGroup* correspondente;

**Etapa 3: Criação do Grupo e Aceitação do Seguidor** A instância *mastodonapp.uk.local* recebe a atividade *as:Follow*. Como o *dcolmeia:CommunityNoteGroup* ainda não existe, ele é criado sob demanda a partir da URL do recurso informada na propriedade *as:context*. O *hash* SHA-1 da URL é validado contra o identificador do grupo na URL, garantindo consistência. O usuário é adicionado à coleção de seguidores do grupo e uma atividade *as:Accept* é enviada de volta;

**Etapa 4: Confirmação do Follow** A instância *mstdn.party.local* recebe a atividade *as:Accept* e confirma a relação de seguidor. A partir deste momento, o usuário receberá notificações via *as:Announce* sempre que novas *dcolmeia:CommunityNote* forem adicionadas ao grupo.

A segunda operação de Follow é executada a partir de *mastodon.art.local*, para o mesmo recurso. Como o *dcolmeia:CommunityNoteGroup* já foi criado pela operação anterior, esta operação apenas adiciona o novo seguidor. A Figura 36 apresenta o *log*.

O *log* da Figura 36 apresenta uma operação de Follow análoga à primeira, porém com uma diferença significativa: o *dcolmeia:CommunityNoteGroup* já existe em *mastodonapp.uk.local*, criado pela operação anterior. Assim, o fluxo segue as mesmas etapas de descoberta via WebFinger, resolução DHT e envio do Follow, mas ao processar a atividade, o grupo é encontrado diretamente sem necessidade de criação. O segundo seguidor, *mastodon.art.local/users/admin*, é adicionado à coleção de seguidores e recebe o *as:Accept* de confirmação. Nota-se que a instância *mastodon.art.local* é posicionada no *bucket* 156 da tabela de roteamento, diferente do *bucket* 158 de *mstdn.party.local*, refletindo as distâncias XOR distintas entre os *hashes* das instâncias. A partir deste ponto, ambos os seguidores receberão notificações via *as:Announce* sobre novas *dcolmeia:CommunityNote* adicionadas ao grupo.

A criação de uma nota de moderação é iniciada por uma requisição no padrão *Activity-Pub* ao *outbox* do usuário que está enviando a nota na instância de origem. O *outbox* espera uma mensagem conforme ilustrado na Figura 37. A chamada a este *endpoint* dispara o processo de armazenamento distribuído do D-COLMEIA, permitindo a análise de seu fluxo de execução

**Figura 36 – Log da segunda operação Follow, com grupo já existente.**

```

1 mastodon.art.local | INFO | FollowOutboxActivityProcessor: Processing Follow
  ↳ activity from user: http://mastodon.art.local/users/admin
2 mastodon.art.local | INFO | FollowOutboxActivityProcessor: Discovering target
  ↳ instance via WebFinger: mastodonapp.uk.local:80
3 mastodonapp.uk.local | INFO | WebfingerController: Found resource:
  ↳ acct:dcormeia@mastodonapp.uk.local
4 mastodon.art.local | INFO | FollowOutboxActivityProcessor: Target instance is a
  ↳ valid dcormeia node: http://mastodonapp.uk.local
5 mastodon.art.local | INFO | RoutingTable: Adding node http://mastodonapp.uk.local
  ↳ to bucket 156
6 mastodon.art.local | INFO | FollowOutboxActivityProcessor: Resolved DHT
  ↳ responsible node for Follow: https://worstcontent.social/posts/abcd-1234 ->
  ↳ http://mastodonapp.uk.local/communitynotes/c295...1d2c73
7 mastodon.art.local | INFO | ActivityPubService: Sending activity to group inbox:
  ↳ http://mastodonapp.uk.local/communitynotes/c295...1d2c73/inbox
8 mastodonapp.uk.local | INFO | GroupInboxService: Processing activity type
  ↳ 'as:Follow' for group 'c295...1d2c73'
9 mastodonapp.uk.local | INFO | GroupInboxService: Handling Follow for group:
  ↳ c295...1d2c73
10 mastodonapp.uk.local | INFO | CommunityNoteGroupService: Adding follower
  ↳ http://mastodon.art.local/users/admin to group c295...1d2c73
11 mastodonapp.uk.local | INFO | GroupInboxService: Sent Accept activity to
  ↳ http://mastodon.art.local/users/admin for group c295...1d2c73
12 mastodonapp.uk.local | INFO | GroupInboxService: User
  ↳ http://mastodon.art.local/users/admin is now following group c295...1d2c73
13 mastodon.art.local | INFO | AcceptInboxActivityProcessor: Processing Accept
  ↳ activity for user: http://mastodon.art.local/users/admin
14 mastodon.art.local | INFO | UserService: Confirm follower user:
  ↳ http://mastodon.art.local/users/admin

```

**Fonte: Autoria própria.**

por meio dos *logs* do protótipo. Esta validação considera que as operações de Follow descritas na seção anterior já foram realizadas, de modo que o *dcormeia:CommunityNoteGroup* possui dois seguidores: *mstdn.party.local/users/admin* e *mastodon.art.local/users/admin*.

O log apresentado na Figura 38 detalha um fluxo completo de comunicação no D-COLMEIA, ilustrando uma operação de armazenamento (STORE) com múltiplos saltos e notificação de seguidores. O processo envolve cinco instâncias: *social.vivaldi.net.local* como iniciador da requisição, *dcormeia.social.local* como nó intermediário na rota, *mastodonapp.uk.local* como o destino final responsável pelo armazenamento, e *mastodon.art.local* e *mstdn.party.local* como seguidores do grupo que recebem a notificação.

**Etapas 1: Iniciação e Primeira Consulta** O fluxo se inicia em *social.vivaldi.net.local*, que recebe uma requisição para moderar o recurso <https://worstcontent.social/posts/abcd-1234>. A instância calcula o *hash* SHA-1 do recurso, consulta sua tabela de roteamento e identifica *dcormeia.social.local* como o nó mais próximo. Uma atividade *as:Add* contendo a *dcormeia:CommunityNote* é enviada para a *inbox* do

**Figura 37 – Mensagem *POST* para a *outbox ActivityPub* <http://social.vivaldi.net.local/users/admin/outbox>**

```

1  {
2    "@context": [
3      "https://www.w3.org/ns/activitystreams",
4      {
5        "dcolmeia": "https://dcolmeia.social/ns#"
6      }
7    ],
8    "type": "Create",
9    "actor": "http://social.vivaldi.net.local/users/admin",
10   "object": {
11     "type": "dcolmeia:CommunityNote",
12     "content": "This article contains misleading information. The study cited was ret
13     "dcolmeia:category": "dcolmeia:disturbingContent",
14     "inReplyTo": "https://worstcontent.social/posts/abcd-1234"
15   }
16 }

```

**Fonte:** Autoria própria.

`dcolmeia:CommunityNoteGroup` correspondente, acompanhada de um identificador de transação (`dcolmeia:inTransaction`) para rastreamento;

**Etapa 2: Resposta Intermediária e Redirecionamento** A instância *dcolmeia.social.local* recebe a requisição e processa a atividade `as:Add`. Ao consultar sua tabela de roteamento, identifica que não é o nó responsável pelo *hash* alvo e descobre um nó mais próximo: *mastodonapp.uk.local*. Consequentemente, responde ao iniciador com uma atividade `as:Reject`, cujo objeto contém um `dcolmeia:NextNode` apontando para *mastodonapp.uk.local*, instruindo o iniciador a continuar a busca iterativa. A resposta mantém o mesmo identificador de transação;

