

**UNIVERSIDADE FEDERAL DE ALAGOAS - UFAL
PROGRAMA DE PÓS GRADUAÇÃO EM INFORMÁTICA - PPGI
MESTRADO EM INFORMÁTICA
CAMPUS A.C SIMÕES**

RENDRIKSON DE OLIVEIRA SOARES

**UMA ARQUITETURA COMPUTACIONAL BASEADA EM TECNOLOGIA
BLOCKCHAIN PARA HABILITAR A TRANSPARÊNCIA DOS SERVIÇOS DE
AUDIÊNCIA E PAGAMENTOS EM PLATAFORMAS DE STREAMING**

**MACEIÓ
2024**

Rendrikson de Oliveira Soares

Uma Arquitetura Computacional Baseada em Tecnologia Blockchain para Habilitar a
Transparência dos Serviços de Audiência e Pagamentos em Plataformas de Streaming

Dissertação apresentada como requisito parcial
para obtenção do grau de Mestre em Informática
no Programa de Pós Graduação em Informática -
PPGI da Universidade Federal de Alagoas - UFAL,
Campus A.C Simões.

Orientador: Prof. Dr. André Magno Costa de
Araújo

Maceió
2024

Catálogo na Fonte
Universidade Federal de Alagoas
Biblioteca Central
Divisão de Tratamento Técnico

Bibliotecário: Marcelino de Carvalho Freitas Neto – CRB-4 - 1767

S676a Soares, Rendrikson de Oliveira.
 Uma arquitetura computacional baseada em tecnologia *blockchain*
 para habilitar a transparência dos serviços de audiência e pagamentos
 em plataformas de *streaming* / Rendrikson de Oliveira Soares. – 2024.
 101 f. : il.

Orientador: André Magno Costa de Araújo.
Dissertação (mestrado em informática) - Universidade Federal de
Alagoas. Instituto de Computação. Maceió, 2024.

Bibliografia: f. 98-101.

1. Arquitetura de *software*. 2. Blockchains (Base de dados). 3. Smart
contract. 4. Tecnologia *streaming* (Telecomunicação). 5. Software -
Desenvolvimento. 6. Armazenamento de dados. I. Título.

CDU: 004.4



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE ALAGOAS
INSTITUTO DE COMPUTAÇÃO
Av. Lourival Melo Mota, S/N, Tabuleiro do Martins, Maceió - AL, 57.072-970
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO (PROPEP)
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

Folha de Aprovação

RENDRIKSON DE OLIVEIRA SOARES

UMA ARQUITETURA COMPUTACIONAL BASEADA EM TECNOLOGIA
BLOCKCHAIN PARA HABILITAR A TRANSPARÊNCIA DOS SERVIÇOS DE
AUDIÊNCIA E PAGAMENTOS EM PLATAFORMAS DE STREAMING

A COMPUTATIONAL ARCHITECTURE BASED ON BLOCKCHAIN TECHNOLOGY
TO ENABLE TRANSPARENCY OF AUDIENCE SERVICES AND PAYMENTS ON
STREAMING PLATFORMS

Dissertação submetida ao corpo docente do
Programa de Pós-Graduação em Informática
da Universidade Federal de Alagoas e
aprovada em 26 de fevereiro de 2024.

Banca Examinadora:

Documento assinado digitalmente
 **ANDRE MAGNO COSTA DE ARAUJO**
Data: 26/02/2024 10:29:25-0300
Verifique em <https://validar.iti.gov.br>

Prof. Dr. ANDRÉ MAGNO COSTA DE ARAUJO
UFAL – Instituto de Computação
Orientador

Documento assinado digitalmente
 **FABIO JOSE COUTINHO DA SILVA**
Data: 27/02/2024 12:32:35-0300
Verifique em <https://validar.iti.gov.br>

Prof. Dr. FABIO JOSE COUTINHO DA SILVA
UFAL – Instituto de Computação
Examinador Interno

Documento assinado digitalmente
 **PAULO CAETANO DA SILVA**
Data: 26/02/2024 15:53:39-0300
Verifique em <https://validar.iti.gov.br>

Prof. Dr. PAULO CAETANO DA SILVA
UNIFACS – Universidade Salvador
Examinador Externo

AGRADECIMENTOS

Primeiramente eu gostaria de agradecer a Deus, por me proporcionar todas as condições necessárias para realização dessa tão sonhada dissertação.

À minha esposa, Emanuelle, expresso profundo agradecimento por seu incansável apoio e cuidado ao longo de toda a trajetória. Sem a sua presença constante, esta jornada dificilmente teria alcançado o seu objetivo final. Sua contribuição foi inestimável e fundamental para os resultados deste percurso.

Aos meus pais, José Renato Soares e Rosemary de Oliveira Soares, e ao meu irmão Rodrigo de Oliveira Soares, que mesmo longe sempre tiveram torcendo pelo meu sucesso e estiveram ao meu lado prestando total apoio nessa jornada.

Ao professor André Magno, minha eterna gratidão por ter me conduzido ao longo de toda a jornada com uma orientação humana e amigável. Expresso meus sinceros agradecimentos pelos valiosos conselhos que carregarei para toda a vida, pelas oportunidades que se apresentaram, por acreditar em mim e pela notável paciência demonstrada. Todos os ensinamentos recebidos serão levados comigo, e farei questão de perpetuá-los em minha trajetória pessoal e profissional.

Por fim, agradeço a todos que participaram, direta ou indiretamente, do desenvolvimento deste trabalho de pesquisa, enriquecendo o meu processo de aprendizado.

O sucesso é a soma de pequenos esforços
repetidos dia após dia.

Robert Collier

RESUMO

A tecnologia *Blockchain* possibilita o registro seguro e o compartilhamento de informações sem uma autoridade central, usando criptografia em uma rede distribuída de computadores. Amplamente utilizada na indústria de software, oferece recursos de autenticidade e segurança em transações online. Plataformas de *streaming*, como *YouTube*, *Spotify*, dentre outras, são exemplos de sistemas que lidam com a transmissão de dados de diferentes tipos, como vídeos, músicas e *podcasts*, e que precisam oferecer recursos de transparência, rastreabilidade e segurança na manipulação desses dados em sistemas de software. No campo de pesquisa da indústria do entretenimento digital, identificou-se no estado da arte uma lacuna de pesquisa no que diz respeito à aplicação da tecnologia *Blockchain* para habilitar a transparência nos serviços de audiência e pagamentos em plataformas de *streaming*. Este trabalho propõe a inserção de uma camada *Blockchain* à infraestrutura operacional existente em plataformas de streaming disponíveis no mercado. Essa camada é encarregada de gerenciar informações individuais dos conteúdos, servindo como meio de verificação e auditoria das transações associadas a cada mídia digital. Além disso, ela permite a transferência de recursos monetários por meio da rede *Blockchain*. Para atingir o objetivo proposto, elaborou-se uma arquitetura de software contendo os componentes, interfaces, relacionamentos e restrições dos requisitos funcionais que representam esse domínio de aplicação. Nesse processo, adotou-se o modelo de especificação de arquitetura denominado *C4 model*, escolhido devido à sua característica de documentação compreensível tanto para um público técnico quanto não técnico. Após a especificação da arquitetura, foram implementados os componentes fundamentais para os testes e operação da solução proposta nesta pesquisa. Dessa maneira, desenvolveu-se um *middleware* encarregado de capturar e gerenciar os mecanismos de audiência de cada conteúdo de maneira individual. Isso ocorre por meio de um serviço de geração automática de contratos inteligentes que é executado quando os conteúdos são disponibilizados e consumidos nas plataformas de streaming. Posteriormente, desenvolveu-se uma aplicação web que simula as operações de uma plataforma de *streaming* para integrar ao *middleware* desenvolvido, permitindo a realização de testes, tanto financeiros, quanto funcionais. Os testes foram conduzidos em três redes *Blockchain* distintas, revelando a viabilidade técnica da solução proposta. Isso foi alcançado por meio da interceptação e do gerenciamento automático das informações de audiência e pagamentos utilizando contratos inteligentes. Além disso, a viabilidade financeira da implementação foi analisada, resultando em um custo médio de US\$ 0,000518 em uma das redes *Blockchain*. Adicionalmente, realizou-se uma pesquisa envolvendo criadores de conteúdo com canais monetizados e desenvolvedores de software especializados em *Blockchain*. O objetivo da consulta a estes profissionais foi avaliar e obter *feedback* sobre a nova arquitetura de software e a viabilidade da solução proposta. Dessa forma, os resultados englobaram *feedbacks* que validaram a implementação da solução computacional, confirmando também a operacionalização da plataforma por parte dos criadores de conteúdo de plataformas de streaming.

Palavras-chave: Arquitetura de Software, *Blockchain*, Smart Contract, Plataforma de Streaming, Desenvolvimento de Software, Armazenamento de Dados.

