Guilherme Camargo de Almeida

Análise do *Authenticated Transfer Protocol* versus requisitos não-funcionais de redes sociais descentralizadas

São Paulo

2025

Guilherme Camargo de Almeida

Análise do *Authenticated Transfer Protocol* versus requisitos não-funcionais de redes sociais descentralizadas

Versão Original

Monografia apresentada ao PECE – Programa de Educação Continuada em Engenharia da Escola Politécnica da Universidade de São Paulo como parte dos requisitos para a conclusão do curso de MBA em Engenharia de Software.

Área de Concentração: Engenharia de Software

Orientador: Prof. Me. Leonardo Dominguez Dias

São Paulo
2025

Catalogação-na-publicação

de Almeida, Guilherme
Análise do Authenticated Transfer Protocol versus requisitos não funcionais de redes sociais descentralizadas / G. de Almeida -- São Paulo, 2025.
61 p.

Monografia (MBA em Engenharia de Software) - Escola Politécnica da Universidade de São Paulo. PECE – Programa de Educação Continuada em Engenharia.

1.REDES SOCIAIS 2.ENGENHARIA DE REQUISITOS 3.SISTEMAS DISTRIBUÍDOS I.Universidade de São Paulo. Escola Politécnica. PECE – Programa de Educação Continuada em Engenharia II.t.

Nome: DE ALMEIDA, Guilherme Camargo

Título: Análise do Authenticated Transfer Protocol versus requisitos não-funcionais de redes sociais descentralizadas

Monografia apresentada ao PECE – Programa de Educação Continuada em Engenharia da Escola Politécnica da Universidade de São Paulo como parte dos requisitos para a conclusão do curso de MBA em Engenharia de Software.

Aprovado em: / /

Banca Examinadora

Prof(a). Dr(a). _____

Instituição: _____

Julgamento: _____

Prof(a). Dr(a). _____

Instituição: _____

Julgamento: _____

Prof(a). Dr(a). _____

Instituição: _____

Julgamento: _____

DEDICATÓRIA

Dedico este trabalho à minha família

AGRADECIMENTOS

A meus pais, que ajudam e apoiam todas minhas ideias, inclusive de fazer este curso.

À Nina, que está ao meu lado sempre.

Ao meu orientador, que com sua paciência e inteligência me guiou ao longo de todo este trabalho.

À Escola Politécnica da Universidade de São Paulo e ao PECE por este MBA.

RESUMO

DE ALMEIDA, G.C. Análise do Authenticated Transfer Protocol versus requisitos não-funcionais de redes sociais descentralizadas. 2025. 62 páginas. Monografia (MBA em Tecnologia de Software). Programa de Educação Continuada em Engenharia da Escola Politécnica da Universidade de São Paulo. São Paulo. 2025.

O cenário contemporâneo das redes sociais é marcado por uma concentração sem precedentes de poder nas mãos de poucas empresas de tecnologia, que controlam não apenas os dados dos usuários, mas também têm capacidade de moldar significativamente a opinião pública através de algoritmos opacos próprios. A monetização agressiva dos dados pessoais e o abuso do poder curatorial criam um conflito de interesses com as verdadeiras intenções de socialização dos usuários, enquanto o aprisionamento em "jardins murados" dificulta a migração para alternativas. Nesse contexto, esta monografia tem como objetivo analisar a adequação do protocolo Authenticated Transfer Protocol (ATProto) como fundação para redes sociais descentralizadas (RSDs), através de uma metodologia que combina análise documental com observação prática de implementações existentes. O trabalho desenvolve um framework analítico que integra conceitos de sistemas distribuídos com análise de requisitos não-funcionais, baseando-se nas características de qualidade definidas pela ISO/IEC 25010:2011 e nos objetivos fundamentais de sistemas distribuídos estabelecidos por Steen e Tanenbaum. Os resultados demonstram que o ATProto estabelece uma base sólida para RSDs, com avanços significativos em relação a implementações anteriores como o Secure Scuttlebutt e o ActivityPub, embora enfrente desafios importantes em escalabilidade e privacidade.

Palavras-chave: Redes Sociais Descentralizadas, ATProto, Sistemas Distribuídos, Requisitos Não-funcionais, Descentralização.

ABSTRACT

DE ALMEIDA, G.C. Analysis of the Authenticated Transfer Protocol versus non-functional requirements of decentralized social networks. 2025. 62 pages. Monograph (MBA in Software Technology). Continuing Engineering Education Program at Polytechnic School of University of São Paulo. São Paulo. 2025.

The contemporary social media landscape is marked by an unprecedented concentration of power in the hands of few technology companies, who control not only user data but also have the ability to significantly shape public opinion through their own opaque algorithms. The aggressive monetization of personal data and abuse of curatorial power create a conflict of interest with users' true socialization intentions, while imprisonment in "walled gardens" makes migration to alternatives difficult. In this context, this monograph aims to analyze the adequacy of the Authenticated Transfer Protocol (ATProto) as a foundation for decentralized social networks (DSNs), through a methodology that combines documentary analysis with practical observation of existing implementations. The work develops an analytical framework that integrates distributed systems concepts with non-functional requirements analysis, based on quality characteristics defined by ISO/IEC 25010:2011 and fundamental objectives of distributed systems established by Steen and Tanenbaum. Results show that ATProto establishes a solid foundation for DSNs, with significant advances compared to previous implementations such as Secure Scuttlebutt and ActivityPub, although facing important challenges in scalability and privacy.

Keywords: Decentralized Social Networks, ATProto, Distributed Systems, Non-functional Requirements, Decentralization.

LISTA DE ILUSTRAÇÕES

Pág.

LISTA DE TABELAS

Pág.

LISTA DE ABREVIATURAS E SIGLAS

API - Application Programming Interface (Interface de Programação de Aplicação)

ATProto - Authenticated Transfer Protocol (Protocolo de Transferência Autenticada)

DID - Decentralized Identifier (Identificador Descentralizado)

DNS - Domain Name System (Sistema de Nomes de Domínio)

ISO/IEC - International Organization for Standardization/International Electrotechnical Commission (Organização Internacional para Padronização/Comissão Eletrotécnica Internacional)

P2P - Peer-to-peer (Ponto a ponto)

PBC - Public Benefit Corporation (Corporação de Benefício Público)

PDS - Personal Data Server (Servidor de Dados Pessoal)

RNF - Requisito Não-Funcional

RSD - Rede Social Descentralizada

SMART - Specific, Measurable, Achievable, Relevant, Time-bound (Específico, Mensurável, Atingível, Relevante, Temporalmente definido)

SSB - Secure Scuttlebutt

UCAN - User Controlled Authorization Networks (Redes de Autorização Controladas pelo Usuário)

W3C - World Wide Web Consortium

1. Introdução

O cenário contemporâneo das redes sociais é marcado por uma concentração sem precedentes de poder nas mãos de poucas empresas de tecnologia. Hoje, mais de 62% da população mundial utiliza redes sociais, com pelo menos 7 delas com mais de 1 bilhão de usuários (KEMP, 2024). São empresas que controlam não apenas os dados dos usuários, mas também têm capacidade de moldar significativamente a opinião pública por meio de algoritmos próprios e opacos de curadoria e moderação.

São algoritmos projetados para maximizar as oportunidades de lucro das redes sociais no que é chamado de "economia da atenção" (BROWN, 2021), que consiste em buscar maior retenção na plataforma, que por sua vez significa maior quantidade de dados gerados, que são usados para oferecer oportunidades de publicidade para anunciantes. Isso cria um inevitável conflito de interesses com as verdadeiras intenções sociais dos usuários: a moderação adequada de conteúdo e o desestímulo a conteúdos polêmicos, embora desejáveis do ponto de vista social, tornam-se economicamente desvantajosos e ativamente evitados por tomadores de decisão (HORWITZ; SEETHARAMAN, 2020).

Além da monetização agressiva de dados pessoais e desse abuso de poder curatorial pensado para promover polarização, as redes sociais centralizadas também aprisionam seus usuários em um "jardim murado" (MCCOWN; NELSON, 2009): todas as interações, amizades e conteúdos produzidos ficam retidos dentro da plataforma. Abandoná-la significa perder não apenas dados, mas todo um tecido social construído naquele ambiente, criando uma barreira significativa à migração para alternativas, mesmo quando os usuários estão insatisfeitos.

Nesse contexto, a descentralização emerge como uma resposta tecnológica e social a estes desafios - e o Authenticated Transfer Protocol (ATProto) representa uma nova abordagem, buscando estabelecer fundações técnicas para redes sociais verdadeiramente descentralizadas que preservem a experiência do usuário sem comprometer autonomia e controle sobre seus próprios dados e interações. Este trabalho se propõe a analisar sistematicamente sua adequação para este propósito,

considerando tanto aspectos técnicos quanto suas implicações práticas para o futuro das redes sociais.

Tabela 1: resumo da comparação entre aspectos fundamentais em redes sociais entre implementações centralizadas e descentralizadas

	Redes Centralizadas	**Redes Descentralizadas**
Dados do usuário	Concentrado nas empresas de tecnologia, sem uso fora de suas plataformas	Pertence ao usuário, que pode disponibilizar para qualquer plataforma que queira usá-los
Moderação e curadoria de conteúdo	Definida unilateralmente pelas plataformas	Distribuída e adaptável às necessidades de cada comunidade
Portabilidade de dados	Restrita, com dados aprisionados em "jardins murados"	Facilitada através de padrões abertos e interoperáveis
Resistência à censura	Vulnerável a pressões corporativas e governamentais	Fortalecida através da distribuição de infraestrutura e ausência de pontos únicos de controle
Acesso e exclusão	Sujeito a decisões arbitrárias das plataformas	A exclusão de uma plataforma não leva a perda dos dados, que podem ser utilizados em outra
Diversidade de aplicações	Limitada pelas regras e APIs das plataformas dominantes	Fomentada através de protocolos abertos que permitem inovação descentralizada

Fonte: Elaborada pelo autor

1.1 Motivações

A motivação para esta pesquisa emerge da observação direta das limitações e problemas enfrentados por usuários de redes sociais centralizadas, particularmente evidenciados durante as recentes mudanças na plataforma X (anteriormente Twitter). A falta de transparência nas decisões algorítmicas, a impossibilidade de

portabilidade de dados e conexões sociais, e a dependência de políticas de moderação intencionalmente ineficazes e centralizadas demonstram a necessidade urgente de alternativas que priorizem a autonomia do usuário, demonstrado também por um latente interesse de instituições da sociedade civil e do Estado para regulamentar esses espaços (PACHECO, 2023).

Esta perspectiva é enriquecida por uma experiência prática do autor com o desenvolvimento e implementação de aplicações no âmbito do ATProto, incluindo participação ativa em discussões técnicas sobre o ATProto através de Request for Comments, operação de Personal Data Servers, e interações diretas com a equipe de desenvolvimento do ATProto.

1.2 Objetivo

Este trabalho tem como objetivo avaliar a adequação do Authenticated Transfer Protocol (ATProto) como fundação para redes sociais descentralizadas (RSDs), através de uma análise sistemática que combina duas perspectivas complementares: os objetivos fundamentais de sistemas distribuídos estabelecidos por Steen e Tanenbaum (2023) e as características de qualidade de software definidas pela norma ISO/IEC 25010:2011. Esta abordagem permite examinar tanto a arquitetura técnica do protocolo quanto sua capacidade de atender aos requisitos não-funcionais críticos para redes sociais descentralizadas.

1.3 Justificativas

A relevância desta pesquisa é fundamentada não apenas na importância crescente das redes sociais descentralizadas, mas também pelo ATProto se provar bastante promissor em termos de adesão e mesmo pela robustez técnica e acadêmica da equipe por trás. O protocolo representa uma confluência única de experiências em descentralização, sendo desenvolvido por especialistas reconhecidos na área, incluindo Martin Kleppmann, professor da Universidade de Cambridge e autor do influente "Designing Data-Intensive Applications", Paul Frazee, pioneiro no desenvolvimento do Secure Scuttlebutt, e Daniel Holmgren, do time por trás da especificação de autenticação descentralizada UCAN.

1.4 Método de Pesquisa

A metodologia adotada combina análise documental aprofundada com observação prática de aplicações do protocolo em casos reais. O trabalho desenvolve um framework analítico que integra conceitos de sistemas distribuídos com análise de requisitos não-funcionais, baseando-se nas características de qualidade definidas pela ISO/IEC 25010:2011 e nos objetivos fundamentais de sistemas distribuídos estabelecidos por Steen e Tanenbaum.

1.5 Estrutura do Trabalho

O trabalho está organizado em cinco capítulos. Após esta introdução, o Capítulo 2 apresenta uma revisão bibliográfica sobre sistemas descentralizados e redes sociais. O Capítulo 3 desenvolve uma análise sistemática dos requisitos não-funcionais de redes sociais descentralizadas e a conformidade do ATProto, enquanto o Capítulo 4 discute os resultados obtidos e suas implicações. Por fim, o Capítulo 5 apresenta as conclusões e sugere direções para trabalhos futuros.