**Etapa 3: Segunda Iteração e Armazenamento** O iniciador, *social.vivaldi.net.local*, recebe a resposta de redirecionamento, aprende o endereço de *mastodonapp.uk.local* (adicionando-o ao *bucket* 159 de sua tabela de roteamento) e reenvia a atividade `as:Add` para a *inbox* do grupo neste novo nó. A instância *mastodonapp.uk.local* confirma que ela própria é o nó mais próximo do *hash* alvo, cria e armazena a `dcolmeia:CommunityNote` no `dcolmeia:CommunityNoteGroup`. Em seguida, envia uma atividade `as:Accept` de volta ao iniciador, confirmando o armazenamento com sucesso;

**Etapa 4: Notificação dos Seguidores** Após o armazenamento e a confirmação ao autor, o

**Figura 38 – Log resumido da operação STORE no D-COLMEIA, com múltiplos saltos e notificação de seguidores.**

```

1  social.vivaldi.net.local | INFO | CreateOutboxActivityProcessor: Processing Create
   ↳ activity from user: http://social.vivaldi.net.local/users/admin
2  social.vivaldi.net.local | INFO | CreateOutboxActivityProcessor: Sending Add to
   ↳ CommunityNoteGroup on DHT node: http://dcolmeia.social.local
3  social.vivaldi.net.local | INFO | ActivityPubService: Sending activity to group
   ↳ inbox: http://dcolmeia.social.local/communitynotes/c295...1d2c73/inbox
4  dcolmeia.social.local    | INFO | GroupInboxService: Processing activity type
   ↳ 'as:Add' for group 'c295...1d2c73'
5  dcolmeia.social.local    | INFO | GroupInboxService: This node is NOT responsible
   ↳ for group c295...1d2c73. Nearest: http://mastodonapp.uk.local
6  social.vivaldi.net.local | INFO | RejectInboxActivityProcessor: Re-routing Add to
   ↳ next node: http://mastodonapp.uk.local/users/dcolmeia
7  social.vivaldi.net.local | INFO | RoutingTable: Adding node
   ↳ http://mastodonapp.uk.local to bucket 159
8  social.vivaldi.net.local | INFO | ActivityPubService: Sending activity to group
   ↳ inbox: http://mastodonapp.uk.local/communitynotes/c295...1d2c73/inbox
9  mastodonapp.uk.local     | INFO | GroupInboxService: Processing activity type
   ↳ 'as:Add' for group 'c295...1d2c73'
10 mastodonapp.uk.local     | INFO | GroupInboxService: This node is responsible for
   ↳ group c295...1d2c73. Creating note.
11 mastodonapp.uk.local     | INFO | CommunityNoteService: Creating community note:
   ↳ {..., @type=dcolmeia:CommunityNote, content=This article contains misleading
   ↳ information. The study cited was retracted in 2023.,
   ↳ inReplyTo=https://worstcontent.social/posts/abcd-1234,
   ↳ category=dcolmeia:disturbingContent}
12 mastodonapp.uk.local     | INFO | GroupInboxService: CommunityNote
   ↳ http://social.vivaldi.net.local/moderate/6224...9d42 added to group
   ↳ c295...1d2c73
13 mastodonapp.uk.local     | INFO | GroupInboxService: Sent Accept to
   ↳ http://social.vivaldi.net.local/users/admin with transaction
   ↳ http://social.vivaldi.net.local/transactions/ad8f...f4a7
14 social.vivaldi.net.local | INFO | AcceptInboxActivityProcessor: Accepted community
   ↳ note for user: http://social.vivaldi.net.local/users/admin
15 mastodonapp.uk.local     | INFO | GroupInboxService: Announcing to 2 followers of
   ↳ group c295...1d2c73
16 mastodon.art.local       | INFO | AnnounceInboxActivityProcessor: Received Announce
   ↳ from http://mastodonapp.uk.local/communitynotes/c295...1d2c73 with embedded
   ↳ object: type=dcolmeia:CommunityNote,
   ↳ id=http://social.vivaldi.net.local/moderate/6224...9d42, content=This article
   ↳ contains misleading information. The study cited was retracted in 2023.
17 mastodonapp.uk.local     | INFO | GroupInboxService: Announced to follower:
   ↳ http://mastodon.art.local/users/admin
18 mstdn.party.local        | INFO | AnnounceInboxActivityProcessor: Received Announce
   ↳ from http://mastodonapp.uk.local/communitynotes/c295...1d2c73 with embedded
   ↳ object: type=dcolmeia:CommunityNote,
   ↳ id=http://social.vivaldi.net.local/moderate/6224...9d42, content=This article
   ↳ contains misleading information. The study cited was retracted in 2023.
19 mastodonapp.uk.local     | INFO | GroupInboxService: Announced to follower:
   ↳ http://mstdn.party.local/users/admin

```

**Fonte: Autoria própria.**

`dcolmeia:CommunityNoteGroup` em *mastodonapp.uk.local* notifica seus dois seguidores. Para cada seguidor, é enviada uma atividade `as:Announce` contendo a `dcolmeia:CommunityNote` recém-criada como objeto embutido. A instância *mastodon.art.local* recebe o `as:Announce` e registra a notificação, identificando o objeto

como uma `dcolmeia:CommunityNote` com o conteúdo da nota. Da mesma forma, `mstdn.party.local` recebe e processa a mesma notificação. Este mecanismo de *push*, baseado no padrão `as:Follow/as:Announce` do ActivityPub, garante que todos os seguidores do grupo sejam informados em tempo real sobre novas contribuições de moderação, sem necessidade de consultas periódicas.

Este fluxo demonstra a integração completa entre o mecanismo de armazenamento distribuído (STORE) e o sistema de notificação de seguidores do D-COLMEIA. O armazenamento iterativo percorre a rede DHT até encontrar o nó responsável, que ao aceitar a nota, automaticamente propaga a informação para todos os seguidores interessados. Desta forma, instâncias que manifestaram interesse na moderação de um recurso via `as:Follow` são mantidas atualizadas de forma proativa, complementando as consultas reativas via `FIND_VALUE`.

#### 4.3 BUSCA DE UM *DCOLMEIA:COMMUNITYNOTEGROUP*

Para possibilitar simular a busca de um `dcolmeia:CommunityNoteGroup`, contendo os `dcolmeia:CommunityNote` relacionados a um recurso, foi utilizado o *endpoint* padrão do ActivityPub `POST /users/{username}/outbox`, que recebe uma atividade do tipo `dcolmeia:FindValue` como a mostrada na Figura 39. A atividade é enviada para a *outbox* do usuário, que inicia a operação de busca iterativa na rede DHT pelo `dcolmeia:CommunityNoteGroup` relacionado ao recurso informado na propriedade `as:target`. Como a comunicação é assíncrona, este *endpoint* não retorna o resultado diretamente, apenas inicia o processo de busca. A validação desta seção considera que a *dcolmeia:CommunityNote* descrita na Seção 4.2 foi criada anteriormente.

Para esta validação, a busca é executada a partir de *alpha.argyle.social.local*, uma instância que não participou das operações anteriores, de modo que sua tabela de roteamento ainda não contém uma referência direta ao nó responsável, forçando a execução de uma busca iterativa com múltiplos saltos.

O *log* apresentado na Figura 40 detalha uma operação completa de busca por valor (`FIND_VALUE`) no D-COLMEIA. Este fluxo demonstra o mecanismo de busca iterativa em múltiplos saltos: o nó iniciador, *alpha.argyle.social.local*, primeiro contata o nó de referência *dcolmeia.social.local*, que o redireciona para o nó final, *mastodonapp.uk.local*, que detém a informação.