ABSTRACT

Blockchain technology enables secure recording and sharing of information without a central authority, using encryption across a distributed network of computers. Widely used in the software industry, it offers authenticity and security features in online transactions. Streaming platforms, such as YouTube and Spotify, among others, are examples of systems that deal with the transmission of data of different types, such as videos, music, and podcasts, and that need to offer transparency, traceability, and security features when handling this data in software systems. In the digital entertainment industry, a research gap was identified in the state of the art concerning the application of Blockchain technology to enable transparency in audience services and payments on streaming. This work proposes inserting a Blockchain layer into the existing operational infrastructure of streaming platforms available on the market. This layer manages individual content information and verifies and audits transactions associated with each digital media. Furthermore, it allows the transfer of monetary resources through the Blockchain network. A software architecture was created to achieve the proposed objective containing the components, interfaces, relationships, and constraints of the functional requirements representing this application domain. In this process, an architecture specification model, the C4 model, was adopted due to its understandable documentation characteristic for both technical and non-technical audiences. After specifying the architecture, the fundamental components for testing and operating the solution proposed in this research were implemented. In this way, middleware was developed to capture and manage the audience mechanisms for each piece of content individually. This occurs through an automatic smart contract generation service executed when content is available and consumed on streaming platforms. Subsequently, a web application was developed that simulates the operations of a streaming platform to integrate with the middleware developed, allowing tests to be carried out, both financial and functional. The tests were conducted on three different Blockchain networks, revealing the technical feasibility of the proposed solution. This was achieved through the interception and automatic management of audience and payment information using smart contracts. Furthermore, the financial viability of the implementation was analyzed, resulting in an average cost of US\$0.000518 on one of the Blockchain networks. Additionally, the research involved content creators with monetized channels and software developers specializing in Blockchain. The objective of consulting these professionals was to evaluate and obtain feedback on the new software architecture and the feasibility of the proposed solution. Thus, the results included feedback that validated the implementation of the computational solution, confirming the platform's operationalization by content creators on streaming platforms.

Keywords: Software Architecture, Blockchain, Smart Contract, Streaming Platform, Software Development, Data Storage.

LISTA DE FIGURAS

Figura 1 – Lista encadeada de Blocos na rede <i>Blockchain</i>	22
Figura 2 – Composição de um bloco <i>Blockchain</i>	23
Figura 3 – Transações de registros de blocos <i>Blockchain</i>	24
Figura 4 – Estrutura Básica de um Smart Contract	25
Figura 5 – Transações com Contratos Tradicionais	26
Figura 6 – Transações com Smart Contracts	26
Figura 7 – Arquitetura de software como uma ponte	29
Figura 8 – Arquitetura orientada a Serviço	30
Figura 9 – Visão geral das quatro camadas do C4 model	33
Figura 10 – Metodologia de pesquisa	36
Figura 11 – String de Busca	37
Figura 12 – Visão geral da arquitetura híbrida com back-end	42
Figura 13 – Visão geral da arquitetura híbrida proposta	42
Figura 14 – Visão geral da arquitetura de plataformas interativas de transmissão ao vivo com redis	43
Figura 15 – Categorização dos trabalhos investigados nessa seção	47
Figura 16 – Principal fluxo de disponibilização de conteúdo da arquitetura do youtube .	54
Figura 17 – Arquitetura Proposta	54
Figura 18 – Fluxo de upload de mídia da plataforma de streaming	56
Figura 19 – Nível de contexto da arquitetura proposta: pessoas, software, modelo de dados e suas relações	60
Figura 20 – Nível de contêiner da arquitetura proposta e fluxo de dados entre os contêine- res e componentes.	62
Figura 21 – Nível de componente da arquitetura proposta e fluxo de dados entre os contêineres e componentes.	64
Figura 22 – Nível de componente do <i>Roteamento</i> e suas interações.	66
Figura 23 – Nível de componente dos <i>Controlers</i> e suas interações	67
Figura 24 – Nível de componente do <i>Contrato inteligente</i> do conteúdo armazenado na plataforma de streaming	69
Figura 25 – Principais Funções do Smart Contract	73
Figura 26 – Plataforma de Streaming baseada em Tecnologia <i>Blockchain</i>	76
Figura 27 – Informações das Requisições	80
Figura 28 – Informações de Hardware	80
Figura 29 – Clareza das regras de monetização	82
Figura 30 – Fornecimento de documentação sobre o cálculo da monetização	83
Figura 31 – Pagamentos diretamente ligados ao número específico de visualizações . . .	83
Figura 32 – Acessibilidade as informações de restrição de conteúdo da plataforma . . .	84

Figura 33 – Eficácia da arquitetura proposta para solucionar a falta de transparência entre criadores de conteúdo e a plataforma de streaming	85
Figura 34 – Viabilidade de implantação da arquitetura proposta	85
Figura 35 – Nuvem de Palavras das respostas sobre dificuldades e melhorias na Plataforma de Streaming	89

LISTA DE TABELAS

Tabela 1 – Critérios de Exclusão	38
Tabela 2 – Critérios de Inclusão	38
Tabela 3 – Formulário de Extração de Dados	39

LISTA DE QUADROS

Quadro 1 – Custo por tipo de transação	76
Quadro 2 – Custo por 10 Mil Transações em Diferentes Redes <i>Blockchain</i>	77

LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application Programming Interface</i>
DApps	<i>Decentralized Applications</i>
GHz	<i>Gigahertz</i>
IGU	Interface Gráfica de Usuário
IOT	<i>Internet of Things</i>
IPFS	<i>InterPlanetary File System</i>
JSON	<i>JavaScript Object Notation</i>
SGBD	Sistema de Gerenciamento de Banco de Dados
HTTP	<i>Hypertext Transfer Protocol</i>
HTTPS	<i>Hyper Text Transfer Protocol Secure</i>
NOSQL	<i>Not Only Structured Query Language</i>
PPP	<i>Pay-Per-Play</i>
RAM	<i>Random Access Memory</i>
REST	<i>Representational State Transfer</i>
RF	Requisito Funcional
RTSP	<i>Real-Time Streaming Protocol</i>
SOAP	<i>Simple Object Access Protocol</i>
SOA	<i>Service-Oriented Architecture</i>
SSD	<i>Solid-State Drive</i>
SQL	<i>Structured Query Language</i>
UDDI	<i>Universal Description, Discovery and Integration</i>
UML	<i>Unified Modeling Language</i>
WSDL	<i>Web Services Description Language</i>
XML	<i>eXtensible Markup Language</i>

SUMÁRIO

1	INTRODUÇÃO	14
1.1	Contextualização	14
1.2	Motivação	15
1.3	Objetivos	17
1.3.1	Objetivo Geral	17
1.3.2	Objetivos Específicos	18
1.4	Metodologia	18
1.5	Organização do Trabalho	19
2	REFERENCIAL TEÓRICO	21
2.1	Blockchain	21
2.2	Smart Contracts	24
2.3	Modelos Arquiteturais de Software	27
2.3.1	Arquitetura Orientada a Serviços	29
2.3.2	Arquitetura Orientada a Eventos	31
2.4	Requisitos Funcionais e Não Funcionais	32
2.5	O modelo C4 de documentação para Arquitetura de software	32
2.6	Considerações Finais	33
3	TRABALHOS CORRELATOS	35
3.1	Metodologia de revisão do estado da arte	35
3.1.1	Objetivo	35
3.1.1.1	Informações da pesquisa	36
3.1.2	Estratégia de Pesquisa	36
3.1.3	Dados do estudo	38
3.1.4	Coleta e avaliação dos dados	39
3.2	Análise do estado da arte	40
3.2.1	Arquiteturas de software	40
3.2.2	Plataformas de streaming com <i>Blockchain</i>	44
3.3	Discussão dos resultados	48
3.4	Considerações finais	50
4	ARQUITETURA DE SOFTWARE DA SOLUÇÃO PROPOSTA	51
4.1	Cenário Funcional	52
4.2	Requisitos de software	55

4.2.1	Requisitos Funcionais e Não Funcionais da Arquitetura StreamingChain . . .	55
4.3	Formalização da Arquitetura de Software com o padrão C4 Model	58
4.3.1	Níveis da Arquitetura	59
4.3.1.1	Nível de Contexto	59
4.3.1.2	Nível de Contêiner	61
4.3.1.3	Nível de <i>Componente</i>	63
4.3.1.4	Nível de <i>Código</i>	69
4.4	Considerações Finais	70
5	AVALIAÇÃO DA SOLUÇÃO PROPOSTA	71
5.1	Avaliação do serviço de geração de transações <i>Blockchain</i>	71
5.1.1	Infraestrutura	72
5.1.2	Primeira Etapa: Construção do Smart Contract e Middleware	73
5.1.3	Segunda Etapa: Construção do protótipo de plataforma de streaming	74
5.1.4	Avaliação em diferentes redes <i>Blockchain</i>	75
5.2	Avaliação de Monitoramento de Recursos da Aplicação	78
5.3	Avaliação quantitativa e qualitativa do modelo de negócio proposto	80
5.3.1	Pesquisa com criadores de conteúdo	81
5.3.1.1	Pesquisa quantitativa	82
5.3.1.2	Pesquisa qualitativa	85
5.3.2	Pesquisa com desenvolvedores <i>Blockchain</i>	90
5.4	Considerações Finais	92
6	CONCLUSÃO	94
6.1	Considerações Finais	94
6.2	Limitações	95
6.3	Trabalhos Futuros	96
	REFERÊNCIAS	98

1 INTRODUÇÃO

Este capítulo tem como objetivo introduzir o avanço científico acerca do tema de pesquisa abordado nesta dissertação, e está estruturado da seguinte forma. A Seção 1.1 contextualiza e discute o atual cenário de funcionamento e consumo de conteúdo em plataformas de streaming. A Seção 1.2 motiva o uso da tecnologia *Blockchain* em serviços de streaming, enquanto a Seção 1.3 apresenta os objetivos gerais e específicos desta pesquisa. A Seção 1.4 descreve a metodologia utilizada no desenvolvimento desta dissertação. Por fim, a Seção 1.5 aborda como os capítulos desta pesquisa estão organizados.