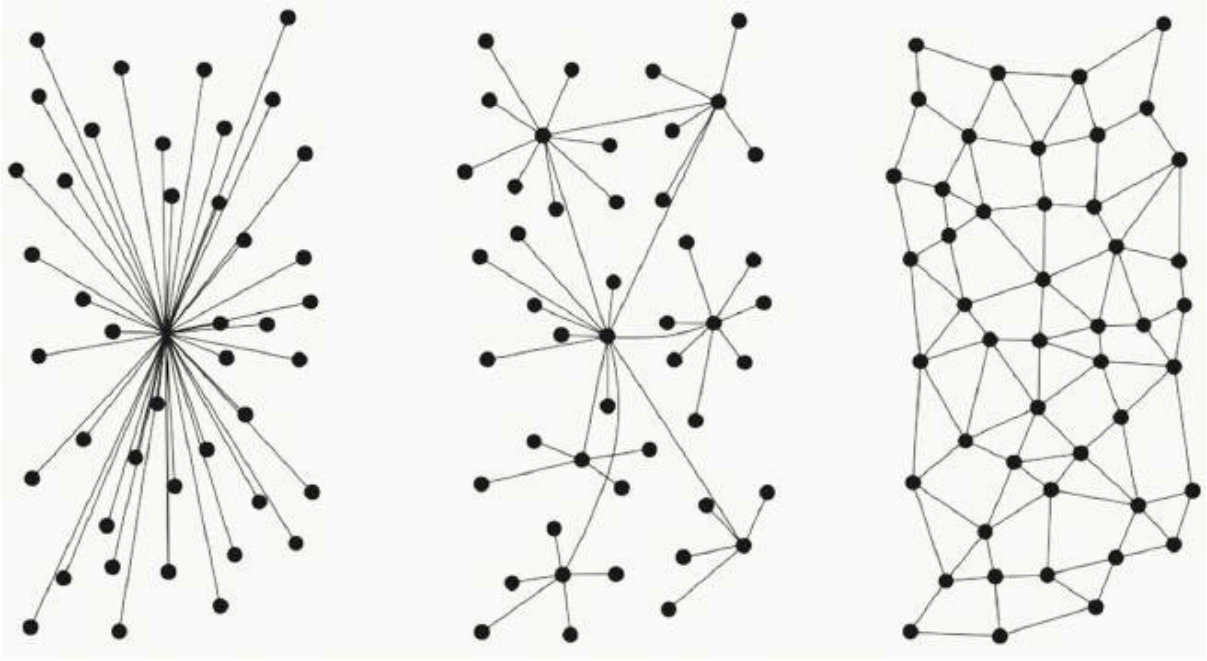
2. Revisão bibliográfica

Compreender redes sociais descentralizadas, seus requisitos não-funcionais e o papel do ATProto nesse contexto exige, primeiramente, compreender o que significa descentralização no contexto de redes sociais, quais arquiteturas existem para isso e o que se entende por requisitos não funcionais. Este capítulo é uma abordagem detalhada da fundamentação teórica desses conceitos.

2.1 Sistemas descentralizados e sistemas distribuídos

A compreensão de redes sociais descentralizadas requer, primeiramente, um entendimento mais amplo sobre descentralização como uma abordagem arquitetural para sistemas em rede. Este entendimento, por sua vez, está intrinsecamente ligado ao conceito de sistemas distribuídos, sendo inclusive muitas vezes tratado como um subconjunto deles, do ponto de vista arquitetural (STEEN; TANENBAUM, 2023).

Figura 1 - Rede centralizada, descentralizada e distribuída, respectivamente



Fonte: Paul Baran (1962)

Em Distributed Systems, Steen e Tanenbaum (2023) fazem questão de fugir da definição meramente topológica representada na figura 1 - em que a diferença entre um sistema descentralizado e um distribuído é como os nós se conectam em um

grafo de rede - passando a adotar uma visão que busca entender a intenção por trás das conexões entre os nós. Com isso, estabelecem que os recursos e processos de sistemas descentralizados são **necessariamente** distribuídos, enquanto sistemas distribuídos são **suficientemente** distribuídos.

Os autores argumentam que a descentralização nunca deve ser, por si só, um objetivo de arquitetura de um sistema; deve-se mirar em torná-lo suficientemente distribuído. O sistema descentralizado emerge quando o espalhamento dos recursos e processos não é uma escolha, mas uma necessidade inerente ao problema sendo resolvido.

Esclarecida a diferença conceitual entre descentralização e distribuição, é importante destacar os desafios técnicos compartilhados por ambos os modelos, os quais se traduzem em objetivos fundamentais de design. Especificamente, Steen e Tanenbaum (2023) delineiam seis objetivos fundamentais que sistemas distribuídos devem almejar: compartilhamento de recursos, invisibilidade na distribuição, abertura para integração, confiabilidade, escalabilidade e segurança.

Por **compartilhamento de recursos**, compreende-se a distribuição de qualquer coisa entre os participantes do sistema, de serviços e processos a repositórios de dados, como o caso em redes sociais descentralizadas, sendo importante considerar os mecanismos que permitem isso.

Invisibilidade na distribuição consiste em não ser possível perceber que o sistema que está sendo utilizado é distribuído; há sete tipos de invisibilidade possíveis, mas as mais importantes no contexto de redes sociais descentralizadas são **invisibilidade de acesso**, que consiste em tornar invisível qualquer detalhe da hospedagem de cada dado do sistema, e **de migração**, que consiste em tornar invisível a transferência dos dados acessados entre partes do sistema. Com isso, almeja-se oferecer uma experiência idêntica a de um sistema centralizado.

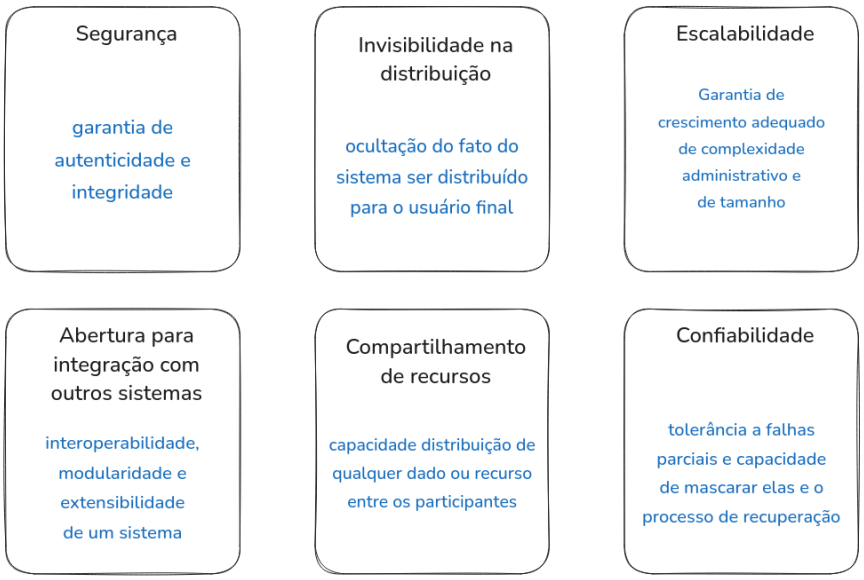
Abertura para integração com outros sistemas, para além do que o nome evidencia, diz respeito à interoperabilidade, modularidade e extensibilidade de um sistema. Em outras, ele deve aderir a regras semânticas que determinam como os participantes do sistema devem trocar informações.

A **confiabilidade**, no contexto de sistemas distribuídos, se refere principalmente à tolerância a falhas parciais e sua capacidade de mascarar tanto elas quanto seu processo de recuperação. Os conceitos fundamentais para avaliar a confiabilidade de um sistema são disponibilidade, confiabilidade, proteção e manutenibilidade, que serão explorados mais a frente.

Em termos de **escalabilidade**, uma rede social descentralizada precisa atender a dois desafios fundamentais: crescimento **em tamanho** e **complexidade administrativa**. O tamanho refere-se à capacidade do sistema de expandir sem degradação perceptível de performance; já a escalabilidade administrativa trata da capacidade de manter uma gestão eficaz mesmo quando o sistema abrange múltiplas organizações ou indivíduos, com suas próprias políticas e necessidades.

Por fim, **segurança** é bastante evidente, uma vez que não se pode depender de um sistema que não é seguro, especialmente quanto à confidencialidade e integridade. Em um sistema distribuído, há uma camada adicional de complexidade ao tratar de autenticação e autorização, já que torna-se necessário perguntar se o participante do sistema capaz de autenticar é, ele próprio, de confiança - adjetivo recorrente ao tratar do assunto nesse âmbito, assim como a criptografia como forma de alcançá-la.

Figura 2 - resumo dos objetivos a serem alcançados por sistemas distribuídos



Fonte: Elaborada pelo autor a partir dos objetivos para sistemas distribuídos determinados por Steen e Tanenbaum (2023)

Estes objetivos de design oferecem uma base robusta para analisar redes sociais descentralizadas, que são, em sua essência, sistemas distribuídos. Utilizando conceitos da literatura de requisitos não-funcionais, que será explorada mais a fundo na seção 2.5, esses objetivos tornam-se excelentes guias para analisar a qualidade dessas redes.

2.2 Descentralização de redes sociais

A descentralização de redes sociais envolve não só dimensões técnicas como também sociais. Como destacam Barabas, Narula e Zuckerman (2017), uma verdadeira descentralização deve contemplar três dimensões fundamentais: **resistência à censura**, protegendo a expressão tanto de interferências corporativas quanto governamentais; **distribuição do poder curatorial**, evitando que uma única entidade controle arbitrariamente a visibilidade do conteúdo; e **garantia de acesso universal**, impedindo exclusões arbitrárias da rede.

A essas dimensões, acrescenta-se o aspecto levantado por Masnick (2019), que o caminho para esta transformação passa necessariamente pelo **fim do monopólio sobre os dados dos usuários**. Uma solução verdadeiramente descentralizada deve permitir que as pessoas mantenham controle sobre suas informações e possam transportá-las livremente entre diferentes serviços e plataformas.

Para materializar estes objetivos de descentralização, duas principais arquiteturas se destacam: as **redes sociais peer-to-peer (P2P)** e as **redes federadas**. Para melhor compreender o que as diferencia, é possível recorrer à conceitualização de P2P dentro de um espectro de descentralização feito por Milojicic et al. (2003). Para os pesquisadores, sistemas P2P puros são aqueles em que todos os nós possuem capacidades e responsabilidades equivalentes, sem qualquer ponto central de coordenação; sistemas P2P híbridos mantêm algum nível de centralização através de servidores centralizados que, de alguma forma, auxiliam a descoberta e/ou coordenação entre os nós. Quando se fala em redes sociais P2P refere-se à taxonomia pura; já redes federadas, melhor se encaixam na definição híbrida.

Como destaca Graber (2020), tanto protocolos P2P quanto federados são abordagens distintas para projetar redes que estruturalmente empoderam usuários,

cada uma com seus próprios benefícios e limitações. A compreensão dessas diferentes estratégias de descentralização, suas implicações técnicas e seus impactos práticos na experiência dos usuários é fundamental para avaliar sua adequação como alternativas ao modelo centralizado dominante.

2.3 Arquitetura Peer-to-Peer em Redes Sociais

Como discutido na seção 2.2, em uma arquitetura peer to peer (P2P) pura, cada participante atua simultaneamente como produtor e consumidor de conteúdo e é capaz de conectar-se diretamente entre si, sem depender de servidores centrais para mediar suas interações (MILOJICIC et al., 2003). Esta arquitetura distribui tanto os dados quanto a necessidade de processamento entre os participantes da rede, criando um sistema inerentemente descentralizado.

No contexto de redes sociais, a implementação de uma arquitetura P2P traz implicações significativas para aspectos como disponibilidade de dados, privacidade, escalabilidade e governança. Ao eliminar a necessidade de infraestrutura centralizada, redes P2P podem teoricamente escalar infinitamente com seu número de usuários, já que cada novo participante depende principalmente de seus próprios recursos computacionais. No entanto, esta mesma característica também introduz desafios únicos em termos de consistência de dados e experiência do usuário.

Dentre as implementações de redes sociais P2P, o Secure Scuttlebutt (SSB) se destaca por sua abordagem inovadora para estes desafios. Desenvolvido inicialmente em 2014, o protocolo estabeleceu importantes fundamentos para redes sociais descentralizadas que continuam influenciando o desenvolvimento de novos protocolos até hoje.