**Figura 39 – Atividade `dcolmeia:FindValue` enviada para a *outbox* do usuário.**

```

1  {
2    "@context": [
3      "https://www.w3.org/ns/activitystreams",
4      {
5        "dcolmeia": "https://dcolmeia.social/ns#"
6      }
7    ],
8    "type": "dcolmeia:FindValue",
9    "actor": "http://alpha.argyle.social.local/users/admin",
10   "target": "https://worstcontent.social/posts/abcd-1234"
11 }

```

**Fonte:** Autoria própria.

**Etapa 1: Iniciação e Primeira Consulta** O fluxo se inicia em *alpha.argyle.social.local*, que busca o `dcolmeia:CommunityNoteGroup` associado ao recurso *https://worstcontent.social/posts/abcd-1234*. Ao consultar sua tabela de roteamento local, a instância identifica *dcolmeia.social.local* como o contato mais promissor e envia a ele uma atividade `dcolmeia:FindValue` para a *inbox* de seu *ModerationNode*;

**Etapa 2: Resposta Negativa e Redirecionamento** A instância *dcolmeia.social.local* recebe a requisição e processa a atividade `dcolmeia:FindValue`. Ao consultar sua base de dados local, não encontra o `dcolmeia:CommunityNoteGroup` solicitado. Por não possuir o valor, ela age como em uma operação `FIND_NODE`: consulta sua própria tabela de roteamento, encontra um nó mais próximo do *hash* alvo, *mastodonapp.uk.local*, e responde ao iniciador com um `dcolmeia:NextNode`, instruindo-o a continuar a busca iterativa;

**Etapa 3: Segunda Iteração e Descoberta do Valor** O iniciador, *alpha.argyle.social.local*, recebe a instrução para contatar o próximo nó. Ele atualiza sua tabela de roteamento com o novo contato e reenvia a mesma requisição `dcolmeia:FindValue` para a *inbox* do `dcolmeia:ModerationNode` de *mastodonapp.uk.local*. Ao receber a mensagem, *mastodonapp.uk.local* também atualiza sua tabela com os dados de *alpha.argyle.social.local* (*bucket* 159) e, ao consultar sua base local, localiza o `dcolmeia:CommunityNoteGroup` com uma `dcolmeia:CommunityNote` associada ao recurso buscado;

**Figura 40 – Log da operação FIND\_VALUE iterativa no D-COLMEIA, com múltiplos saltos.**

```

1  alpha.argyle.social.local | INFO | DcolmeiaFindOutboxActivityProcessor: Sending
   ↳ find moderation request to DHT node: http://dcolmeia.social.local
2  alpha.argyle.social.local | INFO | ActivityPubService: Sending user data to Inbox:
   ↳ http://dcolmeia.social.local/users/dcolmeia | Data:
   ↳ {@type=dcolmeia:ModerationNode, ...}
3  dcolmeia.social.local     | INFO | ModerationNodeInboxService: Processing activity
   ↳ type 'dcolmeia:FindValue' for ModerationNode
4  dcolmeia.social.local     | INFO | DcolmeiaFindValueInboxActivityProcessor: No
   ↳ community notes found locally for target:
   ↳ https://worstcontent.social/posts/abcd-1234
5  dcolmeia.social.local     | INFO | DcolmeiaFindValueInboxActivityProcessor: Sent
   ↳ NextNode response pointing to: http://mastodonapp.uk.local
6  alpha.argyle.social.local | INFO | FindValueResponseInboxActivityProcessor:
   ↳ Continuing FindValue search to next node: http://mastodonapp.uk.local for
   ↳ target: https://worstcontent.social/posts/abcd-1234
7  alpha.argyle.social.local | INFO | ActivityPubService: Sending user data to Inbox:
   ↳ http://mastodonapp.uk.local/users/dcolmeia | Data:
   ↳ {@type=dcolmeia:ModerationNode, ...}
8  mastodonapp.uk.local     | INFO | ModerationNodeInboxService: Processing activity
   ↳ type 'dcolmeia:FindValue' for ModerationNode
9  mastodonapp.uk.local     | INFO | RoutingTable: Adding node
   ↳ http://alpha.argyle.social.local to bucket 159
10 mastodonapp.uk.local     | INFO | DcolmeiaFindValueInboxActivityProcessor: Found 1
   ↳ community notes for target: https://worstcontent.social/posts/abcd-1234
11 mastodonapp.uk.local     | INFO | ActivityPubService: Sending user data to Inbox:
   ↳ http://alpha.argyle.social.local/users/dcolmeia | Data:
   ↳ {@type=dcolmeia:ModerationNode, ...}
12 alpha.argyle.social.local | INFO | ModerationNodeInboxService: Processing activity
   ↳ type 'dcolmeia:FindValueResponse' for ModerationNode
13 alpha.argyle.social.local | INFO | FindValueResponseInboxActivityProcessor:
   ↳ Processing CommunityNoteGroup result: {..., @type=dcolmeia:CommunityNoteGroup,
   ↳ items=[{@type=dcolmeia:CommunityNote, ...}]}
14 alpha.argyle.social.local | INFO | FindValueResponseInboxActivityProcessor:
   ↳ Received community notes for target:
   ↳ https://worstcontent.social/posts/abcd-1234 (group:
   ↳ http://mastodonapp.uk.local/communitynotes/c295...1d2c73)

```

**Fonte: Autoria própria.**

**Etapa 4: Resposta com o Valor e Finalização** Tendo localizado a informação, *mastodonapp.uk.local* envia uma atividade `dcolmeia:FindValueResponse` de volta para *alpha.argyle.social.local*. O conteúdo desta resposta é o `dcolmeia:CommunityNoteGroup` completo, incluindo as `CommunityNotes` armazenadas com seus respectivos metadados, como autor, data de publicação, categoria de moderação e contadores de *likes* e *dislikes*. O iniciador recebe e processa este objeto, finalizando a busca com sucesso.

Em contraste, uma instância que já tivesse interagido com o nó responsável, como as que seguiram o `dcolmeia:CommunityNoteGroup` na Seção 4.2 e passaram a manter *mastodonapp.uk.local* em sua tabela de roteamento, resolveria a mesma busca em um único

salto. Esse comportamento ilustra a convergência progressiva das tabelas de roteamento, uma propriedade fundamental do Kademlia: à medida que os nós interagem na rede, suas tabelas se tornam mais precisas, reduzindo o número de saltos necessários para localizar recursos.

#### 4.4 DISCUSSÃO DOS RESULTADOS

Os cenários apresentados nas Seções 4.2 e 4.3 demonstraram o funcionamento das operações `STORE`, `FIND_NODE` e `FIND_VALUE` adaptadas ao modelo de comunicação assíncrono do ActivityPub. A arquitetura demonstrou ser flexível, permitindo não apenas a moderação de conteúdo do fediverso, mas também de recursos externos identificados por URLs, como domínios completos. Dessa forma, além da criação de notas de comunidade para conteúdos específicos do fediverso, o sistema permite a moderação de *websites* e instâncias inteiras.

Os *logs* da Seção 4.2 evidenciam a integração da tipologia NEON (CHARLES *et al.*, 2022), apresentada no Quadro 3, por meio do atributo `dcolmeia:category` na ontologia da `dcolmeia:CommunityNote`. No cenário demonstrado, a nota criada recebeu a categoria *dcolmeia:disturbingContent*, conferindo significado semântico à informação de moderação. Em conjunto com as atividades `as:Like` e `as:Dislike`, já suportadas pelo protótipo, essa categorização possibilita a construção de um *ranking* colaborativo de tipos de alertas por conteúdo, oferecendo uma informação mais rica e padronizada aos administradores de instâncias.