1.1 CONTEXTUALIZAÇÃO

O consumo de serviços multimídia em plataformas de streaming tem passado por grandes transformações ao longo dos últimos anos. Os meios tradicionais de consumo de entretenimento têm perdido mercado para as plataformas de streaming, e isso ocorre em razão da popularização dos smartphones e aplicativos móveis, da evolução das redes de dados móveis, do fortalecimento das redes de banda larga e do aperfeiçoamento das tecnologias de desenvolvimento de sistemas de software e armazenamento de dados [Lee et al. 2018, Wayne 2020].

As plataformas de streaming têm se consolidado na economia mundial como uma fonte poderosa de geração de receita. Prova disso são os números de audiência registrados nos últimos anos [Moore 2020, Nielsen 2022, Laura 2020]. Somente no terceiro trimestre de 2023, uma das maiores plataformas de streaming de música alcançou a importante marca de 210 milhões de assinantes e 515 milhões de usuários ativos mensais [Götting 2023]. Em 2023, uma das maiores plataformas de streaming de vídeo teve 2,70 bilhões de visitantes logados mensalmente e alcançou a expressiva marca de mais de 70 bilhões de visualizações diárias na modalidade de vídeos curtos. Esta mesma plataforma conseguiu alcançar uma receita US\$ 7,95 bilhões no terceiro trimestre de 2023 [GMI 2024]. Os números das plataformas de streaming voltadas para vídeos ao vivo também atestam o sucesso desse modelo de negócio. Em 2023, o total de horas de uma das maiores organizações de streaming de games atingiu a importante marca de 5,28 bilhões de horas assistidas no primeiro trimestre [Murko 2023].

A transformação digital provocada pelas plataformas de streaming, baseia-se em modelos de negócio no qual os stakeholders (e.g., criadores de conteúdo, influenciadores digitais, anunciantes, gravadoras e editoras) envolvidos neste ecossistema podem monetizar os seus serviços de diferentes formas [Bowman et al. 2020, Elango 2019]. Três estratégias de monetização são co-

mumente adotadas pelas plataformas de streaming para atrair os stakeholders neste ecossistema, são elas: O Ad Supported, estratégia que fornece conteúdo gratuito para os telespectadores e, os anunciantes pagam para exibir os seus produtos e/ou serviços em forma de anúncio para os mesmos. O Freemium, estratégia que dispõe de uma versão gratuita, no qual o usuário possui um catálogo limitado e necessita ver anúncios; e a versão paga, que habilita o acesso a todo o conteúdo desejado sem a necessidade de visualização de anúncios. Por fim, a estratégia Premium, requer dos usuários o pagamento de um plano para acesso ao conteúdo desejado sem interferência de anúncios.

Em virtude da heterogeneidade dos tipos de dados encontrados nas plataformas de streaming, e em razão das diferentes necessidades de processamento dos dados, a academia e a indústria de software têm concentrado seus esforços na busca por soluções computacionais que ofereçam recursos de transparência, rastreabilidade e segurança na manipulação de dados em sistemas de software [Sathish et al. 2019, Chavan et al. 2019].

Nesse sentido, a tecnologia *Blockchain* vem sendo estudada e aplicada na indústria de software como uma opção para garantir a autenticidade das transações de dados trafegadas na internet [Barman et al. 2020]. A tecnologia *Blockchain* consiste em uma rede descentralizada e criptografada que certifica e guarda todas as informações transacionais entre as partes envolvidas de forma imutável [Nakamoto 2008]. Assim, a tecnologia *Blockchain* surge como uma opção promissora para lidar com a complexidade e as demandas de segurança e transparência de dados das plataformas de streaming, oferecendo uma solução descentralizada e criptografada para a manipulação de informações transacionais na internet.

1.2 MOTIVAÇÃO

O uso da tecnologia *Blockchain* tem sido amplamente explorado em diversas áreas do conhecimento, como saúde, segurança pública, internet das coisas e serviços de streaming [Bowman et al. 2020, Xu et al. 2022, Barman et al. 2020, Majeed et al. 2021]. Muitas pesquisas têm utilizado os fundamentos e as características da tecnologia para propor soluções para o ecossistema das organizações de entretenimento digital [Bowman et al. 2020, Liu et al. 2018]. As plataformas de streaming revolucionaram os modelos de negócios da indústria do entretenimento, melhorando a forma como a sociedade acessa e consome serviços multimídia. No entanto, existem desafios e oportunidades de pesquisa que precisam ser investigados [Torbensen e Ciriello 2019]. Entre eles, destaca-se a falta de transparência e rastreabilidade na forma como os serviços de audiência são implementados por cada plataforma, bem como a

maneira como as métricas de monetização são calculadas para pagamento dos stakeholders, o que gera insegurança e insatisfação entre as partes envolvidas [Soares et al. 2023].

A transparência e a rastreabilidade são requisitos importantes para garantir a confiabilidade dos sistemas de informação e fomentar a equidade nas relações de negócio. Na indústria de entretenimento digital, esses requisitos são fundamentais para assegurar a clareza na distribuição dos lucros e para prevenir fraudes. Nesse sentido, a tecnologia *Blockchain* tem sido amplamente estudada como uma solução viável para promover a transparência e a rastreabilidade no ecossistema das organizações de entretenimento digital [Chavan et al. 2019]. A tecnologia *Blockchain* é um registro distribuído e imutável que pode ser utilizado para garantir a transparência e a integridade dos dados em diferentes setores da economia.

Diversas pesquisas têm proposto o uso da tecnologia *Blockchain* para solucionar os problemas de transparência e rastreabilidade em diferentes cenários de uso [Chavan et al. 2019, Xu et al. 2022]. A aplicação da tecnologia *Blockchain* permite a criação de registros distribuídos e imutáveis que podem ser auditados por todas as partes envolvidas, o que aumenta a transparência e a confiança no sistema. Além disso, a tecnologia *Blockchain* pode ser utilizada para implementar smart contracts, que são contratos autônomos que executam as cláusulas automaticamente quando as condições predefinidas são atendidas.

Considerando a problemática abordada anteriormente, percebe-se a necessidade de desenvolver uma solução capaz de mitigar os desafios identificados no ecossistema das organizações de entretenimento digital, de maneira a causar o mínimo impacto possível no modelo de negócio atual e, ao mesmo tempo, utilizar uma implementação já validada e disponível no mercado. Essa solução deve atender às demandas do setor de streaming, promovendo maior transparência e rastreabilidade nas métricas de audiência e monetização, a fim de gerar maior confiança e satisfação entre as partes interessadas. Nesse sentido, é fundamental que a proposta seja fundamentada em uma abordagem tecnológica capaz de integrar os princípios da *Blockchain* com as especificidades do setor de entretenimento, promovendo maior eficiência e segurança aos processos envolvidos. Com isso, busca-se contribuir para o avanço do conhecimento científico nessa área, por meio da proposição de uma solução que possa ser aplicada em diferentes contextos do setor de streaming.

Para tanto, é necessário realizar uma análise do estado da arte, com o objetivo de identificar os principais desafios e oportunidades de pesquisa na área de aplicação da tecnologia *Blockchain* no setor de entretenimento digital. Adicionalmente, é fundamental proceder com o desenvolvimento de uma aplicação de streaming empregando a tecnologia *Blockchain*, conduzir testes e simulações para avaliar a eficácia e eficiência da solução proposta, bem como avaliar os

possíveis impactos no modelo de negócio das empresas envolvidas. Neste contexto, destaca-se que a documentação da arquitetura será elaborada conforme o modelo C4, proporcionando uma estrutura clara e abrangente para representação e comunicação dos elementos arquiteturais. A utilização deste modelo é crucial para garantir uma compreensão holística da arquitetura proposta, facilitando a comunicação entre os envolvidos e proporcionando uma base sólida para a tomada de decisões. Por fim, a partir dos resultados obtidos, será possível propor recomendações e diretrizes que possam orientar a adoção do *Blockchain* no setor de entretenimento digital, promovendo uma maior transparência, segurança e eficiência aos processos envolvidos.