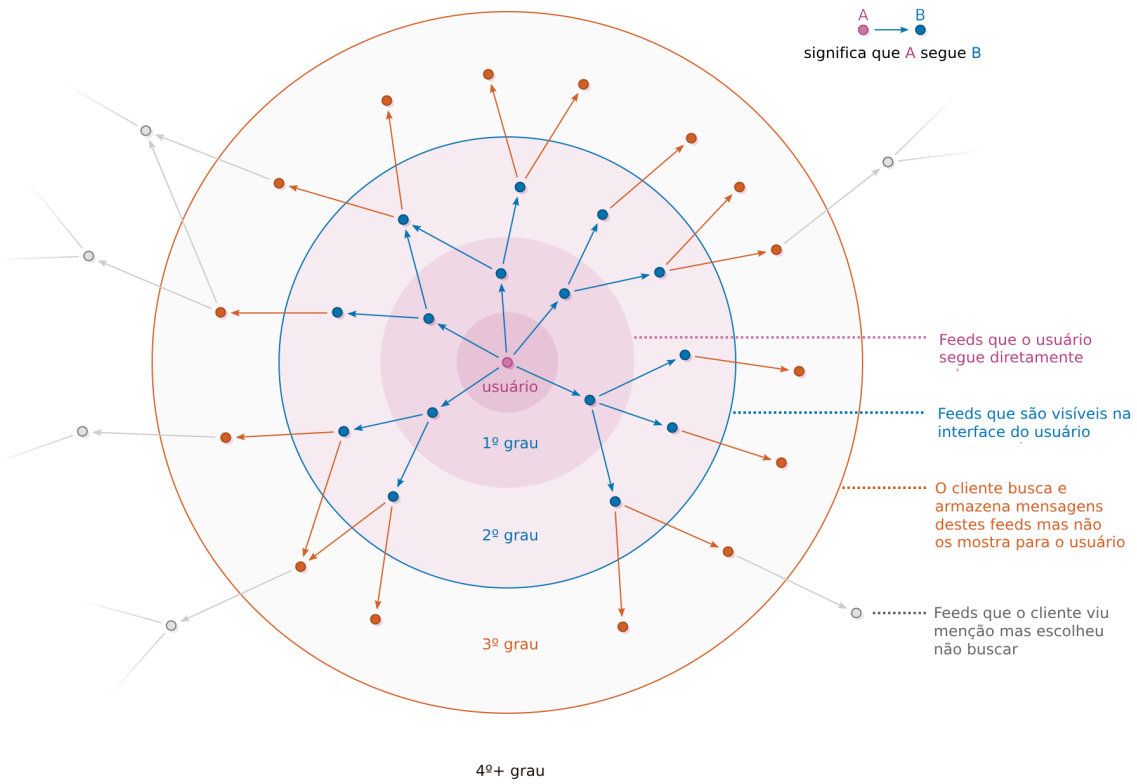
2.3.1 O Protocolo Secure Scuttlebutt

O Secure Scuttlebutt (SSB) destaca-se como uma das implementações mais sofisticadas e bem-sucedidas de rede social totalmente peer-to-peer. O protocolo baseia-se em *logs append-only* assinados criptograficamente pelo usuário,

estabelecendo um sistema que prioriza verificabilidade e integridade dos dados enquanto mantém operação totalmente descentralizada.

No centro do SSB está o conceito de logs pessoais imutáveis, onde cada usuário mantém seu próprio registro ordenado de atividades. As mensagens são encadeadas criptograficamente, com cada entrada contendo uma referência hash à anterior, formando uma cadeia contínua inviolável. A distribuição de conteúdo ocorre através de replicação seletiva baseada em relacionamentos sociais: quando um usuário segue outro, seu cliente SSB passa a replicar o *log* completo daquele usuário e, por extensão, também replica os *logs* dos usuários seguidos até um número configurável de graus de separação (TARR et al., 2019).

Figura 3 - Funcionamento do mecanismo de replicação do SSB de maneira simplificada



Fonte: Secure Scuttlebutt Consortium (2023)

Esta arquitetura resulta em uma rede naturalmente resistente a atores maliciosos e capaz de operar *offline*, já que usuários podem interagir com conteúdo armazenado localmente mesmo sem conectividade. A sincronização ocorre eventualmente

quando conexões são restabelecidas, um modelo particularmente resiliente para cenários de conectividade intermitente.

No entanto, o SSB enfrenta limitações significativas. A gestão de identidades é particularmente problemática, pois está intrinsecamente ligada a pares de chaves criptográficas sem mecanismo de recuperação, além de impossibilitar o uso de uma mesma identidade - e consequentemente sua rede - em múltiplos dispositivos (GRABER, 2020). A replicação completa de *logs* impõe desafios crescentes de escalabilidade, e a natureza imutável dos dados dificulta moderação efetiva de conteúdo. Por fim, a ausência de mecanismos centralizados para descoberta de conteúdo representa uma barreira significativa para adoção em larga escala.

O SSB demonstrou o potencial de redes *peer-to-peer* para interações sociais online, mesmo que suas limitações tenham restringido sua massificação. Suas contribuições seguem influenciando o desenvolvimento de novas abordagens para descentralização de redes sociais, incluindo o ATProto, desenvolvido posteriormente por Paul Frazee, um dos principais contribuidores do SSB. Suas qualidades e limitações serão discutidas em mais detalhes no capítulo 3.

2.4 Arquitetura Federada em Redes Sociais

No espectro da descentralização, as arquiteturas federadas representam uma abordagem intermediária que busca equilibrar os benefícios da descentralização com a praticidade operacional de uma rede centralizada. Diferentemente dos sistemas puramente *peer-to-peer,* onde cada nó possui capacidades equivalentes, a federação implementa um modelo híbrido onde servidores atuam como peers entre si; logo, um usuário escolhe um servidor para se hospedar, mas tem acesso a toda a rede de servidores interconectados (GRABER, 2020).

Esta abordagem atende alguns dos desafios práticos observados em implementações P2P puras, ao mesmo tempo que busca evitar os problemas de centralização excessiva das redes sociais tradicionais. A ideia central da federação em redes sociais é permitir que qualquer participante possa executar partes da infraestrutura necessária para o funcionamento da rede ou apenas escolher apenas participar de uma infraestrutura existente (RAMAN et al., 2019).

Para viabilizar esta interoperabilidade sem depender de uma autoridade central coordenadora, protocolos de federação definem padrões de comunicação e formatos de dados que permitem que diferentes implementações trabalhem em conjunto. No entanto, existe significativa variação em como cada protocolo implementa esta federação, principalmente no que diz respeito à distribuição de responsabilidades entre os diferentes componentes do sistema.

O protocolo ActivityPub, por exemplo, adota uma abordagem mais direta onde servidores comunicam-se diretamente entre si para propagar atualizações, similar ao funcionamento do sistema de email (GRABER, 2020). Outros, como o ATProto, implementam uma arquitetura em camadas com serviços especializados para indexação e agregação de dados em escala global, inspirando-se no funcionamento da própria internet (BLUESKY PBC, 2023).

A seguir, serão analisadas essas duas implementações proeminentes da arquitetura federada, cada uma com suas vantagens e desvantagens: o ActivityPub, que prioriza simplicidade e compatibilidade com modelos web tradicionais através de federação direta entre servidores, e o ATProto, que busca maximizar escalabilidade e autonomia do usuário através de uma arquitetura em camadas mais sofisticada.

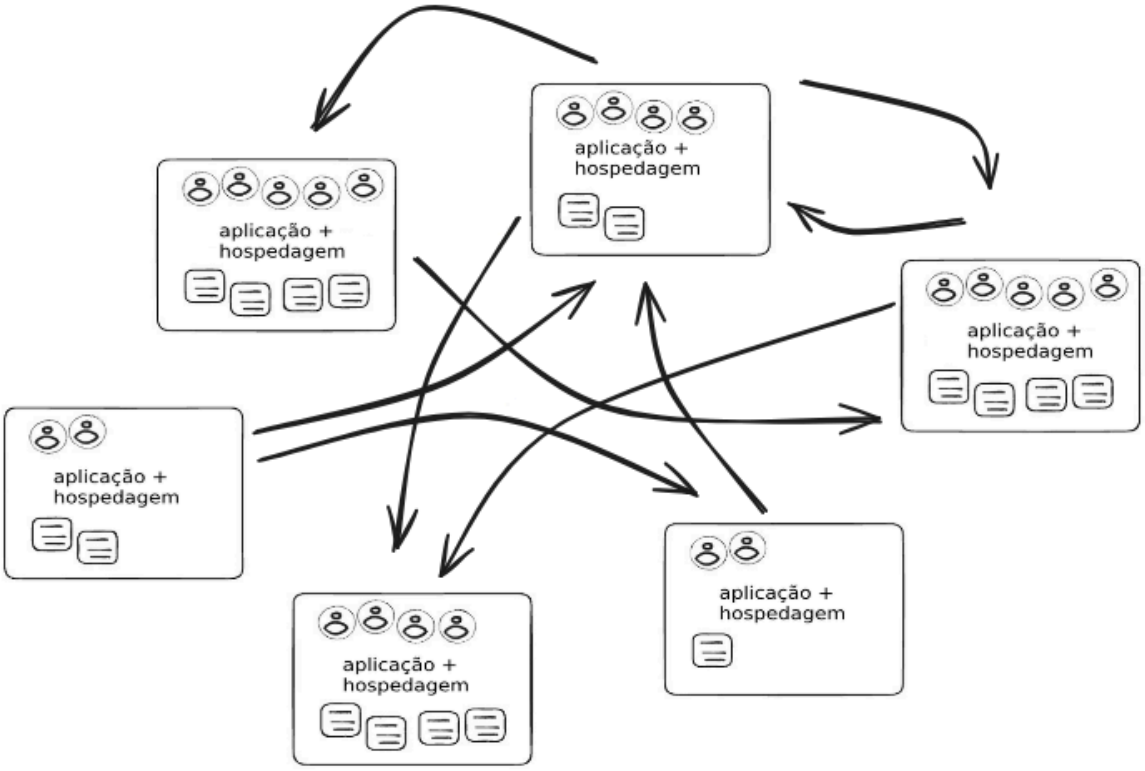
2.4.1 ActivityPub

O ActivityPub é um protocolo de federação padronizado pelo World Wide Web Consortium (W3C) que estabelece uma abordagem estruturada para interconectar servidores de redes sociais. Diferentemente de protocolos anteriores, o ActivityPub define uma separação clara entre duas camadas de interação: uma API cliente-servidor, que permite aos usuários interagirem com sua instância local, e um protocolo servidor-servidor que possibilita a federação entre instâncias (LEMMER-WEBER et al., 2018).

No centro do ActivityPub está o formato ActivityStreams, um vocabulário padronizado que descreve interações sociais como objetos estruturados. Quando um usuário realiza uma ação - como criar uma postagem ou seguir outro usuário - esta é representada como uma Atividade que especifica seu tipo, ator, objeto e destinatários. O servidor de origem então entrega esta Atividade diretamente aos

servidores de todos os destinatários pretendidos, de maneira análoga ao funcionamento do sistema de email.

Figura 4 - Resumo da arquitetura do ActivityPub: instâncias hospedam dados de usuários e suas próprias aplicações, comunicando-se entre si.



Fonte: Dan Abramov (2024)

Como ilustrado na Figura 4, cada instância do ActivityPub hospeda seus próprios dados e aplicações, mantendo autonomia sobre suas políticas e governança. A federação ocorre através da comunicação direta entre instâncias, que trocam mensagens para propagar atualizações e manter a consistência do grafo social distribuído. Esta arquitetura permite que diferentes implementações do protocolo (como as redes sociais Mastodon e Pleroma) interoperem perfeitamente, já que todas utilizam a mesma estrutura de mensagem definida pelo padrão ActivityPub.

A identidade no ActivityPub é vinculada ao servidor, com usuários sendo identificados por uma combinação de nome de usuário e domínio (por exemplo, usuario@servidor.social). Embora isto simplifique aspectos de descoberta e autenticidade, também cria uma dependência significativa do servidor escolhido. A migração entre servidores, embora possível, implica em mudança de identidade e

requer cooperação ativa do servidor original para preservar conexões sociais (KLEPPMANN et al., 2024). Este aspecto específico será explorado em maior profundidade no Capítulo 3, onde são discutidas as implicações práticas desta escolha arquitetural.

O protocolo oferece uma elevada autonomia para cada servidor em termos de moderação e governança. Administradores podem definir suas próprias políticas e decidir com quais outros servidores desejam federar. Esta flexibilidade permite que diferentes comunidades estabeleçam suas próprias normas, mas também pode resultar em fragmentação quando servidores optam por não federar entre si.

Esta arquitetura relativamente simples e construída sobre padrões web estabelecidos facilita a implementação, como demonstrado pelo sucesso do Mastodon. No entanto, também apresenta desafios de escalabilidade e consistência (RAMAN et al., 2019). Servidores precisam manter conexões com potencialmente milhares de outros servidores, e falhas temporárias podem resultar em propagação inconsistente de conteúdo - tópicos que serão aprofundados na análise técnica do Capítulo 3.