Um aspecto importante para a aplicação de conceitos de inteligência coletiva na moderação é o modelo de disseminação de informações. No modelo base do D-COLMEIA, a instância que deseja obter informações de moderação deve iniciar uma busca via `FIND_VALUE` (modelo *pull*). Para complementar essa abordagem, o `dcolmeia:CommunityNoteGroup` foi projetado como um ator ActivityPub, conforme documentado na FEP-1b12 (Fediverse Enhancement Proposals, 2024b), que descreve o uso de atores para fins específicos no fediverso. Como demonstrado nos *logs* das seções anteriores, o `dcolmeia:CommunityNoteGroup` possui sua própria *inbox*, recebe atividades `as:Add` e responde com `as:Accept` ou `as:Reject`. Essa modelagem como ator possibilita que outras instâncias sigam o grupo via `as:Follow` para receberem atualizações automáticas (modelo *push*), sem a necessidade de realizar buscas repetidas. Conforme validado na Seção 4.2, o mecanismo de Follow com criação sob demanda do grupo permite que usuários manifestem interesse na moderação de um recurso antes mesmo da existência de qualquer nota, e que ao armazenar uma nova `dcolmeia:CommunityNote`, o grupo notifica automaticamente todos os seguidores via `as:Announce`, com a nota embutida

como objeto da atividade. Nos *logs* apresentados, os dois seguidores, *mastodon.art.local* e *mstdn.party.local*, receberam a notificação contendo o tipo, identificador e conteúdo da nota criada, confirmando o funcionamento do modelo *push*. Essa combinação dos modelos *pull* e *push* oferece flexibilidade às implementações: instâncias podem optar por buscar informações sob demanda, receber atualizações proativamente, ou combinar ambas as estratégias conforme suas necessidades.

Em relação ao rastreamento de contexto nas mensagens assíncronas, o protótipo implementa o mecanismo `dcolmeia:inTransaction`, presente no fluxo `STORE` apresentado na Figura 38, visível na confirmação `as:Accept` (omitido nos demais *logs* resumidos por concisão). Esse identificador único de transação acompanha a mensagem ao longo de todo o fluxo de comunicação, desde a requisição inicial até a confirmação final, permitindo que os nós intermediários e o nó de destino associem corretamente cada mensagem ao seu contexto original. Essa abordagem simplifica o rastreamento em comparação com a alternativa de reutilizar diversos atributos padrão do `ActivityStreams` (`as:origin`, `as:target` ou `as:id`) para controlar o fluxo.

Da mesma forma, a separação das operações de busca em atividades distintas, `dcolmeia:FindNode` e `dcolmeia:FindValue`, conforme demonstrado na Seção 4.3, elimina a ambiguidade que existiria caso uma única atividade genérica fosse utilizada para ambas as finalidades. Com atividades separadas, a implementação não precisa inferir a intenção da busca a partir de padrões na mensagem, reduzindo a complexidade do processamento e evitando falhas de interpretação.

## 5 CONCLUSÃO

Este trabalho abordou o desafio da moderação de conteúdo em ecossistemas descentralizados como o fediverso, onde a ausência de uma autoridade central exige novas abordagens para a colaboração entre instâncias. Para isso, foi proposta a arquitetura D-COLMEIA, uma extensão ao protocolo ActivityPub que organiza uma rede de sobreposição baseada nos princípios da DHT Kademlia, voltada ao armazenamento e à recuperação distribuída de informações de moderação.

A principal contribuição deste trabalho é a proposição de um modelo que viabiliza a moderação colaborativa de conteúdo no fediverso, permitindo que usuários e instâncias independentes contribuam coletivamente com informações de moderação sobre recursos da rede, aplicando conceitos de inteligência coletiva e *crowdsourcing* em um contexto descentralizado. Para viabilizar essa colaboração, as operações fundamentais do Kademlia foram adaptadas para o paradigma de comunicação assíncrono do ActivityPub, permitindo que instâncias participem de uma DHT sem a necessidade de protocolos ou infraestruturas externas. Uma ontologia OWL que estende o vocabulário ActivityStreams 2.0 formaliza os tipos de atividades, atores e objetos necessários ao funcionamento do protocolo. A utilização do algoritmo de roteamento do Kademlia permite que a distribuição do conteúdo de moderação entre os nós da rede ocorra de forma determinística e transparente, de modo que o nó responsável por armazenar a informação seja definido por critérios objetivos derivados da métrica XOR, sem depender de decisões arbitrárias ou de uma autoridade central. A viabilidade da abordagem foi demonstrada por meio de um protótipo funcional, validado em um ambiente controlado com 64 nós, cujos resultados confirmaram o comportamento esperado para as operações de armazenamento, busca e notificação, conforme discutido na Seção 4.4. Dessa forma, os objetivos específicos definidos na introdução foram atendidos: a definição de um padrão semântico de mensagens sustentado por uma ontologia OWL, o desenvolvimento e a documentação dos algoritmos de distribuição e busca adaptados ao ActivityPub, a implementação de um protótipo funcional integrado a plataformas federadas, a análise do funcionamento da rede DHT de moderação e a validação experimental das operações do protocolo. O cumprimento desses objetivos específicos concretiza o objetivo geral do trabalho, que era propor uma extensão ao protocolo ActivityPub voltada à moderação de conteúdo distribuída e colaborativa.

Os resultados obtidos com o protótipo de 64 nós demonstram que o D-COLMEIA é uma abordagem funcionalmente viável para a moderação colaborativa de conteúdo no fediverso,

sem comprometer seus princípios de descentralização, embora ainda demande melhorias e refinamentos para sua utilização em ambientes de produção. Ao operar nativamente sobre o ActivityPub, a abordagem proposta reduz a barreira de adoção em comparação com alternativas que dependem de protocolos ou infraestruturas externas. Embora a validação apresentada não permita conclusões sobre o comportamento em redes de larga escala, a pesquisa demonstra a viabilidade conceitual e funcional do caminho proposto, abrindo espaço para as investigações futuras descritas a seguir.

## 5.1 TRABALHOS FUTUROS

O ambiente de testes com 64 nós, embora suficiente para a validação funcional, não permite conclusões sobre o comportamento do sistema em redes de larga escala, e métricas como a contagem de saltos na DHT resultariam em valores trivialmente baixos nessa configuração. Para ampliar o escopo da validação, é necessário viabilizar testes com um número significativamente maior de nós, seja por meio do desenvolvimento de um protótipo com menor consumo de recursos ou da utilização de uma infraestrutura de *hardware* mais robusta. Uma análise comparativa de desempenho com soluções equivalentes também não seria viável neste momento, dada a inexistência de outra solução de moderação distribuída baseada em ActivityPub e que possua as mesmas características que o D-COLMEIA.

Complementarmente, o ambiente de testes atual não reproduz condições típicas de redes reais, como latência variável, perda de pacotes e partições de rede. A adoção de ferramentas de emulação como Shadow (<https://shadow.github.io>), Mininet (<https://mininet.org>) ou Containernet (<https://containernet.github.io>) permitiria simular essas condições, oferecendo uma validação mais representativa do comportamento do protocolo.