1.3 OBJETIVOS

1.3.1 Objetivo Geral

Esta dissertação tem como objetivo geral especificar uma arquitetura baseada em Tecnologia *Blockchain* para habilitar a transparência no gerenciamento de audiência e pagamentos em plataformas de Streaming. A arquitetura de software foi especificada utilizando o padrão C4 model, que organiza em quatro camadas os componentes da solução computacional proposta (Contexto, Contêiner, Componente e Código) proporcionando uma visão holística e hierárquica dos artefatos de software.

A solução descrita nesta pesquisa visa avançar no estado da arte e da prática com o desenvolvimento de uma solução computacional que torna clara e auditável as regras de negócios entre as plataformas de streaming e os stakeholders envolvidos neste domínio de problema. Para avaliar a pesquisa descrita nesta dissertação, as seguintes atividades foram realizadas. Desenvolveu-se um *middleware* encarregado de capturar e gerenciar os mecanismos de audiência de cada conteúdo de maneira individual. Posteriormente, implementou-se uma aplicação web que simula as operações de uma plataforma de *streaming* para se integrar ao *middleware* desenvolvido, permitindo a realização de testes, tanto financeiro, quanto dos requisitos funcionais. Os testes foram conduzidos em três redes *Blockchain* distintas, revelando a viabilidade técnica da solução proposta. Por fim, realizou-se uma pesquisa envolvendo criadores de conteúdo com canais monetizados e desenvolvedores de software especializados em *Blockchain*. O objetivo da consulta a estes profissionais foi avaliar e obter *feedback* sobre a nova arquitetura de software e a viabilidade da solução proposta.

1.3.2 Objetivos Específicos

Os objetivos específicos listados abaixo representam as principais ações de trabalho desenvolvidas, são elas:

- Realizar a análise do estado da arte e da prática;
- Projetar a arquitetura computacional baseada na tecnologia *Blockchain*;
- Formalizar a arquitetura de software no padrão C4 model;
- Desenvolver o middleware responsável por integrar os números de audiência das plataformas de streaming com a solução computacional proposta;
- Implementar funcionalidade de monitoramento de hardware e requisições em redes *Blockchain*;
- Avaliar a eficiência da solução computacional proposta por meio de experimentos computacionais automatizados;
- Realizar aplicação de questionários sobre a plataforma de streaming e a arquitetura proposta com criadores de conteúdo;
- Realizar aplicação de questionários sobre a arquitetura proposta com desenvolvedores de software;
- Escrever relatórios técnicos e artigos para publicação em conferências e periódicos com qualis em ciência da computação;
- Realizar o registro de software de todos os artefatos computacionais presentes nesta pesquisa.

1.4 METODOLOGIA

A metodologia utilizada para elaboração desta pesquisa consiste na revisão da literatura e na pesquisa exploratória. A revisão da literatura será realizada para identificar, no estado da arte e da prática, pesquisas que abordam o uso de tecnologia *Blockchain* no ecossistema da indústria de entretenimento. Para isso, as seguintes atividades serão realizadas:

- Definição da String de Busca;

- Execução da string de busca nos principais repositórios da ciência da computação;
- Elaboração dos critérios de aceitação e exclusão dos artigos encontrados pela string de busca;
- Análise das pesquisas incluídas pelo critério de aceitação
- Identificação das oportunidades e desafios descritos no estado da arte

A aplicação da pesquisa exploratória permitirá a identificação, nos modelos computacionais fornecidos pela tecnologia Blockchain, dos componentes essenciais para a elaboração da arquitetura de software e do middleware responsável pela integração da solução proposta com as plataformas de streaming. Adicionalmente, a condução de pesquisas quantitativas e qualitativas desempenhará um papel crucial na coleta de dados e análise dos resultados.

A coleta de dados será conduzida mediante o uso de formulários eletrônicos, incorporando perguntas fechadas utilizando a escala de Likert, além de perguntas abertas[Nemoto e Beglar 2014]. Essa abordagem possibilitará uma análise que contempla tanto métricas quantitativas quanto percepções qualitativas dos participantes. Para a análise das respostas qualitativas, será empregada a metodologia *Grounded Theory*, que inclui a codificação aberta, axial e seletiva, a fim de interpretar de maneira abrangente as informações coletadas[Fernandes e Maia 2001].

Adicionalmente, serão realizados testes computacionais automatizados para avaliar o custo computacional e financeiro do uso da tecnologia Blockchain nos serviços de streaming. Essa abordagem sistemática oferecerá uma base objetiva para a análise dos impactos da implementação da tecnologia Blockchain nesse contexto.

1.5 ORGANIZAÇÃO DO TRABALHO

A estrutura deste trabalho é composta por diversas seções desenvolvidas com o intuito de fornecer uma compreensão clara e organizada do estudo realizado. Este trabalho está organizado da seguinte forma:

- O Capítulo 2 apresenta a referencial teórico, que inclui os conceitos e definições relevantes para a compreensão do estudo realizado;
- O Capítulo 3 apresenta uma revisão dos trabalhos correlatos encontrados no estado da arte, que abordam a utilização da tecnologia *Blockchain* nas plataformas de entretenimento;

- No Capítulo 4 é descrita uma abordagem de arquitetura de software cujo objetivo melhorar a transparência e segurança das informações de audiência em plataformas de streaming.
- O Capítulo 5 mostra a avaliação deste trabalho, em que um conjunto de procedimentos de validação foram realizados para aferir a eficácia da proposta de pesquisa, junto com a aplicação de formulários de pesquisa com criadores de conteúdo e desenvolvedores de software.
- O Capítulo 6 é dedicado a apresentar a conclusão deste trabalho, com as contribuições, limitações e trabalhos futuros identificados nesta pesquisa.

2 REFERENCIAL TEÓRICO

Este capítulo descreve os conceitos básicos das ferramentas e tecnologias utilizadas no desenvolvimento deste trabalho. Na seção 2.1 é apresentado o conceito da tecnologia *Blockchain*. Em seguida a seção 2.2 apresenta os *smart contracts*. Na seção 2.3 é apresentado o conceito de arquitetura de software e os seus principais modelos. Na seção 2.4 são apresentados os conceitos de requisitos funcionais e não funcionais. A Seção 2.5 apresenta o conceito do modelo C4 de documentação para arquiteturas de software. Por fim, a seção 2.6 apresenta as considerações finais deste capítulo.

2.1 BLOCKCHAIN

A tecnologia *Blockchain* teve sua ascensão em 2008 com o Satoshi Nakamoto após a publicação do *white paper* sobre o desenvolvimento da moeda digital Bitcoin, responsável por difundir os conceitos básicos para o funcionamento da tecnologia *Blockchain* [Nakamoto 2008]. A *Blockchain* é uma tecnologia de registro distribuído que permite armazenar e transmitir informações de forma segura, confiável e transparente.

A arquitetura de uma rede *Blockchain* é composta de três componentes principais:

- **Bloco:** O bloco é a unidade básica da tecnologia *Blockchain*. Eles são usados para armazenar dados como transações, informações de usuário, tempo e data. Os blocos são ligados entre si usando criptografia para garantir que os dados sejam resistentes à manipulação;
- **Nós:** Um nó é um computador que está conectado à rede da cadeia de blocos¹. Os nós desempenham um papel crucial na operação da cadeia de bloco porque são responsáveis pela validação e registro das transações;
- **Rede:** A rede é formada por todos os nós que estão conectados à cadeia de blocos. É através da rede que as transações são transmitidas e validadas.

O armazenamento em uma rede *Blockchain* é baseado em uma rede de computadores. As transações são armazenadas em blocos criptografados, formando uma cadeia de blocos. Cada bloco possui um registro de múltiplas transações; após o registro dos dados em um bloco, este não pode ser alterado, pois qualquer modificação resultaria em uma mudança no *hash* desse bloco, afetando também todas as cópias da cadeia distribuída, exigindo o consenso da maioria dos

¹ Uma cadeia de blocos é uma tecnologia descentralizada que consiste em uma sequência contínua de blocos, onde cada bloco contém um registro de transações e informações

participantes para serem validadas. A lista encadeada é composta no início por um bloco gênese e os seguintes blocos possuem uma referência ao anterior, como é exposto na Figura 1. Somente um bloco pode ser adicionado por vez a essa corrente, sendo cada bloco matematicamente verificado através da utilização de criptografia para comprovar que os seguintes blocos seguem a sequência dos anteriores. Esse é o processo comumente chamado de “mineração”[Nakamoto 2008].