2.4.2 Authenticated Transfer Protocol

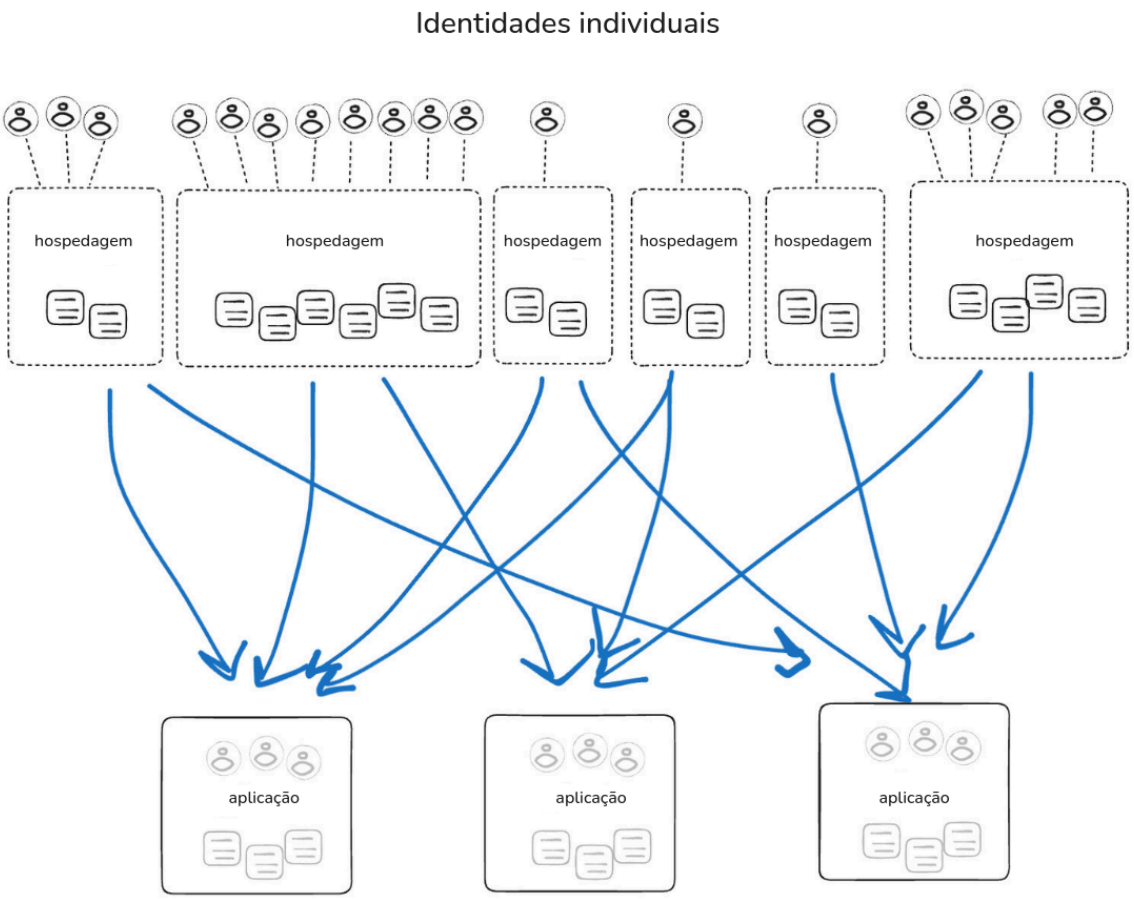
O AT Protocol representa uma abordagem mais sofisticada à federação em redes sociais, posicionando-se entre a descentralização radical do SSB e a federação direta do ActivityPub. Desenvolvido inicialmente pela Bluesky, o protocolo implementa uma arquitetura em camadas que separa claramente identidade, armazenamento de dados e infraestrutura de rede (KLEPPMANN et al., 2024), conforme a figura 5.

O protocolo classifica-se como federado justamente por essas camadas serem independentes entre si em termos de operação (BLUESKY PBC, 2023). Pessoas ou organizações podem oferecer hospedagem para repositórios de dados, para agregadores de conteúdo ou para clientes de rede social, chamados no contexto do ATProto de App Views.

No centro do AT Protocol está o conceito de repositórios pessoais de dados, implementados como estruturas de dados criptograficamente verificáveis chamadas

Merkle Search Trees (AUVOLAT; TAIANI, 2019). Cada usuário possui um repositório que armazena todo seu conteúdo público - como publicações, curtidas e conexões sociais. Alterações nestes repositórios são organizadas em commits assinados, similar ao sistema Git, permitindo que modificações sejam facilmente verificadas e auditadas.

Figura 5 - arquitetura do ATProto. Identidades, hospedagem de dados e aplicação são camadas separadas.



Fonte: Dan Abramov (2024)

A identidade no protocolo é gerenciada através de um sistema de duas camadas que combina Identificadores Descentralizados (DIDs) imutáveis como fundamento, garantindo a autenticidade do usuário, com nomes de domínio DNS para legibilidade, em que um domínio passa a ser um "apelido" intercambiável do DID. Esta separação entre identificador técnico e nome legível permite que usuários mudem de provedor de hospedagem ou nome de exibição sem perder suas

conexões sociais. O protocolo suporta tanto DIDs auto-hospedados (did:web) quanto uma implementação própria e mais estável (did:plc), ao custo de uma maior centralização.

Para federação em larga escala, o AT Protocol implementa uma arquitetura inspirada na web (KLEPPMANN et al., 2024), onde Personal Data Servers (PDS) hospedam repositórios individuais, enquanto serviços especializados chamados Relays agregam e distribuem atualizações pela rede. Esta separação de responsabilidades permite que servidores individuais permaneçam leves e fáceis de operar, enquanto a tarefa computacionalmente intensiva de indexação é delegada a serviços dedicados.

A interoperabilidade entre diferentes implementações é garantida através dos Lexicons, um sistema rígido de schemas que define a estrutura dos dados e APIs de cada aplicação que utiliza o protocolo. Isto permite que diferentes aplicações e modos sociais coexistam na mesma infraestrutura sem necessidade de coordenação prévia, podendo ainda utilizar dados de implementações terceiras.

Esta arquitetura diferencia o AT Protocol de outras abordagens à descentralização de redes sociais através de uma hibridização estratégica. Enquanto o ActivityPub depende de comunicação direta entre servidores para propagar mudanças (LEMMER-WEBER et al., 2018), potencialmente sobrecarregando servidores individuais com múltiplas conexões, e o SSB distribui toda comunicação através da rede de usuários (TARR et al., 2019), sacrificando eficiência em nome da descentralização máxima, o AT Protocol combina elementos de ambas abordagens.

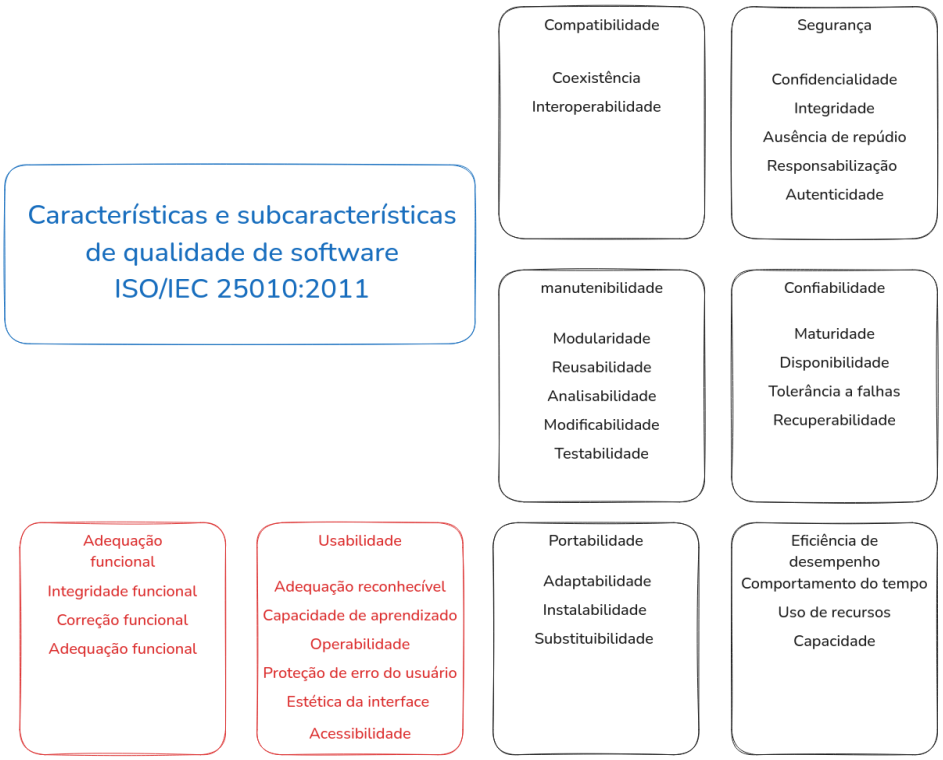
O protocolo enfrenta alguns desafios próprios. A complexidade adicional de sua arquitetura pode aumentar barreiras técnicas à implementação. A separação entre identidade e hospedagem, embora vantajosa para usuários, introduz dependências de sistemas externos como DNS. O modelo de indexação via Relays, presente na arquitetura de referência do protocolo, embora mais eficiente que federação direta, pode criar novos pontos de centralização se não houver diversidade suficiente de provedores, possibilidade real devido ao custo de operar um stream de uma rede com mais de 20 milhões de usuários (NEWBOLD, 2024).

Apesar destas limitações, o protocolo representa uma evolução significativa no design de sistemas sociais federados. Sua arquitetura em camadas e ênfase em verificabilidade criptográfica criam uma fundação mais robusta para descentralização, enquanto sua abordagem pragmática à escalabilidade facilita operação em larga escala. Em razão dessas vantagens e aparente solução a problemas enfrentados tanto pelo SSB quanto ActivityPub, o ATProto será o principal objeto de análise no capítulo 3.

2.5. Requisitos Não Funcionais e Qualidade de Software

Compreender redes sociais descentralizadas exige, além do entendimento de seus aspectos funcionais, uma análise sistemática de suas características qualitativas. Estas características, também chamadas de requisitos não funcionais (RNFs), descrevem as propriedades e restrições sob as quais o sistema deve operar, diferenciando-se das funcionalidades específicas que ele oferece por focarem no "como" o sistema deve se comportar, e não no "o que" ele deve fazer.

Figura 6: características e subcaracterísticas de software segundo a norma ISO/IEC 25010:2011. Em vermelho, as características que não serão usadas na análise



Fonte: Elaborada pelo autor com base na norma ISO/IEC 25010 (2011)

A norma ISO/IEC 25010 (2011) estabelece um modelo que categoriza as características de qualidade de software em oito dimensões principais. A **eficiência de desempenho** avalia o desempenho do sistema em relação aos recursos utilizados, considerando aspectos como tempo de resposta e utilização de recursos. A **compatibilidade** mensura a capacidade do sistema de trocar informações com outros sistemas mantendo suas funcionalidades. A **confiabilidade** determina o grau em que um sistema executa funções específicas sob condições definidas. A **segurança** avalia a proteção de informações e dados conforme níveis de autorização. A **manutenibilidade** mede a eficácia e eficiência com que o sistema pode ser modificado. Por fim, a **portabilidade** analisa a facilidade de transferência do sistema entre diferentes ambientes.

Duas características definidas no padrão não são consideradas na análise do protocolo: **adequação funcional** e **usabilidade**. A **adequação funcional** é excluída por tratar especificamente do atendimento a requisitos funcionais, enquanto o foco desta análise está nos requisitos não-funcionais. A **usabilidade**, por sua vez, opera em uma camada distinta da arquitetura de software - enquanto protocolos estabelecem padrões de comunicação e interoperabilidade, a usabilidade emerge das decisões de design e implementação das aplicações construídas sobre estes protocolos. Avaliar a usabilidade no nível do protocolo seria metodologicamente inadequado, uma vez que as mesmas especificações podem resultar em experiências de uso radicalmente diferentes dependendo de como são implementadas.

No contexto das redes descentralizadas, os protocolos desempenham um papel fundamental na garantia dessas qualidades. Embora não ditem como uma aplicação deve operar, os protocolos estabelecem regras e padrões a serem seguidos. Assim, a análise de seus requisitos não funcionais torna-se uma maneira efetiva de avaliar suas qualidades antes mesmo de serem implementados em uma aplicação. Wiegers et al. (2013) defendem, inclusive, que requisitos de qualidade servem como a origem de muitos requisitos funcionais e decisões arquiteturais e de design.

As definições detalhadas de cada subcaracterística de qualidade, conforme estabelecidas pela ISO/IEC 25010, podem ser encontradas no Anexo 1. Por sua relevância para a análise de protocolos de redes sociais descentralizadas, estas

subcaracterísticas serão utilizadas como critérios objetivos de avaliação nos capítulos subsequentes, permitindo uma análise sistemática e fundamentada do ATProto.

2.6 Considerações do capítulo

O ATProto é um protocolo bastante novo, sua revelação para o público se deu no primeiro semestre de 2023, então poder analisá-lo e compreendê-lo adequadamente exige uma contextualização robusta e que ofereça uma visão dos passos tecnológicos dados antes de sua existência que permitiram que ele fosse criado, seja de maneira direta ou indireta.

Isso se dá pela dimensão técnica, indo de definições mais antigas que a própria internet (BARAN, 1962) até de tentativas anteriores ao ATProto, com menos de 10 anos (LEMMER-WEBER et al., 2018); pela dimensão social, que compreende o que é necessário mudar no panorama de redes hoje; e também pela compreensão dos requisitos necessários para analisar adequadamente o protocolo.

Esta base teórica fornece os instrumentos necessários para, no próximo capítulo, conduzir uma análise detalhada do ATProto que vai além de seus aspectos puramente técnicos. O entendimento das diferentes dimensões da descentralização - técnica, social e qualitativa - permite uma avaliação mais completa e contextualizada de suas capacidades e limitações como fundamento para uma nova geração de redes sociais descentralizadas.

3. ANÁLISE DE CONFORMIDADE DO ATPROTO EM REQUISITOS NÃO FUNCIONAIS DE REDES SOCIAIS DESCENTRALIZADAS

Este capítulo apresenta uma análise sistemática dos requisitos não-funcionais críticos para protocolos de redes sociais descentralizadas (RSDs), utilizando o ATProto como estudo de caso. O objetivo é examinar como diferentes requisitos são interpretados e implementados em uma solução prática, e se sua implementação é suficientemente adequada frente aos objetivos de sistemas distribuídos e da descentralização de redes.