Também deve ser investigada a compatibilidade do D-COLMEIA com os mecanismos de segurança já utilizados no fediverso. Implementações como o Mastodon utilizam requisições HTTP assinadas, tanto no formato *HTTP Signatures* quanto no padrão mais recente *HTTP Message Signatures (RFC 9421)*, para validar que as atividades recebidas foram de fato originadas pelo ator declarado (MASTODON, 2026). Além disso, *Linked Data Signatures* permitem anexar assinaturas criptográficas diretamente aos documentos JSON-LD, possibilitando a verificação de autoria mesmo quando a atividade é retransmitida por intermediários, como *relays*. Investigar como essas assinaturas interagem com as atividades do namespace `dcolmeia:` e com o encaminhamento de mensagens entre nós intermediários na DHT é essencial para garantir a

integridade e a autenticidade das operações de moderação distribuída.

Ainda no âmbito da segurança, pode ser investigada a utilização de *Decentralized Identifiers* (DIDs) (W3C, 2022) como identificadores padronizados para nós e recursos na rede D-COLMEIA, de forma que o identificador do recurso não fique vinculado ao domínio da instância.

Quanto ao roteamento, melhorias podem ser exploradas a partir da literatura existente, como as propostas do S/Kademlia (BAUMGART; MIES, 2007) para resistência a ataques *Sybil* e *Eclipse*. Nessa direção, a biblioteca *LibP2P*, que incorpora modificações do S/Kademlia em sua implementação de DHT (LibP2P Team, 2024), pode ser utilizada tanto para o fortalecimento do roteamento quanto para o estabelecimento de canais seguros de comunicação entre os nós.

Quanto à resiliência dos dados, é necessário definir estratégias de replicação em múltiplas instâncias e de rebalanceamento da DHT para cenários de entrada e saída de nós, investigação que pode se beneficiar da *FEP-1580* (Fediverse Enhancement Proposals, 2025), atualmente em elaboração, que propõe mecanismos de migração de atores entre instâncias utilizando o próprio ActivityPub.

O desenvolvimento dessas frentes permitirá evoluir o D-COLMEIA de um protótipo funcional para uma solução robusta de moderação colaborativa e distribuída, adequada à adoção em ambientes reais do fediverso.

## REFERÊNCIAS

- ABBING, Roel Roscam; DIEHM, Cade; WARRETH, Shahed. Decentralised social media. **Internet Policy Review**, v. 12, n. 1, 2023. ISSN 2197-6775.
- ANAOBI, Ishaku Hassan; RAMAN, Aravindh; CASTRO, Ignacio; ZIA, Haris Bin; IBOSIOLA, Damilola; TYSON, Gareth. Will Admins Cope? Decentralized Moderation in the Fediverse. *In: **Proceedings of the ACM Web Conference 2023***. Austin TX USA: ACM, 2023. p. 3109–3120. ISBN 978-1-4503-9416-1. Disponível em: <https://dl.acm.org/doi/10.1145/3543507.3583487>. Acesso em: 18 mar. 2025.
- ATLEE, Tom. Co-intelligence, collective intelligence, and conscious evolution. *In: TOVEY, Mark (Ed.). **Collective intelligence: creating a prosperous world at peace***. Oakton: Earth Intelligence Network, 2008. p. 37–52. ISBN 978-0-9715661-6-3.
- BADIS, L.; AMAD, M.; AÏSSANI, D.; ABBAR, S. P2PCF: A collaborative filtering based recommender system for peer to peer social networks. **Journal of High Speed Networks**, v. 27, n. 1, p. 13–31, 2021. ISSN 0926-6801.
- BAUMGART, Ingmar; MIES, Sebastian. S/Kademlia: A practicable approach towards secure key-based routing. *In: **2007 International Conference on Parallel and Distributed Systems***. Hsinchu, Taiwan: IEEE, 2007. p. 1–8.
- BONO, C.; CAVA, L. La; LUCERI, L.; PIERRI, F. An Exploration of Decentralized Moderation on Mastodon. *In: Aiello L.C.; Mejova Y.; Seneviratne O.; Sun J.; Kaiser S.; Staab S. (Ed.). **Proc. ACM Web Sci. Conf., WebSci***. Stuttgart, Germany: Association for Computing Machinery, Inc, 2024. p. 53–58. ISBN 979-8-4007-0334-8. Disponível em: <https://dl.acm.org/doi/10.1145/3614419.3644016>. Acesso em: 12 maio 2026.
- BREMBS, B.; LENARDIC, A.; MURRAY-RUST, P.; CHAN, L.; IRAWAN, D.E. Mastodon over Mammon: towards publicly owned scholarly knowledge. **Royal Society Open Science**, v. 10, n. 7, 2023. ISSN 2054-5703.
- CAVA, L. La; MANDAGLIO, D.; TAGARELLI, A. Polarization in Decentralized Online Social Networks. *In: Aiello L.C.; Mejova Y.; Seneviratne O.; Sun J.; Kaiser S.; Staab S. (Ed.). **Proc. ACM Web Sci. Conf., WebSci***. Stuttgart, Germany: Association for Computing Machinery, Inc, 2024. p. 48–52. ISBN 979-8-4007-0334-8.
- CHARLES, Ashleigh; HARE-DUKE, Laurie; NUDDS, Hannah; FRANKLIN, Donna; LLEWELLYN-BEARDSLEY, Joy; RENNICK-EGGLESTONE, Stefan; GUST, Onni; NG, Fiona; EVANS, Elizabeth; KNOX, Emily; TOWNSEND, Ellen; YEO, Caroline; SLADE, Mike. Typology of content warnings and trigger warnings: Systematic

review. **PLOS ONE**, v. 17, n. 5, p. e0266722, 2022. ISSN 1932-6203. Disponível em: <https://dx.plos.org/10.1371/journal.pone.0266722>. Acesso em: 13 jul. 2025.

CHUAI, Yuwei; TIAN, Haoye; PRÖLLOCHS, Nicolas; LENZINI, Gabriele. Did the Roll-Out of Community Notes Reduce Engagement With Misinformation on X/Twitter? **Proceedings of the ACM on Human-Computer Interaction**, v. 8, n. CSCW2, p. 1–52, nov. 2024. ISSN 2573-0142. Disponível em: <https://dl.acm.org/doi/10.1145/3686967>. Acesso em: 15 fev. 2025.

COULOURIS, G.; DOLLIMORE, J.; KINDBERG, T.; BLAIR, G. **Sistemas Distribuídos - 5ed: Conceitos e Projeto**. Porto Alegre: Bookman Editora, 2013. ISBN 9788582600542. Disponível em: <https://books.google.com.br/books?id=6WU3AgAAQBAJ>.

Docker Inc. **Docker Overview**. 2025. Disponível em: <https://docs.docker.com/get-started/docker-overview/>. Acesso em: 12 mar. 2026.

ERMOSHINA, Ksenia; MUSIANI, Francesca. Safer spaces by design? federated socio-technical architectures in content moderation. **Internet Policy Review**, v. 14, n. 1, p. 1–26, 03 2025. ISSN 2197-6775. Disponível em: <https://policyreview.info/articles/analysis/safer-spaces-design>. Acesso em: 5 abr. 2025.

ETHEREUM. **Node Discovery Protocol v5**. 2020. Disponível em: <https://github.com/ethereum/devp2p/blob/master/discv5/discv5.md>. Acesso em: 29 jul. 2024.

Fediverse Enhancement Proposals. **Fediverse Enhancement Proposals (FEP)**. 2024. Disponível em: <https://codeberg.org/fediverse/fep>. Acesso em: 2 ago. 2025.

Fediverse Enhancement Proposals. **FEP-1b12: Group federation**. Fediverse Enhancement Proposals, 2024. Disponível em: <https://codeberg.org/fediverse/fep/src/branch/main/fep/1b12/fep-1b12.md>. Acesso em: 2 ago. 2025.