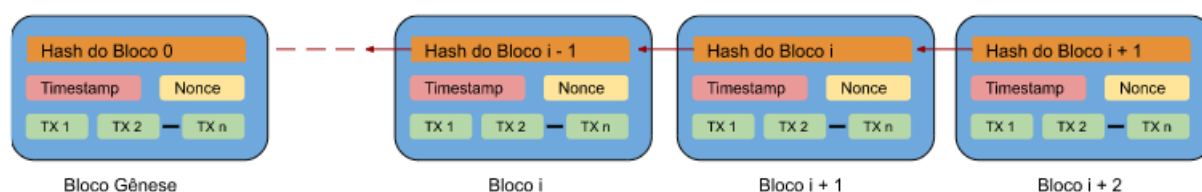


Figura 1 – Lista encadeada de Blocos na rede *Blockchain*

Fonte: [Zheng et al. 2018]

Um bloco *Blockchain* possui em sua composição o cabeçalho do bloco e o corpo do bloco. O cabeçalho do bloco possui a versão do bloco, responsável por indicar qual o conjunto de regras de validação é adotado por cada bloco; o *hash* do bloco anterior, é o campo responsável por salvar a referência do bloco anterior para manter a conexão dos blocos; a *hash* do bloco refere-se ao identificador único baseado no conteúdo do bloco e nas regras estabelecidas na rede; o timestamp, são registros de data e hora que são incorporados aos blocos de dados no momento de sua transação na rede *Blockchain*. Estes registros são usados para ajudar a verificar a autenticidade e a integridade dos dados contidos nos blocos, e impedir a reutilização de dados antigos; o nBits, é responsável por armazenar a *hash* do bloco com tamanho compacto; e o Nonce é um valor aleatório de tamanho máximo de 4 bytes, usado na prova de consenso em conjunto com a entrada do usuário para criar uma *hash* única para cada transação. O corpo do bloco possui um contador de transações e os dados armazenados no bloco. O tamanho máximo de transações de um bloco depende do conteúdo armazenado, do tamanho da transação e do tamanho do bloco [Zheng et al. 2018]. A Figura 2 ilustra a composição de um bloco.

As transações realizadas com a tecnologia *Blockchain* são baseadas em uma cadeia de assinaturas digitais. A transação será realizada por assinaturas digitais de um *hash* da transação anterior, um *hash* é um código que transforma as informações presente em um bloco em uma sequência numérica hexadecimal de tamanho fixo. As transações *Blockchain* são separadas em blocos, cada bloco possui uma chave pública do usuário e um *hash* baseado no bloco anterior, cada usuário possui uma chave privada que será responsável por confirmar sua autenticidade.

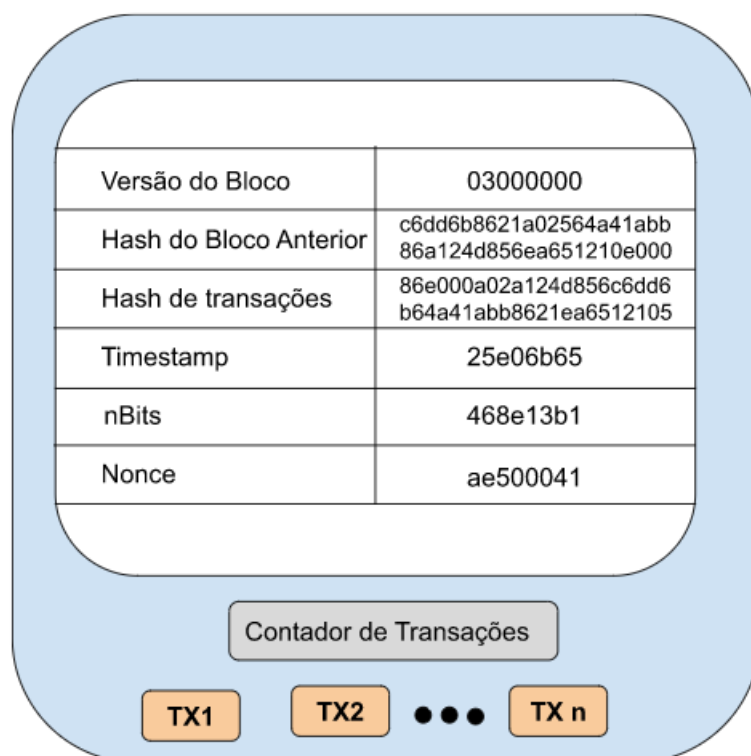


Figura 2 – Composição de um bloco *Blockchain*

Fonte: [Zheng et al. 2018]

As transações somente são realizadas após a utilização de três assinaturas digitais, a primeira contém o *hash* da transação anterior, a segunda contém a chave pública do usuário que vai receber a transação e a última assinatura digital contém a chave privada que libera o conteúdo presente na transação para o próximo usuário. Devido a esse mecanismo, ao tentar fraudar um dos blocos que já está presente na rede *Blockchain*, será gerado um *hash* incompatível com os anteriores, impossibilitando a modificação do seu conteúdo. A Figura 3 mostra a transação de registros de blocos *Blockchain*.

Para entender como funciona uma cadeia de blocos, é útil imaginar um grupo de pessoas em pé em círculo, cada uma segurando uma cópia de um livro-razão². Quando uma nova transação é feita, ela é anunciada ao grupo e adicionada ao livro-razão. Cada pessoa do grupo então verifica a transação para ter certeza de que é válida, e uma vez verificada, ela é adicionada ao livro razão e uma cópia é passada para cada pessoa do grupo. Este processo é repetido até que cada membro do grupo tenha uma cópia do livro razão atualizado. Como o livro razão é distribuído pela rede e está sendo continuamente verificado, é extremamente difícil alterar ou adulterar os dados. Isto é o que torna a cadeia de blocos uma forma segura e transparente de armazenar e transferir informações.

² Um livro-razão é um registro contábil que documenta todas as transações financeiras de uma entidade.

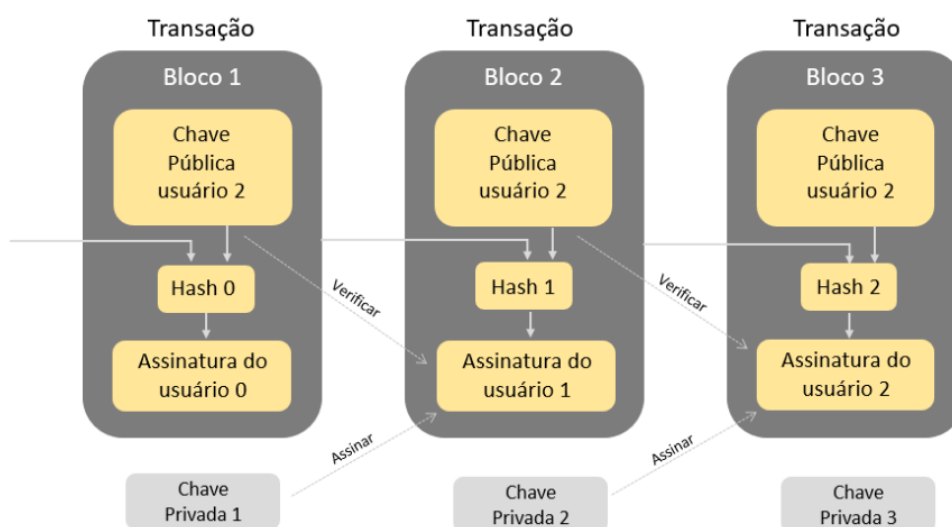


Figura 3 – Transações de registros de blocos *Blockchain*

Fonte: [Nakamoto 2008]

Realizar transações somente com as assinaturas digitais não garante que os dados cheguem ao usuário mesmo que a transação seja realizada. Logo, [Nakamoto 2008] afirma que uma forma de confirmar a ausência de uma transação é estar atento a todas as transações que ocorrem. Por esse motivo, foi criado o servidor de carimbo de tempos bem distribuídos em uma base ponto-a-ponto que se utiliza de sistemas de prova de trabalho³. O servidor de carimbos de tempo é responsável por gravar a data e hora de uma função *hash* que será carimbada. A Prova de trabalho é adicionada ao bloco *Nonce*. Isso causa a necessidade da finalização de todo o trabalho realizado no bloco, sendo necessário realizar todo o trabalho novamente caso seja preciso mudar algo no bloco. Dessa forma, para fraudar um conteúdo de um bloco é preciso fraudar todos os blocos presentes no conjunto da rede *Blockchain*, fazendo com que essa prática não seja viável ou possível com os recursos computacionais atuais.

2.2 SMART CONTRACTS

A *Blockchain* fornece as bases para diversos domínios de aplicações, incluindo criptomoedas, IOT, Reputação de Sistemas e Redes Sociais [Zheng et al. 2018] e aplicativos descentralizados (DApps) [Johnston et al. 2014]. Os DApps são aplicativos baseados em *Blockchain* que utilizam protocolos de código aberto para executar um conjunto de regras específicas para criar um mercado justo e seguro. Ao contrário dos aplicativos tradicionais, os DApps não são

³ Prova de trabalho é um mecanismo utilizado em tecnologias de *Blockchain*, como o Bitcoin, onde mineiros competem para resolver problemas matemáticos, validando transações. O vencedor adiciona um bloco à cadeia e recebe recompensas.

controlados por qualquer indivíduo ou entidade, mas sim por vários usuários em um sistema descentralizado. Embora os DApps possam ser construídos em várias plataformas *Blockchain*, a maioria deles são desenvolvidos em plataformas de *smart contracts*, como Ethereum. Esses aplicativos têm múltiplas vantagens, incluindo segurança, desempenho, escalabilidade e transparência.