3.1 Framework para Análise de Requisitos Não-Funcionais

A identificação e análise sistemática dos requisitos não-funcionais críticos para RSDs demanda uma metodologia que combina o rigor do modelo de qualidade definido pela ISO/IEC 25010 (2011) com o entendimento dos objetivos fundamentais de sistemas distribuídos, conforme estabelecidos por Steen e Tanenbaum (2023). Esta abordagem visa identificar o conjunto mínimo de requisitos essenciais que caracterizam uma rede social verdadeiramente descentralizada, independentemente de sua implementação específica.

O processo de análise fundamenta-se em três fontes primárias de informação: especificações técnicas dos protocolos, documentação das implementações existentes e discussões arquiteturais publicadas pelas equipes de desenvolvimento. Ainda que o objeto de análise será o protocolo em si, ela será compreendida considerando três implementações significativas que representam diferentes abordagens para descentralização: Mastodon (ActivityPub), Manyverse (Secure Scuttlebutt) e Bluesky (ATProto).

A leitura dos seis objetivos de sistemas distribuídos de Steen e Tannenbaum (2023) torna evidente sua similaridade com as características de qualidade de software definidas pela norma ISO, sendo esse o ponto de partida da metodologia. Ao mapear quais subcaracterísticas de qualidade descrevem cada objetivo, criam-se critérios claros de análise, com foco adequado aos desafios inerentes a sistemas

distribuídos. Um resumo desse mapeamento pode ser visto na conclusão do capítulo.

Estabelecidos os critérios de análise através desse mapeamento, cada objetivo de sistema distribuído é examinado através de uma estrutura analítica que consiste em quatro partes: primeiro, uma explicação dos objetivos do sistema distribuído e suas subcaracterísticas de qualidade associadas no contexto de RSDs; em segundo, uma análise de como SSB e ActivityPub interpretam e implementam estas características; terceiro, a formulação dos requisitos não-funcionais que emergem tanto dessas análises quanto das considerações do panorama social discutidas na introdução; e por fim, é verificada a validade desses requisitos elaborados ao utilizá-los para realizar uma avaliação detalhada da conformidade do ATProto a eles.

Esta última etapa da estrutura analítica segue critérios objetivos de classificação do atendimento aos requisitos pelo protocolo: atendimento completo indica que o protocolo fornece mecanismos nativos que satisfazem integralmente o requisito; atendimento parcial significa que o protocolo oferece suporte limitado ou depende de implementações específicas para satisfazer completamente o requisito; e atendimento insuficiente indica ausência de mecanismos adequados para satisfazer o requisito.

Vale ressaltar que esta metodologia não busca um levantamento exaustivo de todos os requisitos não-funcionais possíveis em RSDs. Ao contrário, seu objetivo é identificar o conjunto fundamental de requisitos sem os quais uma rede social não pode ser considerada verdadeiramente descentralizada. Esta abordagem permite uma análise mais focada e relevante do ATProto, avaliando sua adequação como fundação técnica para diferentes tipos de implementações.

Por conta deste mesmo objetivo e do fato que um protocolo buscar **descrever** como aplicações devem se comunicar e não **prescrever** como aplicações devem ser implementadas, a elaboração destes requisitos não-funcionais segue uma abordagem não convencional quando comparada à literatura de engenharia de requisitos. Wiegers et al. (2013), por exemplo, defendem que se utilize o método **SMART** para determinar conformidade de um RNF - acrônimo em inglês para

específico, **mensurável**, **atingível**, **relevante** e **sensível ao tempo** -, mas esses aspectos fazem sentido no contexto de um sistema e não de um protocolo.

Levando isso em consideração, os requisitos elaborados sejam específicos, atingíveis e relevantes, características como mensurabilidade e sensibilidade temporal tornam-se menos aplicáveis. O resultado são requisitos não-funcionais mais enxutos, porém que expressam com precisão os aspectos de qualidade verdadeiramente fundamentais e mínimos para a existência de uma RSD.

3.2 Análise dos Objetivos de Sistemas Distribuídos

Nesta seção, cada objetivo fundamental será analisado seguindo a estrutura metodológica descrita na seção anterior, permitindo uma avaliação sistemática e objetiva das capacidades e limitações do ATProto no contexto de RSDs. Esta organização facilita não apenas a compreensão das interdependências entre diferentes requisitos, mas também permite uma avaliação mais coesa do protocolo como um todo.

3.2.1 Compartilhamento de recursos

O **compartilhamento de recursos** no contexto das RSDs engloba tanto os dados compartilhados entre os participantes da rede quanto os mecanismos que viabilizam essa troca de informações. Este objetivo está intrinsecamente ligado às subcaracterísticas de **interoperabilidade** e **reusabilidade** definidas pela ISO 25010, uma vez que demanda não apenas meios padronizados de troca de informações, mas também que os recursos compartilhados possam ser utilizados consistentemente por todos os participantes do sistema.

Em RSDs, estas subcaracterísticas manifestam-se de forma particular. A **interoperabilidade** traduz-se na capacidade de diferentes implementações compreenderem e processarem os dados trocados na rede sem necessidade de coordenação central. A **reusabilidade** reflete-se na possibilidade de qualquer participante estabelecer sua própria instância na rede mantendo funcionalidade plena, aproveitando a infraestrutura e os padrões existentes.

O Secure Scuttlebutt implementa estas características através de uma abordagem radical onde cada participante mantém uma cópia completa dos dados de seu grafo social. Esta estratégia garante máxima disponibilidade e verificabilidade das informações, já que cada usuário possui localmente todos os dados necessários para validar a autenticidade das mensagens. No entanto, esta abordagem resulta em significativa redundância e limitações de escalabilidade, pois o volume de dados armazenados cresce exponencialmente com o número de conexões sociais.

O ActivityPub, por sua vez, adota uma estratégia de federação direta entre servidores, onde cada instância mantém os dados de seus usuários e os compartilha sob demanda com outras instâncias. Esta abordagem é mais eficiente em termos de recursos computacionais, mas cria dependências da disponibilidade contínua dos servidores para acesso às informações. A federação é viabilizada pelo formato ActivityStreams, que padroniza a estrutura das interações sociais, permitindo que diferentes implementações interpretem os dados consistentemente.

Da análise destas implementações emergem dois requisitos não-funcionais essenciais para o compartilhamento efetivo de recursos em RSDs:

1. ***"A RSD deve implementar um modelo de dados padronizado e verificável que permita armazenamento, distribuição e interpretação consistente de informações entre diferentes implementações."***
2. ***"A RSD deve permitir que diferentes tipos de recursos sejam distribuídos de forma independente entre provedores."***

O ATProto aborda estes requisitos através de uma arquitetura que separa claramente a hospedagem de dados da infraestrutura de rede e aplicações. Para garantir **interoperabilidade** entre diferentes implementações, o protocolo define um sistema formal de schemas chamado Lexicons, que não apenas padroniza a estrutura dos dados, mas também estabelece regras claras para evolução do protocolo. A autoridade sobre cada Lexicon é determinada pelo controle do domínio DNS correspondente, criando um mecanismo descentralizado de governança.

Além disso, a Bluesky PBC disponibiliza em código aberto todas as implementações do protocolo que utilizam em sua própria infraestrutura. Como mencionado anteriormente, isso de certa forma faz parte da arquitetura da rede social em si,

mas trata-se da implementação de recursos essenciais ao protocolo em linguagens de programação específicas, seguindo os mesmos princípios técnicos exigidos em qualquer implementação.

Esta arquitetura permite que o ATProto atenda completamente ambos os requisitos identificados. O primeiro é satisfeito através do sistema Lexicon, que garante interpretação consistente dos dados por todas as implementações. O segundo é atendido pela clara separação entre hospedagem de dados, infraestrutura de rede e aplicações, permitindo que diferentes provedores se especializem em diferentes aspectos do sistema.

Esta abordagem representa um avanço significativo em relação às implementações anteriores, oferecendo um equilíbrio entre a redundância excessiva do SSB e a dependência de servidores do ActivityPub. Por permitir evolução independente de diferentes aspectos do protocolo através dos Lexicons, ao mesmo tempo em que mantém a **interoperabilidade** através de schemas rigorosamente definidos, o protocolo estabelece fundações sólidas para um ecossistema verdadeiramente descentralizado de redes sociais.

3.2.2 Invisibilidade na distribuição

A **invisibilidade na distribuição** representa um dos objetivos mais desafiadores em RSDs: fazer com que a natureza distribuída do sistema seja imperceptível para seus usuários. Este objetivo relaciona-se diretamente com as subcaracterísticas de **adaptabilidade**, **instalabilidade** e **substituibilidade**, uma vez que demanda que os usuários possam migrar entre diferentes implementações e provedores sem impacto em sua experiência ou conexões sociais.

No contexto de RSDs, a **adaptabilidade** manifesta-se na capacidade do sistema funcionar de forma consistente em diferentes ambientes e dispositivos. A **instalabilidade** traduz-se na facilidade com que usuários podem começar a utilizar o sistema ou migrar entre provedores, enquanto a **substituibilidade** reflete a capacidade de trocar componentes da infraestrutura sem afetar a experiência do usuário final.

As implementações existentes demonstram diferentes interpretações deste desafio. O Secure Scuttlebutt, embora ofereça alta redundância através de sua arquitetura peer-to-peer, expõe significativamente sua natureza distribuída aos usuários, que precisam lidar com conceitos como sincronização de pares e replicação de dados. Esta exposição, embora tecnicamente precisa, cria barreiras significativas à adoção por usuários não técnicos. Essa questão é ainda agravada pelas chaves do usuário serem atreladas ao dispositivo onde configurou, não sendo possível acessar a mesma rede em diversos dispositivos.

O ActivityPub busca maior invisibilidade através de sua arquitetura federada, onde servidores coordenam-se para oferecer uma experiência mais integrada. No entanto, sua abordagem ainda expõe aspectos de distribuição aos usuários, que precisam escolher servidores específicos e lidar com limitações na comunicação entre diferentes instâncias determinadas pelos administradores de sua instância. A identidade do usuário permanece intrinsecamente ligada ao servidor escolhido, tornando migrações entre provedores um processo complexo e bastante visível.

Da análise destas implementações, emergem dois requisitos não-funcionais essenciais para invisibilidade na distribuição em RSDs e que não são completamente alcançados pelos protocolos mencionados:

3. ***"A RSD deve permitir que usuários transfiram a totalidade de seus dados e conexões sociais entre provedores sem interrupção de serviço."***
4. ***"A RSD deve abstrair a complexidade da infraestrutura distribuída, apresentando uma interface unificada independente da implementação específica utilizada."***

O ATProto aborda estes requisitos através de sua arquitetura em camadas e sistema de identidades descentralizadas. Como discutido no tópico 2.2, o protocolo separa claramente identidade, armazenamento e aplicação, permitindo que falhas em componentes individuais sejam isoladas sem comprometer a experiência global. O sistema de identificadores descentralizados (DIDs) desacopla a identidade da infraestrutura, permitindo que usuários migrem entre provedores mantendo suas conexões sociais intactas.

No entanto, a atual implementação do protocolo apresenta uma limitação significativa em termos de **substituibilidade** através do método did:plc, que é administrado exclusivamente pela Bluesky. Esta centralização parcial poderia comprometer a capacidade dos usuários de migrar livremente entre provedores caso a empresa deixasse de operar este serviço. Embora exista planejamento para que seja gerido por um consórcio de certificadores de confiança no futuro (KLEPPMANN et al., 2024), até que esta transição ocorra, a gestão centralizada de DIDs permanece um ponto de fragilidade na arquitetura do protocolo.

O protocolo mitiga parcialmente este risco ao oferecer suporte ao método did:web como alternativa totalmente descentralizada, permitindo que usuários mantenham controle completo sobre suas identidades através de domínios próprios. No entanto, esta solução apresenta suas próprias limitações: usuários precisam escolher entre os dois métodos no momento da criação da conta, não sendo possível alternar posteriormente, e o did:web não possui mecanismos nativos para recuperação ou migração em caso de perda do acesso ao domínio.

Com essa análise, observa-se que o ATProto atende parcialmente os requisitos não-funcionais identificados: A portabilidade é tecnicamente possível através da separação entre identidade e infraestrutura, mas sua efetividade depende criticamente do método DID escolhido pelo usuário. A abstração da complexidade, por sua vez, é alcançada através da arquitetura em camadas, que permite experiências consistentes independentemente da implementação utilizada, porém ao custo de uma dependência temporária da infraestrutura centralizada da Bluesky para a maioria dos usuários que optam pelo did:plc.

Esta abordagem, embora represente uma evolução significativa em relação às implementações anteriores, evidencia o desafio fundamental de balancear descentralização com usabilidade em redes sociais distribuídas. O protocolo estabelece fundações técnicas que permitem experiências verdadeiramente integradas, mas sua implementação atual ainda depende de compromissos que podem impactar a autonomia dos usuários dependendo de suas escolhas iniciais de identificação.