Fediverse Enhancement Proposals. **FEP-1580: Move Actor Objects with a Migration Collection**. 2025. Disponível em: <https://codeberg.org/fediverse/fep/src/branch/main/fep/1580/fep-1580.md>. Acesso em: 4 out. 2025.

FRIEDL, P.; MORGAN, J. Decentralised content moderation. **Internet Policy Review**, v. 13, n. 2, 2024. ISSN 2197-6775.

GEHL, Robert W. Fediverse blocklists: Moderation in noncapitalist social media. **tripleC: Communication, Capitalism & Critique**, v. 23, n. 2, p. 278–294, 2025. Disponível em: <https://www.triple-c.at/index.php/tripleC/article/view/1556>. Acesso em: 3 maio 2026.

GUIDI, B.; MICHIEZI, A.; RICCI, L. Data persistence in decentralized social applications: The IPFS approach. In: **IEEE Annu. Consum. Commun. Netw. Conf., CCNC**. Las Vegas, NV, USA: Institute of Electrical and Electronics Engineers Inc., 2021. ISBN 978-1-7281-9794-4.

GUIDI, B.; MICHIEZI, A.; RICCI, L. Managing communities in decentralised social environments. **Peer-to-Peer Networking and Applications**, v. 15, n. 5, p. 2404–2429, 2022. ISSN 1936-6442.

HEVNER, Alan R.; MARCH, Salvatore T.; PARK, Jinsoo; RAM, Sudha. Design science in information systems research. **MIS Quarterly**, v. 28, n. 1, p. 75–105, 2004.

HITZLER, Pascal; KRÖTZSCH, Markus; PARSIA, Bijan; PATEL-SCHNEIDER, Peter F.; RUDOLPH, Sebastian. **OWL 2 Web Ontology Language Primer (Second Edition)**. 2012. W3C Recommendation. Disponível em: <https://www.w3.org/TR/owl2-primer/>. Acesso em: 14 dez. 2025.

IPFS. **Distributed Hash Tables (DHT) | IPFS Docs**. 2024. Disponível em: <https://docs.ipfs.tech/concepts/dht/>. Acesso em: 29 jul. 2024.

ISOTANI, Seiji; BITTENCOURT, Ig Ibert. **Dados abertos conectados**. São Paulo: Novatec Editora, 2022. ISBN 978-85-7522-449-6.

JHAVER, Shagun; GHOSHAL, Sucheta; BRUCKMAN, Amy; GILBERT, Eric. Online harassment and content moderation: The case of blocklists. **ACM Transactions on Computer-Human Interaction**, v. 25, n. 2, p. 1–33, 03 2018.

JONES, Paul; SALGUEIRO, Gonzalo; JONES, Michael B; SMARR, Joseph. **RFC 7033: WebFinger**. 2013. Disponível em: <https://datatracker.ietf.org/doc/html/rfc7033>. Acesso em: 6 set. 2013.

Lemmy Project. **Lemmy - A decentralised discussion platform for communities**. 2026. Disponível em: <https://join-lemmy.org/?lang=en>. Acesso em: 1 abr. 2026.

LibP2P Team. **libp2p Kademlia DHT specification**. 2024. Disponível em: <https://github.com/libp2p/specs/blob/master/kad-dht/README.md>. Acesso em: 7 set. 2024.

LOEWENSTERN, Andrew. **DHT Protocol (BitTorrent)**. 2008. Disponível em: [https://www.bittorrent.org/beps/bep\\_0005.html](https://www.bittorrent.org/beps/bep_0005.html). Acesso em: 29 jul. 2024.

MARX, J.; CHEONG, M. Decentralised Social Media: Scoping Review and Future Research Directions. In: **Australas. Conf. Inf. Syst., ACIS**. Wellington, New Zealand: Association for Information Systems, 2023. p. 1–15. Disponível em: <https://aisel.aisnet.org/acis2023/58>. Acesso em: 26 jun. 2026.

MASTODON. **mastodon/FEDERATION.md at main · mastodon/mastodon**. Mastodon, 2021. Disponível em: <https://github.com/mastodon/mastodon/blob/main/FEDERATION.md>. Acesso em: 12 ago. 2023.

MASTODON. **Report - Mastodon documentation**. 2025. Disponível em: <https://docs.joinmastodon.org/entities/Report/>. Acesso em: 20 set. 2025.

MASTODON. **Reporting profiles and posts (Flag) - ActivityPub - Mastodon documentation**. 2025. Disponível em: <https://docs.joinmastodon.org/spec/activitypub/#Flag>. Acesso em: 20 set. 2025.

MASTODON. **Security: Public key cryptography and supported signature schemes over HTTP and JSON-LD**. 2026. Disponível em: <https://docs.joinmastodon.org/spec/security/>. Acesso em: 4 abr. 2026.

MAYMOUNKOV, Petar; MAZIERES, David. Kademlia: A peer-to-peer information system based on the XOR metric. *In: International workshop on peer-to-peer systems*. Cambridge, MA, USA: Springer, 2002. p. 53–65.

MELDER, Erika; LERNER, Ada; DeVito, Michael Ann. "a blocklist is a boundary": Tensions between community protection and mutual aid on federated social networks. **Proceedings of the ACM on Human-Computer Interaction**, v. 9, n. 2, p. 1–30, 2025. ISSN 2573-0142. Disponível em: <https://dl.acm.org/doi/10.1145/3710919>. Acesso em: 25 abr. 2026.

META. **Introducing Threads: A New Way to Share With Text**. 2023. Disponível em: <https://about.fb.com/news/2023/07/introducing-threads-new-app-text>. Acesso em: 15 jul. 2023.

META. **Introducing Community Notes - Adding Context to Posts | Meta**. 2025. Disponível em: <https://www.meta.com/technologies/community-notes/>. Acesso em: 20 abr. 2025.

MORROW, Garrett; SWIRE-THOMPSON, Briony; POLNY, Jessica Montgomery; KOPEC, Matthew; WIHBAY, John P. The emerging science of content labeling: Contextualizing social media content moderation. **Journal of the Association for Information Science and Technology**, v. 73, n. 10, p. 1365–1386, 03 2022.

MOUALKIA, Y.; AMAD, M.; BAADACHE, A. Hierarchical and scalable peer-to-peer architecture for online social network. **Journal of King Saud University - Computer and Information Sciences**, v. 34, n. 10, p. 8623–8636, 2022. ISSN 1319-1578.

PEERTUBE. **PeerTube documentation | PeerTube documentation**. 2023. Disponível em: <https://docs.joinpeertube.org>. Acesso em: 25 jul. 2024.

PEFFERS, Ken; TUUNANEN, Tuure; ROTHENBERGER, Marcus A.; CHATTERJEE, Samir. A design science research methodology for information systems research. **Journal of Management Information Systems**, v. 24, n. 3, p. 45–77, 2007.

PIXELFED. **Pixelfed Documentation**. 2022. Disponível em: <https://docs.pixelfed.org/>. Acesso em: 12 ago. 2023.

PRODROMOU, E. **ActivityPub**. Sebastopol: O'Reilly Media, 2024. ISBN 9781098162719. Disponível em: <https://books.google.com.br/books?id=Q40jEQAAQBAJ>.

PRÖLLOCHS, Nicolas. Community-Based Fact-Checking on Twitter's Birdwatch Platform. **Proceedings of the International AAAI Conference on Web and Social Media**, v. 16, p. 794–805, maio 2022. ISSN 2334-0770. Disponível em: <https://ojs.aaai.org/index.php/ICWSM/article/view/19335>. Acesso em: 16 fev. 2025.