Os *Smart Contracts* podem ser definidos como um código executado na *Blockchain* que facilita, executa e impõe os termos de um contrato entre partes não confiáveis [Alharby e Moorsel 2017]. Essa tecnologia não necessita de uma terceira parte confiável para validar uma transação, como nos contratos tradicionais. Diante disso, o custo por transação realizada torna-se menor comparado a esses.

A estrutura básica de um *Smart contract* é composta por valor, endereço, funções e estado [Bahga e Madisetti 2016]. O valor é toda a informação monetária armazenada no contrato. O endereço é a hash de identificação *Blockchain* onde o contrato inteligente está armazenado. As funções são os códigos que especificam como as partes podem interagir com o contrato. Por fim, o estado consiste na condição atual do contrato, que pode ser alterado ao longo do tempo com base nas interações com o mesmo. Assim, quando o usuário deseja realizar uma transação com o contrato, é preciso apenas enviar o conteúdo da transação para o endereço do contrato, no qual será realizada a prova de consenso para em seguida ser atualizado em conformidade. A Figura 4 apresenta a estrutura básica de um *smart contract*.

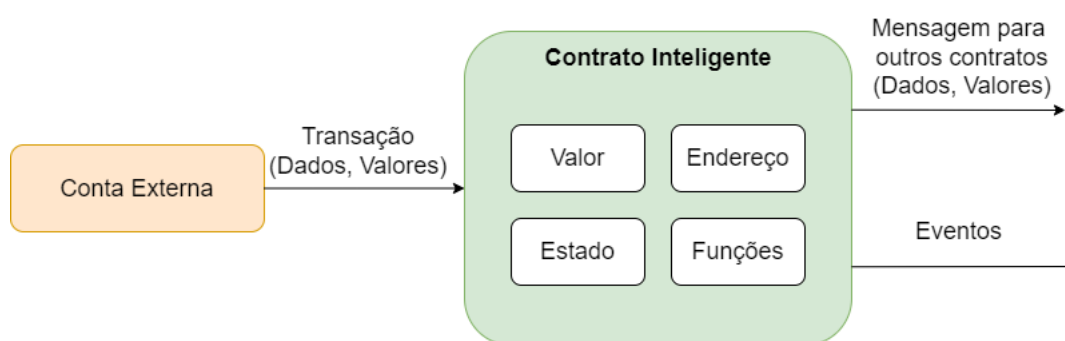


Figura 4 – Estrutura Básica de um Smart Contract

Fonte: Adaptado de [Bahga e Madisetti 2016]

Segundo [Alharby e Moorsel 2017], os contratos são capazes de receber, ler e escrever para seu armazenamento privado, armazenar dinheiro em seu saldo de conta, enviar e receber mensagens ou dinheiro de usuários, ou outros contratos, ou até mesmo criar novos contratos. A figura 5 e 6 mostra a diferença entre as transações com contratos usuais e os contratos inteligentes.

O desenvolvimento e execução dos smart contract são baseados em dois principais com-



Figura 5 – Transações com Contratos Tradicionais

Fonte: [Osetskyyi 2018]



Figura 6 – Transações com Smart Contracts

Fonte: [Osetskyyi 2018]

ponentes: a linguagem de programação e a plataforma de execução. A linguagem de programação mais utilizada para o desenvolvimento de *smart contracts* é a linguagem de programação Solidity [Solidity 2022], que foi desenvolvida para criação de aplicativos complexos para a plataforma Ethereum. Os códigos do contrato inteligente Ethereum são escritos em uma linguagem de bytecode baseada em pilha e executados na Ethereum Virtual Machine (EVM).

Para o desenvolvimento dos Smart Contracts, são identificadas diversas plataformas *Blockchain* usadas para essa finalidade, no entanto, o Ethereum é a plataforma mais utilizada para o desenvolvimento dos mesmos [Alharby e Moorsel 2017]. Ethereum é uma plataforma global de código aberto para aplicações descentralizadas que permite o desenvolvimento e execução dos contratos inteligentes, como, por exemplo aplicações de criptomoedas, redes sociais que requerem algum tipo de transações e ativos digitais, entre outros. Seu uso em contratos inteligentes é conhecido devido a flexibilidade do sistema de Turing-completeness, que permite a personalização dos contratos inteligentes em seu desenvolvimento (BUTERIN, 2015). Além da Ethereum, existem algumas outras plataformas que podem ser utilizadas para desenvolver *smart contracts*, como Hyperledger [Hyperledger 2022], Corda [Corda 2022], BigChainDB

[Bigchaindb 2022], entre outras.

Os mecanismos operacionais de um *smart contract* são baseados na assinatura entre duas ou mais partes, conforme realizado em contratos convencionais. No entanto, diferentemente de um contrato tradicional, as partes não são obrigadas a cumprir as obrigações do contrato por meio de ações judiciais ou penalidades legais. Em vez disso, o contrato inteligente usa o código incorporado nele para implementar as obrigações acordadas entre as partes. O código é executado automaticamente quando as condições especificadas no contrato são atendidas. Por exemplo, se um contrato inteligente especifica que a parte A deve pagar à parte B se determinadas condições forem atendidas, o contrato inteligente executará automaticamente o pagamento quando essas condições forem atendidas.

Segundo [Wang et al. 2018] uma vez que o smart contract é assinado por todas as partes, eles são adicionados à *Blockchain* como códigos de programa (por exemplo, uma transação Bitcoin) e são registrados na *Blockchain* após serem distribuídos pela rede P2P e verificados pelos nós. Os contratos inteligentes encapsulam uma série de estados predefinidos e regras de transição, cenários que ativam a execução do contrato (como em um dado momento ou quando ocorre um evento específico), respostas em um cenário específico, etc. A *Blockchain* acompanha o status em tempo real dos contratos inteligentes e executa o contrato após determinadas condições de disparo serem atendidas.

2.3 MODELOS ARQUITETURAIS DE SOFTWARE

A arquitetura de software é o conjunto de princípios, diretrizes, estruturas e padrões que orientam o desenvolvimento de um sistema de software. Ela descreve como o sistema é dividido em componentes, como esses componentes se comunicam e colaboram entre si, e como o sistema atende aos requisitos de qualidade, tais como escalabilidade, segurança e facilidade de manutenção [Garlan 2001].

A arquitetura de software é importante porque ajuda a garantir que o sistema seja desenvolvido de forma coerente e organizada. Isso facilita a identificação e correção de problemas, além de tornar o processo de manutenção mais fácil. Desse modo, a construção de uma arquitetura de software deve se basear em quatro principais conceitos que incluem:

- **Componentes:** os componentes são unidades independentes de software que podem ser combinadas para formar o sistema completo. Eles são projetados para serem reutilizáveis e podem ser substituídos ou atualizados sem afetar o funcionamento do sistema como um

todo;

- **Padrões de projeto:** os padrões de projeto são soluções comuns para problemas comuns de arquitetura de software. Eles fornecem uma abordagem testada e comprovada para resolver problemas específicos, como gerenciamento de dependências entre componentes ou comunicação entre processos;
- **Interações entre componentes:** a arquitetura deve especificar como os componentes se comunicam e colaboram entre si. Isso inclui detalhes sobre como os dados são transmitidos entre componentes e como as operações são sincronizadas.
- **Qualidade de software:** a arquitetura deve garantir que o sistema atenda aos requisitos de qualidade, como escalabilidade, segurança e facilidade de manutenção. A arquitetura deve fornecer as ferramentas e mecanismos para garantir que o sistema seja escalável e possa ser facilmente atualizado e modificado.

Segundo [Garlan 2001], uma arquitetura de software geralmente desempenha um papel fundamental como uma ponte entre os requisitos e o código. Em suma, uma arquitetura descreve sua estrutura bruta usando uma ou mais visualizações. A arquitetura em uma perspectiva mostra uma série de escolhas de design de nível superior, incluindo como o sistema é construído por componentes interativos, onde se encontram as principais rotas de interação e quais são as principais características desses componentes. Portanto, uma descrição arquitetônica ideal fornece informações suficientes para permitir uma análise e avaliação crítica de alto nível. A Figura 7 apresenta o design de uma arquitetura de software com pontes entre os componentes.

A arquitetura, ao apresentar uma representação abstrata (ou modelo) de um sistema, revela algumas de suas características enquanto esconde outras. Essa representação fornece uma visão geral inteligível do sistema, possibilitando que os projetistas analisem a capacidade do sistema de atender a certos requisitos e sugerindo uma estratégia para a construção e organização do sistema.

Existem várias maneiras de se estruturar um sistema de software, tais como a arquitetura baseada em serviços (SOA), arquitetura hexagonal e outras, cada uma com suas particularidades e objetivos. É fundamental escolher a abordagem correta para atender às necessidades do projeto e garantir a qualidade, escalabilidade e manutenibilidade do sistema.