3.2.3 Abertura para integração

A **abertura para integração** em RSDs representa um objetivo fundamental que vai além da capacidade de conexão entre sistemas distintos. Este objetivo relaciona-se diretamente com as subcaracterísticas de **modularidade**, **modificabilidade** e **adaptabilidade**, demandando que o protocolo não apenas permita a integração de novos componentes, mas também suporte sua evolução contínua sem comprometer a interoperabilidade existente.

Nas RSDs, a **modularidade** reflete-se na capacidade do sistema de isolar diferentes aspectos funcionais em componentes independentes. A **modificabilidade** traduz-se na facilidade com que estes componentes podem ser atualizados ou substituídos sem afetar o funcionamento do sistema como um todo. A **adaptabilidade** representa a capacidade do sistema de incorporar novas funcionalidades e casos de uso não previstos inicialmente.

O Secure Scuttlebutt implementa uma arquitetura minimalista onde a abertura é alcançada através de um protocolo base extremamente simples, sobre o qual diferentes aplicações podem ser construídas. Esta abordagem oferece flexibilidade para implementações, mas resulta em fragmentação significativa do ecossistema pela ausência de mecanismos formais para garantir compatibilidade entre diferentes extensões do protocolo.

O ActivityPub, por sua vez, oferece um conjunto robusto de interações padronizadas através do formato ActivityStreams. No entanto, sua arquitetura monolítica apresenta limitações significativas em termos de **modularidade** - a ausência de separação clara entre as camadas de hospedagem e aplicação significa que novas funcionalidades frequentemente requerem modificações em múltiplos níveis.

Considerando estas limitações, emergem dois requisitos não-funcionais essenciais para abertura à integração em RSDs:

5. ***"A RSD deve oferecer interfaces claramente definidas que permitam a integração de novos componentes sem afetar os existentes."***

6. ***"A RSD deve permitir a evolução independente de diferentes aspectos do protocolo sem comprometer a interoperabilidade entre implementações existentes."***

O ATProto aborda estes requisitos através de um sistema chamado Lexicon, que estabelece schemas rigorosamente definidos com regras claras de evolução. Cada Lexicon é identificado por uma nomenclatura vinculada a um domínio DNS específico, onde apenas o controlador daquele domínio pode definir ou modificar schemas com aquela nomenclatura.

Para garantir estabilidade e compatibilidade, os Lexicons implementam regras estritas de versionamento: campos existentes não podem ser modificados ou removidos, apenas novos campos podem ser adicionados. Quando mudanças incompatíveis são necessárias, um novo Lexicon deve ser criado com um identificador de versão diferente, permitindo que implementações antigas continuem funcionando enquanto novas funcionalidades são introduzidas.

Esta análise demonstra que o ATProto atende completamente aos dois requisitos não-funcionais identificados. A integração de novos componentes é facilitada pela clara separação de responsabilidades através dos Lexicons, enquanto a evolução independente é garantida pelo sistema de versionamento que permite que diferentes aspectos do protocolo evoluam em ritmos distintos sem comprometer a interoperabilidade do sistema como um todo.

3.2.4 Confiabilidade

A confiabilidade em RSDs manifesta-se como a capacidade do sistema de manter sua operação e integridade mesmo diante de falhas parciais em seus componentes. Este objetivo relaciona-se diretamente com as subcaracterísticas de **disponibilidade**, **tolerância a falhas** e **recuperabilidade**, demandando que o sistema não apenas continue operacional durante falhas, mas também preserve a consistência e rastreabilidade das informações.

No contexto específico de RSDs, estas subcaracterísticas manifestam-se de forma particular: a **disponibilidade** determina a probabilidade do sistema estar operando corretamente em um dado momento; a **tolerância a falhas** representa a capacidade

do sistema de manter a operação mesmo quando componentes falham; e a **recuperabilidade** indica a habilidade do sistema de retornar ao estado normal após uma falha. Um sistema distribuído robusto deve equilibrar estas três características, permitindo que falhas localizadas não comprometam o funcionamento global do sistema.

Em seu artigo sobre protocolos descentralizados, Graber (2020) destaca como o Secure Scuttlebutt e o ActivityPub adotam estratégias fundamentalmente diferentes para garantir confiabilidade. O SSB implementa **tolerância a falhas** através de replicação peer-to-peer, onde cada nó mantém cópias dos dados de seus contatos. Esta abordagem garante alta **recuperabilidade**, pois os dados podem ser restaurados a partir de qualquer *peer* conectado, e permite que o sistema continue operando mesmo com conectividade intermitente ou falhas em nós específicos. A **disponibilidade** de aplicações SSB também se destaca, já que podem continuar operando mesmo quando a internet está completamente indisponível, desde que exista conexão local entre dispositivos.

O ActivityPub, por sua vez, adota uma abordagem de federação onde cada servidor mantém autonomia sobre seus dados e políticas. A **tolerância a falhas** é implementada no nível de instância - se um servidor específico falha, apenas seus usuários diretos são afetados enquanto o resto da rede continua operacional. A **recuperabilidade** depende das políticas de backup de cada servidor, e a **disponibilidade** do serviço para um usuário está intrinsecamente ligada à operação de sua instância escolhida. Esta arquitetura simplifica a administração distribuída, mas cria pontos únicos de falha para grupos específicos de usuários.

Considerando estes desafios, emergem dois requisitos não-funcionais essenciais para confiabilidade em RSDs:

7. ***"A RSD deve continuar funcional mesmo com a indisponibilidade temporária de parte de seus componentes."***
8. ***"A RSD deve garantir integridade e rastreabilidade das informações mesmo em cenários de falha parcial."***

O ATProto aborda estes requisitos através de uma arquitetura que separa claramente as responsabilidades entre seus componentes. Os Personal Data

Servers (PDS) são responsáveis por hospedar repositórios individuais, enquanto serviços especializados como Relays - que são uma camada de otimização não obrigatória - e App Views gerenciam a distribuição e processamento dos dados. Esta separação permite que falhas em componentes individuais sejam isoladas sem comprometer a operação global do sistema.

Esta abordagem representa um avanço significativo em relação às implementações anteriores, oferecendo um equilíbrio entre a alta disponibilidade do SSB e a autonomia administrativa do ActivityPub, enquanto mitiga suas respectivas limitações de escalabilidade e dependência de servidores específicos. A análise demonstra que o ATProto atende completamente aos requisitos não-funcionais identificados, estabelecendo bases sólidas para redes sociais verdadeiramente descentralizadas e confiáveis.

3.2.5 Segurança

A segurança em RSDs relaciona-se diretamente com as subcaracterísticas de **confidencialidade**, **integridade**, **autenticidade**, **não-repúdio** e **responsabilização**. No contexto das RSDs, estas subcaracterísticas manifestam-se de formas específicas: a **confidencialidade** garante que apenas usuários autorizados possam acessar informações sensíveis; a **integridade** assegura que dados não sejam alterados sem autorização durante seu trânsito ou armazenamento; a **autenticidade** permite verificar a identidade de usuários e origem das mensagens; o **não-repúdio** impede que usuários neguem ações realizadas; e a **responsabilização** permite rastrear e atribuir ações a atores específicos na rede.

O Secure Scuttlebutt implementa segurança primáriamente através de criptografia assimétrica, onde cada usuário possui um par de chaves que define sua identidade e permite assinar suas mensagens. Esta abordagem garante **confidencialidade** através de criptografia ponta-a-ponta, **integridade** e **autenticidade** via assinaturas digitais, e **não-repúdio** pela natureza imutável do log de eventos. No entanto, a complexidade da gestão de chaves torna a experiência desafiadora para usuários não técnicos.

O ActivityPub, por sua vez, delega aspectos de segurança aos servidores federados. A **confidencialidade** é mantida através de canais TLS entre servidores; a **integridade** e **autenticidade** dependem das práticas de segurança de cada instância; o **não-repúdio** é limitado aos registros mantidos pelos servidores; e a **responsabilização** ocorre primariamente no nível de instância. Esta arquitetura simplifica a experiência do usuário ao custo de criar dependências significativas dos servidores escolhidos.

Considerando estas implementações, emergem três requisitos não-funcionais essenciais para segurança em RSDs:

9. ***"A RSD deve garantir que toda informação distribuída tenha sua autenticidade e integridade verificáveis sem depender de autoridade central."***
10. ***"A RSD deve implementar mecanismos que garantam que apenas usuários autorizados possam modificar ou publicar conteúdo em seu nome."***
11. ***"A RSD deve fornecer mecanismos para verificação da origem e não-repúdio de todas as ações realizadas na rede."***

O ATProto aborda estes requisitos através de uma arquitetura que combina identificadores descentralizados (DIDs) com repositórios pessoais criptograficamente verificáveis. Cada usuário possui um DID que está associado a um par de chaves públicas/privadas, onde a chave privada é utilizada para assinar atualizações em seu repositório, enquanto a chave pública, disponível no documento DID, permite que qualquer participante da rede verifique estas assinaturas.

Para garantir integridade dos dados ao longo do tempo, o protocolo organiza todas as informações em Merkle Search Trees, onde cada alteração gera um novo commit assinado que inclui referências criptográficas ao estado anterior. Esta estrutura permite que qualquer participante da rede verifique não apenas a autenticidade de uma informação específica, mas também sua posição na história completa do repositório e todas as modificações realizadas.

O protocolo implementa ainda um sistema robusto de controle de acesso onde cada alteração em um repositório deve ser acompanhada de uma assinatura válida de uma chave autorizada no documento DID correspondente. Isto impede que publicações sejam feitas em nome de um usuário sem sua autorização, mesmo em um ambiente descentralizado.

Esta análise demonstra que o ATProto atende completamente aos requisitos não-funcionais identificados, estabelecendo uma fundação técnica que combina garantias criptográficas robustas com usabilidade adequada para usuários finais.

3.2.6 Escalabilidade

No contexto das RSDs, a escalabilidade engloba tanto aspectos técnicos quanto organizacionais. Este objetivo relaciona-se diretamente com as subcaracterísticas de **comportamento temporal**, **capacidade**, **substituibilidade** e **adaptabilidade**, exigindo que o sistema não apenas acomode crescimento em volume de dados e usuários, mas também suporte a coexistência de múltiplas organizações administrativas independentes.

O **comportamento temporal** traduz-se na capacidade do sistema de manter tempos de resposta aceitáveis mesmo com aumento significativo de carga. A **capacidade** reflete-se na habilidade de processar volumes crescentes de dados e interações sem degradação perceptível. A **substituibilidade** e **adaptabilidade**, por sua vez, representam a flexibilidade necessária para que diferentes organizações possam operar partes do sistema de forma independente, mantendo a interoperabilidade do conjunto.

O Secure Scuttlebutt (SSB) enfrenta desafios fundamentais de escalabilidade à medida que a rede cresce, pois cada participante precisa armazenar e processar uma porção significativa dos dados totais do sistema. Embora ofereça alta redundância, esta abordagem resulta em limitações práticas de crescimento.

O ActivityPub, por sua vez, apresenta problemas de coordenação entre instâncias que se intensificam com o crescimento da rede. A federação direta entre servidores pode resultar em inconsistências na propagação de atualizações e sobrecarga de servidores populares.

Da análise destas implementações, emergem três requisitos não-funcionais essenciais para escalabilidade em RSDs:

12. ***A RSD deve suportar crescimento sustentável no número de usuários e volume de dados***
13. ***A RSD deve permitir que diferentes organizações administrativas operem de forma independente sob políticas próprias, mantendo a interoperabilidade.***
14. ***A propagação de atualizações deve ocorrer de forma eficiente para alcançar os participantes da rede em condições normais de operação.***

O ATProto aborda estes requisitos através de uma arquitetura em camadas que separa claramente as responsabilidades de armazenamento, distribuição e processamento de dados. Os Personal Data Servers (PDS) que hospedam repositórios individuais são intencionalmente leves, permitindo operação mesmo com recursos computacionais modestos. A distribuição de atualizações é gerenciada por Relays especializados que agregam e transmitem mudanças em tempo real, enquanto App Views processam e organizam estas informações para consumo por aplicações.

Considerando que o Bluesky já alcançou 25 milhões de usuários (dezembro de 2024), os desafios de escalabilidade estão latentes. Em julho, com 6 milhões de usuários, operar um relay que que atendesse toda a rede custava 153 dólares por mês (NEWBOLD, 2024); um preço alto, mas não proibitivo. No entanto, com os números atuais, se o valor tiver escalado proporcionalmente, há preocupações quanto à viabilidade de relays independentes no longo prazo. O custo e complexidade de operar um relay completo torna-se cada vez mais proibitivo para desenvolvedores individuais e pequenas organizações.

Para mitigar esta limitação, o projeto Jetstream foi desenvolvido oficialmente como uma alternativa mais leve e acessível ao firehose completo. Ele oferece um stream de dados com custo operacional reduzido ao abrir mão de alguns mecanismos de autenticação. No entanto, essa abordagem vem com suas próprias contrapartidas, como a perda de verificabilidade dos dados. Embora seja possível começar a construir aplicações sem depender de um relay massivo, substituir completamente a

infraestrutura do Bluesky de forma independente permanece um desafio econômico significativo.

Nesse contexto de alta escala, para garantir propagação eficiente, o protocolo implementa um sistema de eventos em tempo real via WebSocket. Este mecanismo inclui suporte para preenchimento retroativo, permitindo que clientes recuperem atualizações perdidas durante desconexões temporárias sem necessidade de ressincronização completa.