ROWSTRON, Antony; DRUSCHEL, Peter. Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems. In: GOOS, Gerhard; HARTMANIS, Juris; LEEUWEN, Jan Van; GUERRAOUI, Rachid (Ed.). **Middleware 2001**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, (Lecture Notes in Computer Science, v. 2218). p. 329–350. ISBN 978-3-540-42800-8. Disponível em: [http://link.springer.com/10.1007/3-540-45518-3\\_18](http://link.springer.com/10.1007/3-540-45518-3_18). Acesso em: 30 jul. 2024.

SABO, Eduard; GESTHUIZEN, Tim; BOUMA, Kelvin J. A.; KARASTOYANOVA, Dimka; RIVENI, Mirela. An analysis of mastodon adoption dynamics based on instance types. **Social Network Analysis and Mining**, v. 14, n. 1, p. 184, set. 2024. ISSN 1869-5469. Disponível em: <https://link.springer.com/10.1007/s13278-024-01341-7>. Acesso em: 14 set. 2024.

SAINT-ANDRE, Peter. **RFC 7565: The 'acct' URI Scheme**. 2015. Disponível em: <https://datatracker.ietf.org/doc/html/rfc7565>. Acesso em: 6 jul. 2025.

SALMINEN, Juho. Collective intelligence in humans: a literature review. **CoRR**, abs/1204.3401, 2012. Disponível em: <https://arxiv.org/abs/1204.3401>. Acesso em: 26 jun. 2026.

SALVE, A. De; MORI, P.; RICCI, L.; PIETRO, R. Di. Content privacy enforcement models in decentralized online social networks: State of play, solutions, limitations, and future directions. **Computer Communications**, v. 203, p. 199–225, 2023. ISSN 0140-3664.

STEEN, M. van; TANENBAUM, A.S. **Distributed Systems**. Amsterdam: Maarten van Steen, 2023. ISBN 9789081540643. Disponível em: [https://books.google.com.br/books?id=\\_yhb0AEACAAJ](https://books.google.com.br/books?id=_yhb0AEACAAJ).

STOICA, Ion; MORRIS, Robert; KARGER, David; KAASHOEK, M. Frans; BALAKRISHNAN, Hari. Chord: A scalable peer-to-peer lookup service for internet applications. **ACM SIGCOMM**

**Computer Communication Review**, v. 31, n. 4, p. 149–160, out. 2001. ISSN 0146-4833. Disponível em: <https://dl.acm.org/doi/10.1145/964723.383071>. Acesso em: 5 jul. 2024.

THIEL, David; DIRESTA, Renée; HANCOCK, Jeffrey; MCCAIN, R. Miles; WALD, Eli. Child safety on federated social media. **Stanford Internet Repository**, 07 2023. Disponível em: <https://purl.stanford.edu/vb515nd6874>. Acesso em: 12 ago. 2023.

TIKTOK. **Testing a new feature to enhance content on TikTok - Newsroom | TikTok**. 2025. Disponível em: <https://newsroom.tiktok.com/en-us/footnotes>. Acesso em: 20 abr. 2025.

Tor Project. **Protocol overview - Tor Specifications**. 2024. Disponível em: <https://spec.torproject.org/rend-spec/protocol-overview.html>. Acesso em: 24 ago. 2024.

Traefik Labs. **Traefik: The Cloud Native Application Proxy**. 2025. Disponível em: <https://github.com/traefik/traefik>. Acesso em: 12 mar. 2026.

VIANNA, Fernando Ressetti; PEINADO, Jurandir; GRAEML, Alexandre Reis. Crowdsourcing Platforms: Objective, Activities and Motivation. *In: AMCIS 2019 Proceedings*. Cancún, Mexico: Association for Information Systems, 2019. ISBN 978-0-9966831-8-0. Disponível em: [https://aisel.aisnet.org/amcis2019/virtual\\_communities/virtual\\_communities/8](https://aisel.aisnet.org/amcis2019/virtual_communities/virtual_communities/8). Acesso em: 12 maio 2026.

W3C. **Activity Streams 2.0**. W3C Consortium, 2017. Disponível em: <https://www.w3.org/TR/activitystreams-core/>. Acesso em: 28 mar. 2025.

W3C. **ActivityPub**. W3C Consortium, 2018. Disponível em: <https://www.w3.org/TR/activitypub/>. Acesso em: 28 mar. 2025.

W3C. **JSON-LD 1.1**. W3C Consortium, 2020. Disponível em: <https://www.w3.org/TR/json-ld11/>. Acesso em: 28 mar. 2025.

W3C. **Decentralized Identifiers (DIDs) v1.0**. 2022. Disponível em: <https://www.w3.org/TR/did-1.0/>. Acesso em: 3 ago. 2025.

WIEGMANN, M.; REIMER, J.H.; ERNST, M.; POTTHAST, M.; HAGEN, M.; STEIN, B. A Mastodon Corpus to Evaluate Federated Microblog Search. *In: Farzana S.M.; Frobe M.; Hendriksen G.; Granitzer M.; Hiemstra D.; Potthast M.; Zerhoubi S. (Ed.). Proceedings of the 1st International Workshop on Open Web Search (WOWS 2024)*. Glasgow, UK: CEUR-WS, 2024. (CEUR Workshop Proceedings, v. 3689), p. 37–49. ISSN 1613-0073. Disponível em: [https://ceur-ws.org/Vol-3689/WOWS\\_2024\\_paper\\_5.pdf](https://ceur-ws.org/Vol-3689/WOWS_2024_paper_5.pdf). Acesso em: 26 jun. 2026.

ZHANG, Hao; WEN, Yonggang; XIE, Haiyong; YU, Nenghai. **Distributed Hash Table: Theory, Platforms and Applications**. New York, NY: Springer New York,

2013. (SpringerBriefs in Computer Science). ISBN 978-1-4614-9007-4. Disponível em: <https://link.springer.com/10.1007/978-1-4614-9008-1>. Acesso em: 4 jul. 2024.