Dado o exposto, pode-se concluir que a arquitetura de software é fundamental para garantir a qualidade, escalabilidade e manutenibilidade de um sistema. Ela define a estrutura

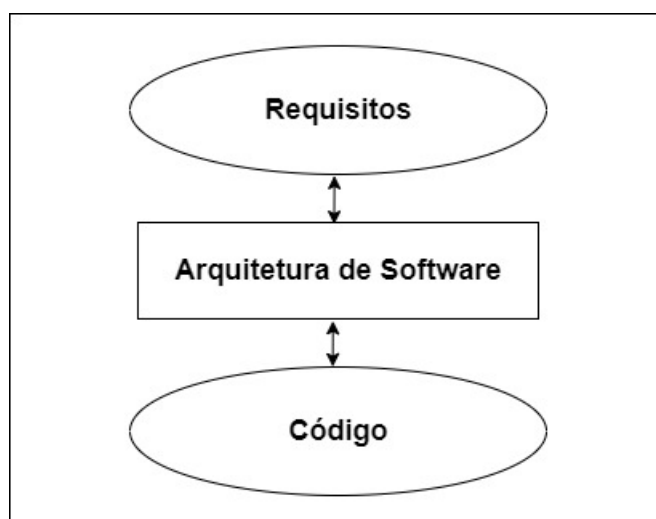


Figura 7 – Arquitetura de software como uma ponte

Fonte: Adaptado de [Garlan 2001]

geral do sistema, sua organização e interação entre componentes, bem como sua capacidade de adaptação a mudanças. A escolha da arquitetura adequada é crucial para garantir que o sistema possa atender às necessidades do negócio e ser facilmente mantido e atualizado. A implementação de boas práticas de arquitetura de software também pode ajudar a garantir que o sistema seja escalável, confiável e seguro.

2.3.1 Arquitetura Orientada a Serviços

A arquitetura Orientada a Serviços (SOA, do inglês, Service-Oriented Architecture) é uma abordagem para o desenvolvimento de sistemas de software baseada na criação de serviços independentes que podem ser reutilizados em vários sistemas. Ela se concentra em projetar sistemas como conjuntos de serviços que se comunicam entre si para realizar tarefas específicas.

Os serviços em uma arquitetura SOA são projetados com base em protocolos de comunicação padrão, como o SOAP ou o REST, o que permite que eles sejam acessados por diferentes sistemas e plataformas. Isso também facilita a integração de novos sistemas e aplicativos. Além disso, os serviços podem ser gerenciados e monitorados de forma centralizada, o que facilita a manutenção e o gerenciamento do sistema.

A ideia principal por trás da arquitetura SOA é que os serviços possam ser utilizados de forma independente. Isso significa que os serviços podem ser atualizados ou substituídos sem afetar o funcionamento do sistema como um todo. Desse modo, os serviços podem ser facilmente integrados em sistemas diferentes, o que aumenta a flexibilidade e escalabilidade do sistema. Em suma, a essência de um serviço está no fornecimento de serviço ser independente da aplicação

que o usa [Turner et al. 2003].

A Figura 8 ilustra o conceito da arquitetura SOA. Os componentes presentes comunicam entre si e fornecem as informações necessárias para cada agente. Os agentes que desejam solicitar informações de algum serviço devem especificar o serviço e emitir uma solicitação ao seu provedor para que haja a conexão com os serviços especificados e a troca de informação entre os agentes. Os provedores de serviços são o componente que projeta e implanta os serviços e especifica as interfaces de acesso ao mesmo.

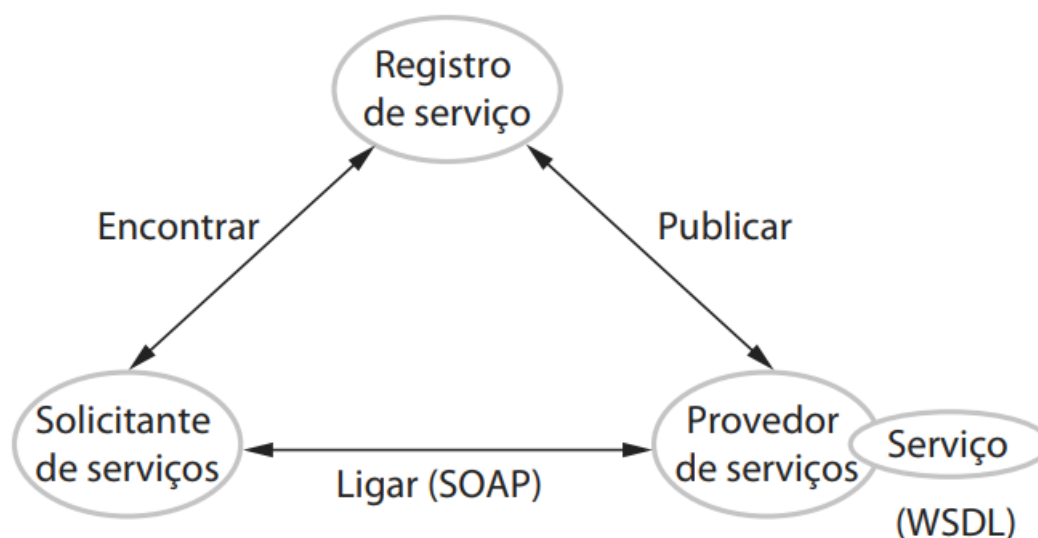


Figura 8 – Arquitetura orientada a Serviço

Fonte: [Sommerville 2011]

Os protocolos de *web services* abrangem todos os aspectos da SOA [Erl 2008]. Estes fornecem mecanismos para segurança da informação (WS-Security), descrição de serviços (WSDL) e a descrição semântica de serviços. Além disso, fornecem também mecanismos para otimização de performance (WS-Reliability) e assinatura digital (WS-DigitalSignature). Estes protocolos são importantes para a comunicação entre serviços e aplicativos, fornecendo recursos para permitir que os serviços sejam descobertos, compartilhados e integrados de forma segura.

Para facilitar o entendimento humano e utilizar uma linguagem padrão de comunicação de máquina, o *web service* faz uso da XML. A XML (Extensible Markup Language) é uma linguagem de marcação baseada em texto desenvolvida para compartilhar dados estruturados entre sistemas. Por ser uma linguagem padronizada, pode ser lida por qualquer sistema, independentemente da plataforma usada para criar ou ler os dados.

Segundo [Sommerville 2011] os principais padrões para o SOA de web são: SOAP, WSDL e WS-BPEL. O primeiro é responsável por oferecer suporte à comunicação entre os

serviços. O segundo tem o objetivo de realizar a definição de interface de serviço. O último é o padrão desenvolvido para definir programas de processo que possuam diversos serviços, denominado como linguagem *workflow*.

Com o surgimento do padrão de comunicação com API REST, os atuais padrões de *web services* têm sofrido críticas por ser um padrão "pesado". As APIs REST são melhores opções porque são mais flexíveis, escaláveis e mais fáceis de usar do que *Web Services*. AS APIs REST estão mais preparadas para o funcionamento em nuvem, têm suporte a formatos de dados comuns, como JSON e XML, possuem fácil implementação e podem ser utilizadas por várias plataformas.

Dado o exposto, a arquitetura SOA é uma abordagem de design de software ideal para organizações que buscam maior flexibilidade, reutilização de serviços e escalabilidade. Ao usar os principais princípios da arquitetura SOA, as empresas podem otimizar a integração e o desenvolvimento de aplicativos, proporcionando um alto nível de performance e estabilidade. Além disso, a arquitetura SOA oferece recursos de segurança robustos, permitindo que as empresas mantenham a conformidade com as regulamentações de segurança e privacidade. Por esses motivos, a arquitetura SOA pode ser uma ferramenta extremamente útil para as organizações que buscam obter vantagem competitiva no mercado.

2.3.2 Arquitetura Orientada a Eventos

Arquitetura Orientada a Eventos (EDA, do inglês, Event-Driven Architecture) é uma abordagem que delinea uma metodologia para conceber e implementar aplicações e sistemas nos quais eventos são transmitidos entre componentes de software e serviços de forma desacoplada [Maréchaux 2006]. O EDA feito para capturar, comunicar e processar eventos entre serviços desacoplados. Isso significa que os sistemas podem permanecer assíncronos enquanto compartilham informações e realizam tarefas. É importante destacar que a EDA não busca substituir, mas sim complementar a SOA. Enquanto a SOA é geralmente mais apropriada para uma troca de solicitação/resposta, a EDA introduz recursos para processos assíncronos de longa duração.

Uma das características distintivas da arquitetura orientada a eventos é a assincronicidade, que permite a eficiente resposta a eventos sem interrupção de outras operações, e o desacoplamento, que promove a independência entre os componentes, facilitando a manutenção, evolução e reutilização de partes específicas do sistema.

A flexibilidade e extensibilidade são características marcantes, permitindo a adição de novos componentes ou modificações sem impacto significativo no sistema. A introdução de novos

tipos de eventos viabiliza a implementação eficaz de funcionalidades adicionais. A escalabilidade é facilitada pela comunicação assíncrona e desacoplamento, possibilitando a distribuição eficiente de tarefas para uma resposta ágil a demandas variáveis.