Essa análise detalhada demonstra que o ATProto atende completamente aos requisitos 2 e 3, relativos à operação independente de organizações e propagação eficiente de atualizações. No entanto, atende apenas parcialmente ao requisito 1, sobre crescimento sustentável. Embora a arquitetura em camadas permita escalonamento independente de diferentes aspectos, favorecendo crescimento, os custos crescentes associados à operação de relays completos representam uma barreira significativa para organizações menores, limitando a descentralização efetiva a longo prazo.

Em suma, com relação à escalabilidade, o ATProto apresenta uma arquitetura promissora para acomodar crescimento, mas enfrenta desafios para manter descentralização efetiva em larga escala. Embora estabeleça uma base sólida, mais inovações serão necessárias para que o AT Protocol realize plenamente seu potencial como fundação para redes sociais verdadeiramente descentralizadas e democráticas.

3.3 Resumo da análise

Tabela 2: Resumo da análise de conformidade do ATProto aos requisitos não funcionais de uma rede social descentralizada. "+" significa conformidade total e "O" conformidade parcial.

Objetivo do Sistema Distribuído	Subcaracterísticas ISO 25010	Requisito não-funcional	ATProto
Compartilhamento de recursos	Interoperabilidade, reusabilidade	A RSD deve implementar um modelo de dados padronizado e verificável que permita armazenamento, distribuição e interpretação consistente de informações entre diferentes implementações.	+
		A RSD deve permitir que diferentes tipos de recursos sejam distribuídos de forma independente entre provedores.	+
Invisibilidade na distribuição	adaptabilidade, instalabilidade e substituibilidade	A RSD deve abstrair a complexidade da infraestrutura distribuída, apresentando uma interface unificada independente da implementação específica utilizada.	+
		A RSD deve permitir que usuários transfiram a totalidade de seus dados e conexões sociais entre provedores sem interrupção de serviço.	O
Abertura para integração	Modularidade, Modificabilidade, Adaptabilidade	A RSD deve oferecer interfaces claramente definidas que permitam a integração de novos componentes sem afetar os existentes.	+
		A RSD deve permitir a evolução independente de diferentes aspectos do protocolo sem comprometer a interoperabilidade entre implementações existentes.	+
Confiabilidade	Disponibilidade, Tolerância a falhas, Recuperabilidade	A RSD deve continuar funcional mesmo com a indisponibilidade temporária de parte de seus componentes.	+
		A RSD deve garantir integridade e rastreabilidade das informações mesmo em cenários de falha parcial.	+
Segurança	Confidencialidade, Integridade, Autenticidade, Responsabilização, não-repúdio	A RSD deve garantir que toda informação distribuída tenha sua autenticidade e integridade verificáveis sem depender de autoridade central.	+
		A RSD deve implementar mecanismos que garantam que apenas usuários autorizados possam modificar ou publicar conteúdo em seu nome.	+
		A RSD deve fornecer mecanismos para verificação da origem e não-repúdio de todas as ações realizadas na rede.	+
Escalabilidade	Comportamento temporal, Capacidade, Substituibilidade, Adaptabilidade	A RSD deve suportar crescimento sustentável no número de usuários e volume de dados	O
		A RSD deve permitir que diferentes organizações administrativas operem de forma independente sob políticas próprias, mantendo a interoperabilidade.	+
		A propagação de atualizações deve ocorrer de forma eficiente para alcançar os participantes da rede em condições normais de operação.	+

Fonte: elaborada pelo autor

Este capítulo teve como objetivo central analisar de forma sistemática a adequação do ATProto como fundação para RSDs. Para isso, foi desenvolvida uma metodologia que combinou as características de qualidade de software definidas pela norma ISO 25010 (2011) com os objetivos fundamentais de sistemas distribuídos estabelecidos por Steen e Tanenbaum (2023). Esta abordagem permitiu identificar um conjunto de requisitos não-funcionais críticos para RSDs e avaliar o grau de conformidade do ATProto a cada um deles.

A análise detalhada, resumida na tabela comparativa apresentada no início do capítulo, demonstrou que o ATProto estabelece uma base sólida para RSDs, com avanços significativos em relação a implementações anteriores como o Secure Scuttlebutt e o ActivityPub. O protocolo apresentou alto grau de conformidade com a maioria dos requisitos estabelecidos.

No entanto, a análise também identificou limitações importantes que precisam ser trabalhadas para que o protocolo possa servir como base para redes sociais completas e verdadeiramente descentralizadas. Os desafios de escalabilidade a longo prazo, evidenciados pelos custos crescentes de operação de relays completos, representam um risco para a descentralização efetiva da rede em larga escala. Além disso, a centralização parcial introduzida pelo método did:plc para identidades descentralizadas cria uma dependência temporária da infraestrutura do Bluesky que pode afetar a autonomia dos usuários.

Para além dos requisitos não-funcionais avaliados, é importante mencionar que a ausência de uma camada de comunicação privativa no protocolo também representa uma barreira significativa para o ATProto atingir seu objetivo de que todas interações realizadas entre usuários em aplicações baseadas sejam verdadeiramente descentralizadas.

Interações privadas, como mensagens diretas e a opção de tornar perfis visíveis apenas para seguidores aprovados, são funcionalidades críticas para diversos usuários e comunidades. Atualmente, o Bluesky implementa mensagens diretas através de seus próprios servidores centralizados, o que introduz uma barreira de saída para usuários que optem por deixar a rede. Essa dependência vai contra os princípios de portabilidade e autonomia do usuário que o protocolo busca promover.

É importante ressaltar que o ATProto ainda está em desenvolvimento ativo e que a equipe responsável reconhece as limitações identificadas nesta análise. Esforços já estão em andamento para endereçar lacunas críticas, como a descentralização do método did:plc (KLEPPMANN et al., 2024) e a implementação de uma camada de comunicação privada (BLUESKY PBC, 2024). Embora não haja prazos definitivos para estas atualizações, o compromisso da equipe em aprimorar continuamente o protocolo é um sinal positivo para seu futuro como base para RSDs.

Em conclusão, a análise apresentada neste capítulo demonstra que, apesar de limitações importantes, o ATProto representa um avanço significativo em direção a redes sociais verdadeiramente descentralizadas. Sua arquitetura modular, esquemas de verificação criptográfica e separação clara entre hospedagem e aplicação estabelecem uma fundação técnica robusta sobre a qual diferentes implementações e casos de uso podem ser construídos.

4. Análise de resultados

A análise dos resultados deste trabalho revela algumas descobertas significativas, tanto em termos metodológicos quanto em relação aos desafios práticos de estudar um protocolo em rápida evolução. A abordagem interdisciplinar adotada, combinando conceitos de sistemas distribuídos com análise de requisitos não-funcionais, proporcionou perspectivas valiosas que merecem reflexão detalhada.

Uma das descobertas mais relevantes foi a surpreendente escassez de diálogo entre a literatura acadêmica sobre sistemas distribuídos e aquela dedicada a redes sociais descentralizadas. Embora estas redes sejam, por definição, sistemas distribuídos, as publicações em cada área raramente fazem referência uma à outra. No entanto, ao aplicar os princípios fundamentais de sistemas distribuídos estabelecidos por Steen e Tanenbaum (2023) à análise de redes sociais descentralizadas, observou-se uma correspondência notável. Os desafios arquiteturais, objetivos de design e compromissos necessários descritos na literatura de sistemas distribuídos mostraram-se diretamente aplicáveis e extremamente úteis para compreender as diferentes abordagens à descentralização de redes sociais.

A metodologia desenvolvida para análise de requisitos não-funcionais também merece discussão crítica. Conforme discutido na seção 3.1, a criação de requisitos não-funcionais tem como objetivo trazer objetivos mensuráveis e detalhados para a qualidade de um sistema, o que não foi possível ao falar da camada de protocolo. No entanto, seria interessante expandir os requisitos desenvolvidos para que contemplem esses aspectos no desenvolvimento das redes em si.

O tempo de existência do ATProto também apresentou desafios metodológicos significativos para esta pesquisa. Com menos de dois anos desde seu lançamento público, a literatura acadêmica sobre o protocolo mostrou-se extremamente limitada, consistindo principalmente do whitepaper original publicado pela equipe da Bluesky (KLEPPMANN et al., 2024) e um estudo focado em análise de dados da plataforma (BALDUF et al., 2024) que, embora relevante em seu contexto, ofereceu contribuições limitadas para uma análise arquitetural do protocolo.

Um desafio particular deste trabalho foi acompanhar e incorporar a rápida evolução do objeto de estudo. Durante o período de desenvolvimento da pesquisa, o Bluesky - principal implementação do ATProto - cresceu de 6 milhões para 25 milhões de usuários. Este crescimento acelerado não apenas validou a relevância do estudo, mas também trouxe à tona novos desafios e limitações do protocolo que precisaram ser incorporados à análise. Questões como a escalabilidade dos relays e a ausência de uma camada de privacidade tornaram-se mais relevantes conforme a base de usuários expandia.

Esta dinâmica de desenvolvimento ativo do protocolo apresentou tanto oportunidades quanto desafios metodológicos. Por um lado, permitiu observar em tempo real como diferentes aspectos do protocolo se comportavam sob pressão crescente, fornecendo dados valiosos para a análise. Por outro, exigiu revisões recorrentes de trechos para refletirem adequadamente o momento.

Por fim, a decisão de adotar uma abordagem que combina rigor teórico com observação prática mostrou-se acertada. O framework desenvolvido não apenas permitiu uma análise abrangente do ATProto em seu estado atual, mas também estabeleceu bases metodológicas que podem ser aplicadas à avaliação de outros protocolos de redes sociais descentralizadas.

5. CONSIDERAÇÕES FINAIS

5.1 Conclusões

Esta pesquisa teve como objetivo principal analisar a adequação do ATProto como fundação para redes sociais descentralizadas, desenvolvendo um framework metodológico que permitisse avaliar seus requisitos não-funcionais de forma sistemática. Os resultados evidenciam não apenas os méritos técnicos do protocolo, mas também sua relevância como resposta aos desafios sociais causados pelas redes sociais contemporâneas discutidos no primeiro capítulo.

A análise demonstrou que o protocolo representa um avanço significativo em direção à descentralização efetiva de redes sociais, oferecendo respostas concretas a questões como concentração de poder, controle sobre dados pessoais e transparência algorítmica. Sua arquitetura permite que usuários mantenham propriedade sobre seus dados enquanto participam de uma rede social interoperável, respondendo diretamente às preocupações levantadas por pesquisadores como Barabas, Narula e Zuckerman (2017) sobre a necessidade de redistribuir poder em plataformas digitais.

A abordagem metodológica desenvolvida, combinando conceitos de sistemas distribuídos com análise de requisitos não-funcionais, provou-se particularmente valiosa para compreender e avaliar protocolos de comunicação social. A integração dos objetivos fundamentais de sistemas distribuídos com as características de qualidade definidas pela ISO 25010 ofereceu um framework robusto para análise, capaz de capturar tanto aspectos técnicos quanto implicações práticas do protocolo.

Os resultados indicam que o ATProto atende completamente à maioria dos requisitos não-funcionais identificados como críticos para redes sociais descentralizadas. Sua arquitetura em camadas, sistema de identidades descentralizadas e mecanismos de verificação criptográfica estabelecem uma fundação técnica robusta que permite diferentes implementações e casos de uso. No entanto, desafios significativos permanecem, particularmente em relação à escalabilidade a longo prazo e à necessidade de uma camada de privacidade integrada ao protocolo.

5.2 Contribuições do Trabalho

A principal contribuição deste trabalho é a análise sistemática e focada no Authenticated Transfer Protocol, consolidando conhecimento anteriormente disperso em diversas fontes - incluindo documentações técnicas, análises críticas de especialistas e discussões da equipe de desenvolvimento. Esta consolidação oferece uma visão abrangente do protocolo, documentando não apenas suas capacidades técnicas, mas também os compromissos arquiteturais e desafios práticos envolvidos em sua implementação.

Sendo o ATProto um projeto de código aberto em desenvolvimento ativo, pretende-se que o conhecimento consolidado neste trabalho contribua para aprimorar sua documentação oficial. Durante a pesquisa, foram identificadas diversas oportunidades de melhoria na documentação existente, e a análise feita aqui poderá beneficiar tanto desenvolvedores quanto pesquisadores interessados em compreender e implementar o protocolo.

5.3 Trabalhos Futuros

A partir das descobertas e limitações identificadas nesta pesquisa, diversos caminhos para trabalhos futuros se apresentam. Uma área particularmente relevante seria o desenvolvimento de métricas quantitativas para avaliar o grau de descentralização efetiva em implementações do protocolo.

Outra linha de investigação importante seria o estudo comparativo do comportamento do protocolo em diferentes escalas de implementação, analisando como os compromissos entre descentralização e eficiência se manifestam em redes de diferentes tamanhos.

Por fim, pesquisas futuras poderiam focar no desenvolvimento e avaliação de soluções para as limitações identificadas no protocolo, particularmente em relação à implementação de uma camada de privacidade integrada e mecanismos mais eficientes para distribuição de atualizações em larga escala. A evolução contínua do protocolo e sua crescente adoção oferecem oportunidades valiosas para estudos que contribuam diretamente para seu aprimoramento.