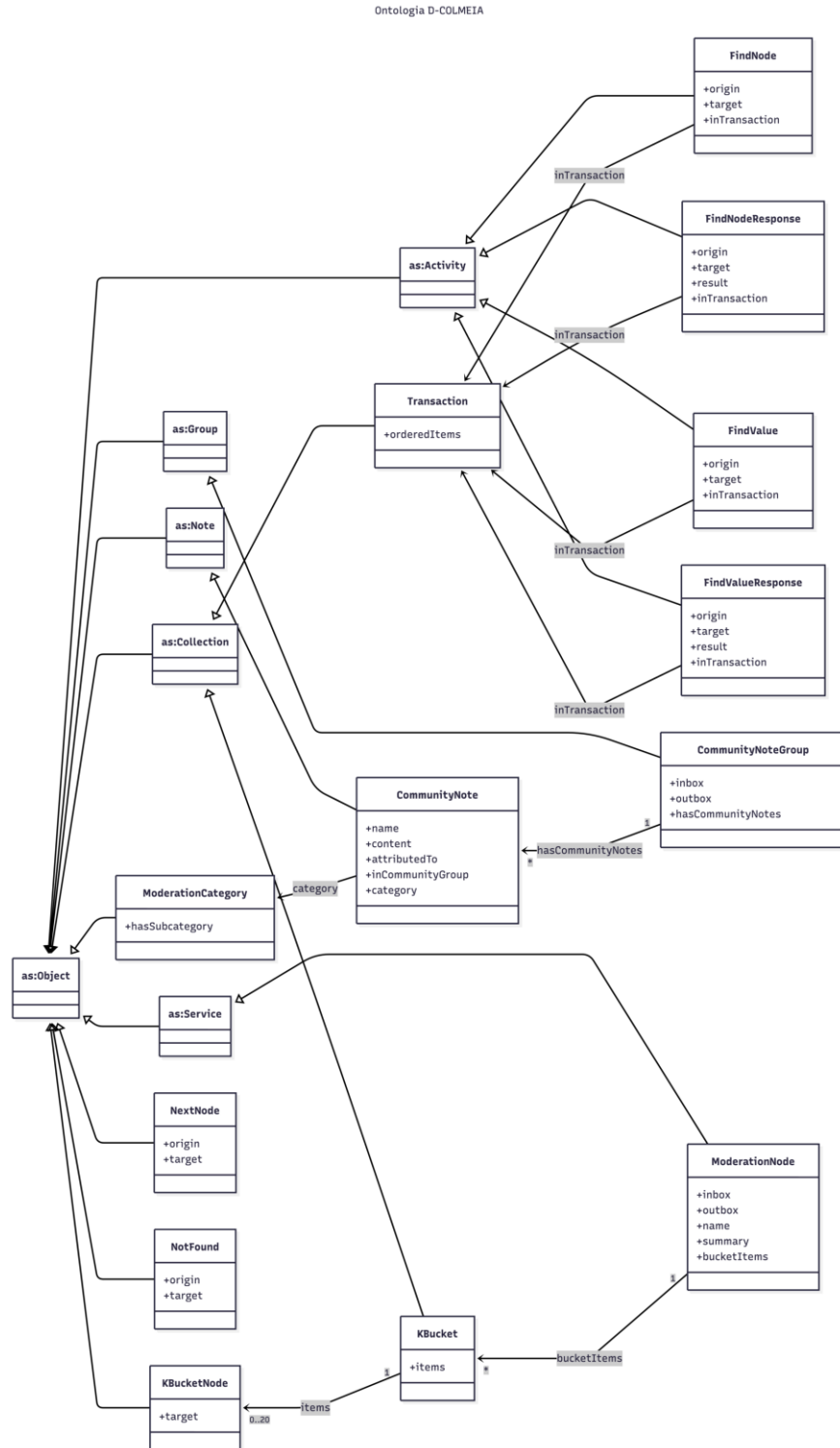
ZHANG, L.; HAN, W.; ZHANG, R.; WANG, L.; MENG, X. Identity-Based Key Management Scheme for Secure Discussion Group Establishment in DOSNs. **IEEE Transactions on Information Forensics and Security**, v. 18, p. 3706–3719, 2023. ISSN 1556-6013.

ZHANG, Zhilin; ZHAO, Jun; WANG, Ge; JOHNSTON, Samantha-Kaye; CHALHOUB, George; ROSS, Tala; LIU, Diyi; TINSMAN, Claudine; ZHAO, Rui; KLEEK, Max Van; SHADBOLT, Nigel. Trouble in paradise? understanding mastodon admin’s motivations, experiences, and challenges running decentralised social media. **Proceedings of the ACM on Human-Computer Interaction**, Association for Computing Machinery, v. 8, n. CSCW2, p. 1–24, 11 2024.

ZIA, Haris Bin; RAMAN, Aravindh; CASTRO, Ignacio; ANAOBI, Ishaku Hassan; CRISTOFARO, Emiliano De; SASTRY, Nishanth; TYSON, Gareth. Toxicity in the Decentralized Web and the Potential for Model Sharing. **Proceedings of the ACM on Measurement and Analysis of Computing Systems**, v. 6, n. 2, p. 1–25, maio 2022. ISSN 2476-1249. Disponível em: <https://dl.acm.org/doi/10.1145/3530901>. Acesso em: 18 mar. 2025.

## APÊNDICE A – REPRESENTAÇÃO GRÁFICA DA ONTOLOGIA DO D-COLMEIA.

Figura 41 – Representação gráfica da ontologia do D-Colmeia.



Fonte: Autoria própria.

## ANEXO A – TIPOLOGIA DE AVISOS DE CONTEÚDO NEON.

**Quadro 3 – Tipologia de avisos de conteúdo NEON.**

| Category (n) and definition                                                                                                              | Sub-categories                                                                                                                                                                                                                                                        |
|------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>1. Violence</b> (n = 536)<br>Content contains violence                                                                                | Violence; War; Weapons; Terrorism; Police brutality; Motiveless killing; Sexual violence; Animal cruelty; Torture; Genocide                                                                                                                                           |
| <b>2. Sex</b> (n = 332)<br>Content contains sexual themes, including nudity, sexual content and relationships                            | Nudity; Mild sexual content; Explicit sexual content; Relationship conflict; Reproductive health                                                                                                                                                                      |
| <b>3. Stigma</b> (n = 328)<br>Content depicts negative stereotypes about or attitudes towards a specific group, such as racism or sexism | Racism; Anti-religious (sub-categories: Anti-Semitic; Anti-Christian; Islamophobia); Colonialism; (sub-category: Slavery); Classism; Sexism (sub-categories: Misogyny; Misandry); Transphobia; Gender-identity; Sexuality (sub-category: Homophobia); Anti-disability |
| <b>4. Disturbing Content</b> (n = 236)<br>Content contains imagery, sounds, or effects that may frighten, disgust or scare               | Disturbing content with threat; Horror and terror; Disturbing imagery; Medical content; Human bodies and functions                                                                                                                                                    |
| <b>5. Language</b> (n = 235)<br>Content contains language which is sexual, crude or offensive                                            | Sexual language; Adult humour; Swearing; Offensive language                                                                                                                                                                                                           |
| <b>6. Risky Behaviours</b> (n = 200)<br>Content depicts risky lifestyle behaviours                                                       | Drug misuse; Alcohol misuse; Tobacco; Gambling                                                                                                                                                                                                                        |
| <b>7. Mental Health</b> (n = 108)<br>Content relates to mental health issues                                                             | Mental health; Eating disorders; Trauma; Self-harm and suicide; Depression; OCD; Panic attacks; Anxiety (sub-categories: Spiders; Snakes; Insects; Needles; Eye contact; Irregular patterns); Hair pulling                                                            |
| <b>8. Death</b> (n = 49)<br>Content relates to human death or dying                                                                      | Death; Accidents; Natural disasters                                                                                                                                                                                                                                   |
| <b>9. Parental Guidance</b> (n = 47)<br>Content may not be appropriate for children                                                      | Online access; Cyber-bullying; Competitive content; Imitative content; Upsetting content; Non-realistic violence                                                                                                                                                      |
| <b>10. Crime</b> (n = 38)<br>Content depicts or relates to criminal activity                                                             |                                                                                                                                                                                                                                                                       |
| <b>11. Abuse</b> (n = 37)<br>Content depicts or relates to abuse                                                                         | Child abuse; Emotional abuse; Physical/sexual abuse; Neglect                                                                                                                                                                                                          |
| <b>12. Sociopolitical</b> (n = 27)<br>Content includes social or political issues                                                        | Injustice; Political issues; Nazism; Class issues                                                                                                                                                                                                                     |
| <b>13. Flashing Lights</b> (n = 27)<br>Content includes strobe or flashing lighting                                                      |                                                                                                                                                                                                                                                                       |
| <b>14. Objects</b> (n = 4)<br>Content contains specific objects                                                                          |                                                                                                                                                                                                                                                                       |

<https://doi.org/10.1371/journal.pone.0266722.t001>

**Fonte: Charles *et al.* (2022).**