REFERÊNCIAS

ABRAMOV, Dan. **dan (@danabra.mov)**. 2024. Disponível em: https://bsky.app/profile/danabra.mov/post/3l7oxg72zd22t. Acesso em: 14 dez. 2024.

AUVOLAT, Alex; TAIANI, Francois. Merkle Search Trees: Efficient State-Based CRDTs in Open Networks. *Em*: 2019, Lyon, France. **Anais** [...]. . *Em*: 38TH IEEE INTERNATIONAL SYMPOSIUM ON RELIABLE DISTRIBUTED SYSTEMS. Lyon, France: IEEE, 2019. DOI: 10.1109/SRDS47363.2019.00032. Disponível em: https://ieeexplore.ieee.org/document/9049566/. Acesso em: 17 nov. 2024.

BALDUF, Leonhard; SOKOTO, Saidu; ASCIGIL, Onur; TYSON, Gareth; SCHEUERMANN, Björn; KORCZYŃSKI, Maciej; CASTRO, Ignacio; KRÓL, Michał. **Looking AT the Blue Skies of Bluesky**. arXiv, , 2024. Disponível em: http://arxiv.org/abs/2408.12449. Acesso em: 3 nov. 2024.

BARABAS, Chelsea; NARULA, Neha; ZUCKERMAN, Ethan. Defending Internet Freedom through Decentralization: Back to the Future? **The Center for Civic Media & The Digital Currency Initiative**, *[S. l.]*, 2017.

BARAN, Paul. On Distributed Communications Networks. *Em*: 1962, Santa Monica. **Anais** [...]. Santa Monica: RAND Corporation, 1962. Disponível em: https://www.rand.org/pubs/papers/P2626.html.

BLUESKY PBC. **Federation Architecture Overview**. **Bluesky**, 2023. Disponível em: https://bsky.social/about/blog/5-5-2023-federation-architecture. Acesso em: 7 dez. 2024.

BLUESKY PBC. **2024 Protocol Roadmap**. 2024. Disponível em: https://docs.bsky.app/blog/2024-protocol-roadmap. Acesso em: 27 nov. 2024.

BROWN, Sara. The case for new social media business models. **MIT Sloan**, *[S. l.]*, 2021. Disponível em: https://mitsloan.mit.edu/ideas-made-to-matter/case-new-social-media-business-models. Acesso em: 14 dez. 2024.

GRABER, Jay. **Decentralized Social Networks**. **Stories from the Decentralized Web**, 2020. Disponível em:

https://medium.com/decentralized-web/decentralized-social-networks-e5a7a2603f53. Acesso em: 7 dez. 2024.

HORWITZ, Jeff; SEETHARAMAN, Deepa. Facebook Executives Shut Down Efforts to Make the Site Less Divisive. **Wall Street Journal**, *[S. l.]*, 2020. Disponível em: https://www.wsj.com/articles/facebook-knows-it-encourages-division-top-executives-nixed-solutions-11590507499. Acesso em: 2 dez. 2024.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION; INTERNATIONAL ELECTROTECHNICAL; COMMISSION. **Systems and software engineering. Systems and software quality requirements and evaluation (SQuaRE). System and software quality models:** BSI British Standards, , 2011. DOI: 10.3403/30215101. Disponível em: https://linkresolver.bsigroup.com/junction/resolve/00000000030215101?restype=standard. Acesso em: 1 dez. 2024.

KEMP, Simon. **Digital 2024: Global Overview Report**. [s.l.] : DataReportal, 2024. Disponível em: https://datareportal.com/reports/digital-2024-global-overview-report. Acesso em: 7 dez. 2024.

KLEPPMANN, Martin; FRAZEE, Paul; GOLD, Jake; GRABER, Jay; HOLMGREN, Daniel; IVY, Devin; JOHNSON, Jeromy; NEWBOLD, Bryan; VOLPERT, Jaz. **Bluesky and the AT Protocol: Usable Decentralized Social Media**. , 2024. DOI: 10.1145/3694809.3700740. Disponível em: http://arxiv.org/abs/2402.03239. Acesso em: 2 nov. 2024.

LEMMER-WEBER, Christine; TALLON, Jessica; SHEPHERD, Erin; GUY, Amy; PRODROMOU, Evan. **ActivityPub**. W3C, , 2018. Disponível em: https://www.w3.org/TR/activitypub/. Acesso em: 2 nov. 2024.

MASNICK, Mike. Protocols, Not Platforms. **Knight First Amendment Institute**, *[S. l.]*, 2019. . Acesso em: 27 nov. 2024.

MCCOWN, Frank; NELSON, Michael L. What happens when facebook is gone? *Em*: PROCEEDINGS OF THE 9TH ACM/IEEE-CS JOINT CONFERENCE ON DIGITAL LIBRARIES 2009, Austin TX USA. **Anais** [...]. . *Em*: JCDL '09: JOINT CONFERENCE ON DIGITAL LIBRARIES. Austin TX USA: ACM, 2009. p. 251–254.

DOI: 10.1145/1555400.1555440. Disponível em: https://dl.acm.org/doi/10.1145/1555400.1555440. Acesso em: 2 dez. 2024.

MILOJICIC, Dejan S.; KALOGERAKI, Vana; LUKOSE, Rajan; NAGARAJA, Kiran; PRUYNE, Jim; RICHARD, Bruno; ROLLINS, Sami; XU, Zhichen. Peer-to-Peer Computing. *[S. l.]*, 2003.

NEWBOLD, Bryan. **Notes on Running a Full-Network atproto Relay (July 2024)**. **WhtieWind**, 2024. Disponível em: https://whtwnd.com/bnewbold.net/3kwzl7tye6u2y. Acesso em: 2 nov. 2024.

PACHECO, Denis. Navegar é preciso! Regular (as redes) também. **Jornal da USP**, *[S. l.]*, 2023. Disponível em: https://jornal.usp.br/atualidades/especial-desconstruindo-a-desinformacao-navegar-e-preciso-regular-as-redes-tambem/. Acesso em: 2 dez. 2024.

RAMAN, Aravindh; JOGLEKAR, Sagar; CRISTOFARO, Emiliano De; SASTRY, Nishanth; TYSON, Gareth. Challenges in the Decentralised Web: The Mastodon Case. *Em*: PROCEEDINGS OF THE INTERNET MEASUREMENT CONFERENCE 2019, Amsterdam Netherlands. **Anais** [...]. . *Em*: IMC '19: ACM INTERNET MEASUREMENT CONFERENCE. Amsterdam Netherlands: ACM, 2019. p. 217–229. DOI: 10.1145/3355369.3355572. Disponível em: https://dl.acm.org/doi/10.1145/3355369.3355572. Acesso em: 11 nov. 2024.

SECURE SCUTTLEBUTT CONSORTIUM. **Scuttlebutt Protocol Guide - How Scuttlebutt peers find and talk to each other**. 2023. Disponível em: https://ssbc.github.io/scuttlebutt-protocol-guide/.

STEEN, Maarten Van; TANENBAUM, Andrew S. **Distributed Systems**. Fourth edition, version 4.01 (January 2023) ed. Erscheinungsort nicht ermittelbar: Maarten van Steen, 2023.

TARR, Dominic; LAVOIE, Erick; MEYER, Aljoscha; TSCHUDIN, Christian. Secure Scuttlebutt: An Identity-Centric Protocol for Subjective and Decentralized Applications. *Em*: PROCEEDINGS OF THE 6TH ACM CONFERENCE ON INFORMATION-CENTRIC NETWORKING 2019, Macao China. **Anais** [...]. . *Em*: ICN '19: 6TH ACM CONFERENCE ON INFORMATION-CENTRIC NETWORKING.

Macao China: ACM, 2019. p. 1–11. DOI: 10.1145/3357150.3357396. Disponível em: https://dl.acm.org/doi/10.1145/3357150.3357396. Acesso em: 2 nov. 2024.

WIEGERS, Karl Eugene; BEATTY, Joy. **Software requirements**. Third edition ed. Redmond, Washington: Microsoft Press, s division of Microsoft Corporation, 2013.

APÊNDICE A - Definições de cada subcaracterística de qualidade ISO/IEC 25010 (2011)

Tradução realizada pelo autor com base nas definições do documento oficial.

Adequação Funcional

Grau em que um produto ou sistema fornece funções que atendem às necessidades declaradas e implícitas quando usado sob condições específicas.

- **Completude funcional:** Grau em que o conjunto de funções cobre todas as tarefas e objetivos especificados do usuário.
- **Correção funcional:** Grau em que um produto ou sistema fornece resultados corretos com o nível necessário de precisão.
- **Adequação funcional:** Grau em que as funções facilitam a realização de tarefas e objetivos específicos.

Eficiência de Desempenho

Desempenho em relação à quantidade de recursos utilizados sob condições estabelecidas.

- **Comportamento temporal:** Grau em que os tempos de resposta e processamento e as taxas de transferência de um produto atendem aos requisitos.
- **Utilização de recursos:** Grau em que as quantidades e tipos de recursos usados por um produto atendem aos requisitos.
- **Capacidade:** Grau em que os limites máximos de um parâmetro do produto ou sistema atendem aos requisitos.

Compatibilidade

Grau em que um produto, sistema ou componente pode trocar informações com outros produtos, sistemas ou componentes e/ou realizar suas funções necessárias enquanto compartilha o mesmo ambiente de hardware ou software.

- **Coexistência**: Grau em que um produto pode realizar suas funções de forma eficiente enquanto compartilha ambiente e recursos com outros produtos.
- **Interoperabilidade**: Grau em que dois ou mais sistemas, produtos ou componentes podem trocar informações e usar as informações trocadas.

Usabilidade

Grau em que um produto ou sistema pode ser usado por usuários específicos para atingir objetivos específicos com eficácia, eficiência e satisfação em um contexto específico de uso.

- **Reconhecimento de adequação:** Grau em que os usuários podem reconhecer se um produto ou sistema é apropriado para suas necessidades.
- **Apreensibilidade**: Grau em que um produto ou sistema pode ser usado por usuários específicos para atingir objetivos específicos de aprendizado.
- **Operabilidade**: Grau em que um produto ou sistema possui atributos que o tornam fácil de operar e controlar.
- **Proteção contra erros:** Grau em que o sistema protege os usuários contra erros.
- **Estética da interface:** Grau em que uma interface permite interação agradável e satisfatória para o usuário.
- **Acessibilidade**: Grau em que um produto ou sistema pode ser usado por pessoas com a mais ampla gama de características e capacidades.

Confiabilidade

Grau em que um sistema, produto ou componente executa funções especificadas sob condições especificadas por um período especificado.

- **Maturidade**: Grau em que um sistema atende às necessidades de confiabilidade em operação normal.
- **Disponibilidade**: Grau em que um sistema, produto ou componente está operacional e acessível quando necessário.
- **Tolerância a falhas**: Grau em que um sistema, produto ou componente opera conforme pretendido apesar da presença de falhas.

- **Recuperabilidade**: Grau em que um produto ou sistema pode recuperar dados afetados e restabelecer o estado desejado do sistema em caso de interrupção ou falha.

Segurança

Grau em que um produto ou sistema protege informações e dados para que pessoas ou outros produtos ou sistemas tenham o grau de acesso aos dados apropriado aos seus tipos e níveis de autorização.

- **Confidencialidade**: Grau em que um produto ou sistema garante que os dados sejam acessíveis apenas por aqueles autorizados.
- **Integridade**: Grau em que um sistema, produto ou componente impede acesso ou modificação não autorizada de programas ou dados.
- **Não repúdio:** Grau em que ações ou eventos podem ser comprovados como tendo ocorrido.
- **Responsabilização**: Grau em que as ações de uma entidade podem ser rastreadas de forma única até a entidade.
- **Autenticidade**: Grau em que a identidade de um sujeito ou recurso pode ser comprovada como sendo a alegada.

Manutenibilidade

Grau de eficácia e eficiência com que um produto ou sistema pode ser modificado pelos mantenedores pretendidos.

- **Modularidade**: Grau em que um sistema é composto por componentes discretos de forma que a mudança em um componente tenha impacto mínimo em outros.
- **Reusabilidade**: Grau em que um ativo pode ser usado em mais de um sistema ou na construção de outros ativos.
- **Analisabilidade**: Grau de eficácia e eficiência com que é possível avaliar o impacto de uma mudança pretendida.
- **Modificabilidade**: Grau em que um produto ou sistema pode ser modificado sem degradar a qualidade existente.

- **Testabilidade**: Grau de eficácia e eficiência com que critérios de teste podem ser estabelecidos e os testes executados.

Portabilidade

Grau de eficácia e eficiência com que um sistema, produto ou componente pode ser transferido de um hardware, software ou outro ambiente operacional ou de uso para outro.

- **Adaptabilidade**: Grau em que um produto ou sistema pode ser adaptado para diferentes ambientes de hardware, software ou outros ambientes operacionais.
- **Instalabilidade**: Grau de eficácia e eficiência com que um produto ou sistema pode ser instalado/desinstalado com sucesso.
- **Substituibilidade**: Grau em que um produto pode ser substituído por outro produto de software especificado para o mesmo propósito no mesmo ambiente.