2.4 REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS

A elaboração de requisitos é uma das etapas mais importantes no processo de desenvolvimento de software, pois tem como objetivo definir o que um sistema deve fazer e como ele deve se comportar. Para isso, é necessário identificar os requisitos funcionais e não funcionais, que juntos compõem as especificações do sistema. Os requisitos funcionais são uma parte fundamental do processo de desenvolvimento de software, pois descrevem as funcionalidades que o sistema deve apresentar para atender às necessidades do usuário ou do negócio. Eles são definidos a partir dos objetivos da aplicação e dos requisitos do usuário, e são utilizados como base para o planejamento e execução do projeto [Pressman e Maxim 2021].

Os requisitos funcionais são importantes porque garantem que o software desenvolvido atenda às necessidades do usuário, entregando as funcionalidades esperadas. Isso permite que o desenvolvedor tenha uma visão clara do que precisa ser implementado e quais são as prioridades do projeto.

Além dos requisitos funcionais, é importante destacar a importância dos requisitos não funcionais na arquitetura de software. Os mesmos são aqueles que especificam os atributos de qualidade do sistema, como desempenho, segurança, escalabilidade, disponibilidade, entre outros [Pressman e Maxim 2021]. Eles não se relacionam diretamente com as funcionalidades do sistema, mas sim com as características que afetam a qualidade e a usabilidade do mesmo.

Dessa forma, é fundamental que os requisitos não funcionais sejam levados em consideração durante todo o processo de desenvolvimento de software, desde a análise dos requisitos até a implementação e teste do sistema. É preciso garantir que a arquitetura de software atenda aos requisitos não funcionais especificados, a fim de garantir a qualidade e a efetividade do sistema como um todo.

2.5 O MODELO C4 DE DOCUMENTAÇÃO PARA ARQUITETURA DE SOFTWARE

O modelo arquitetural C4 apresenta como característica principal uma documentação feita para diferentes públicos, sendo de fácil elaboração e compreensão. Devido a essas características, é um padrão amplamente utilizado em várias investigações no campo da Engenharia de Software [Rowse 2022, Tian et al. 2022]. O C4 é fundamentado em um conjunto de diagramas

hierárquicos baseados em camadas, proporcionando uma descrição abrangente da arquitetura de software em diferentes níveis de abstração. À medida que a profundidade aumenta, ocorre uma ampliação nos níveis de detalhes [Brown 2022].

A nomenclatura "C4"reflete a quantidade de camadas presentes no modelo, nomeadamente: Contexto, Contêiner, Componentes e Código. O nível de contexto visa oferecer uma visão panorâmica, permitindo a visualização das fronteiras e relacionamentos com elementos externos. O sistema é representado centralmente como uma entidade, com os demais itens ao seu redor, destacando as interações. Neste contexto, detalhes específicos não são prioritários, visto que a ênfase recai sobre uma visão geral do cenário do sistema. Esse enfoque é particularmente relevante para audiências não técnicas, abrangendo atores, papéis, personas, entre outros, em detrimento de tecnologias e protocolos de baixo nível. A Figura 9 apresenta a visão geral das 4 camadas.

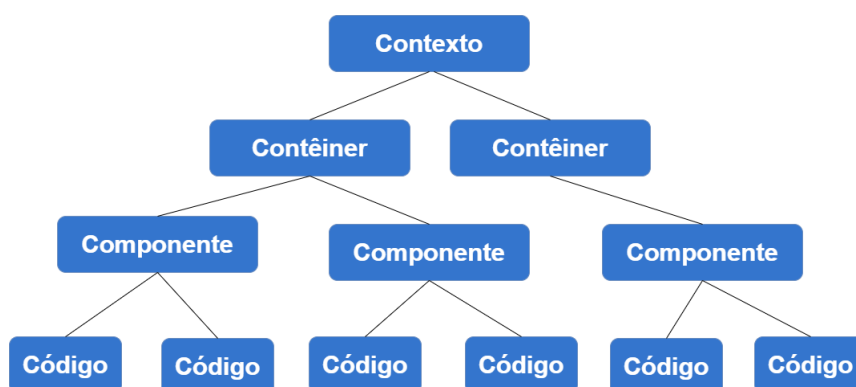


Figura 9 – Visão geral das quatro camadas do C4 model

Fonte: Autor

No segundo nível, o de Contêiner, a arquitetura é simplificada apresentada, destacando as tecnologias empregadas e os processos de comunicação. Este diagrama, mais detalhado, é valioso para as equipes de desenvolvimento, arquitetos e DevOps.

A terceira camada corresponde aos Componentes, onde os contêineres são exibidos com foco nas funcionalidades, revelando suas responsabilidades e detalhes tecnológicos. Por fim, o nível de Código, opcional, é recomendado para componentes complexos e de relevância, visando apresentar de maneira simples a implementação de um componente.

2.6 CONSIDERAÇÕES FINAIS

Este capítulo apresentou os principais conceitos para o entendimento deste trabalho. Na seção 2.1 mostrou-se a definição da tecnologia *Blockchain*, suas principais características

funcionais, os aspectos de composição de uma rede *Blockchain* e os atributos necessários para realização de uma transação. A Seção 2.2 abordou o conceito de Smart Contract, tecnologia responsável por executar regras acordadas entre partes não confiáveis de forma segura, transparente e auditável. A Seção 2.3 definiu os principais conceitos de arquitetura de software, arquitetura SOA e arquitetura orientada a eventos, permitindo entender melhor os principais componentes, atributos e características presentes em cada design arquitetural apresentado na seção. Na Seção 2.4 é apresentado o conceito de requisitos funcionais e não funcionais na etapa de desenvolvimento de um software. Por fim, a Seção 2.5 apresenta o modelo C4 de documentação para arquiteturas de software utilizado para documentar e facilitar a leitura por diferentes públicos sobre uma arquitetura de software.

Esses conceitos são utilizados para a construção da arquitetura de software especificada nesta dissertação. Isto é, a tecnologia *Blockchain* junto com os smart contracts pode fornecer uma camada a mais de segurança e transparência dos processos internos e das informações armazenadas em uma aplicação. A arquitetura de software é fundamental para projetar a estrutura de um sistema de software proposto neste trabalho. Esses conceitos também são utilizados em trabalhos encontrados no estado da arte. Porém, há uma lacuna na literatura, a qual se trata de não haver nenhum trabalho que faça uso desses conceitos em conjunto com arquiteturas de software para plataformas de streaming que mantenha a estrutura operacional existente e inclua a criação de serviços intermediários para gerenciar a entrada de informações das plataformas e realize o controle de audiência e monetização baseado na tecnologia *Blockchain*. A especificação da arquitetura proposta é descrita no Capítulo 4.

3 TRABALHOS CORRELATOS

Este capítulo descreve a metodologia, as estratégias de pesquisas e os trabalhos correlatos investigados no desenvolvimento deste trabalho. A seção 3.1 apresenta a metodologia de pesquisa, os repositórios de periódicos científicos em ciência da computação, os critérios de inclusão e exclusão dos trabalhos selecionados e a estratégia de extração e análise dos resultados. A seção 3.2 apresenta respectivamente a análise do estado da arte apresentando os trabalhos identificados com as linhas de pesquisa: Arquiteturas utilizadas em plataformas de streaming e uso da tecnologia *Blockchain* em plataformas de streaming. A seção 3.3 expõe os resultados e discussões sobre os trabalhos identificados. Por fim, na seção 3.4 são apresentadas as considerações finais desta seção.

3.1 METODOLOGIA DE REVISÃO DO ESTADO DA ARTE

A metodologia de pesquisa utilizada neste trabalho consiste em uma revisão do estado da arte. A revisão do estado da arte é um método empírico que tem por objetivos identificar, avaliar e interpretar questões de pesquisa, área de um tópico, ou fenômeno em uma pesquisa. O objetivo desta revisão é investigar os desafios nos processos de transparência, métodos de pagamentos, desenvolvimento dos smart contracts e modelos arquiteturais em plataformas de streaming utilizando a tecnologia *Blockchain*. Neste trabalho foi utilizada a adaptação dos modelos [Xiao e Watson 2019, Rocha et al. 2014] para a análise do estado da arte. Conforme ilustrado na Figura 10, o processo da pesquisa teve início com a etapa de planejamento, responsável por enfatizar o desenvolvimento de um protocolo para o estudo, depois foi realizada a etapa de realização de pesquisa, no qual serão aplicados os critérios de seleção, procedimentos de avaliação da qualidade, estratégia de extração de dados e a síntese dos dados extraídos. Por fim, foi realizada a análise dos resultados em busca de lacunas não investigadas na literatura.

3.1.1 Objetivo

Os principais objetivos desta pesquisa, são: i) reunir conhecimento sobre a utilização da tecnologia *Blockchain* em Plataformas de Streaming; ii) entender como a tecnologia *Blockchain* pode auxiliar os criadores de conteúdos dessas plataformas; iii) investigar os modelos arquiteturais utilizados pelas plataformas de streaming; iv) identificar possíveis lacunas de pesquisa e questões em aberto que serão apresentadas com direções futuras.

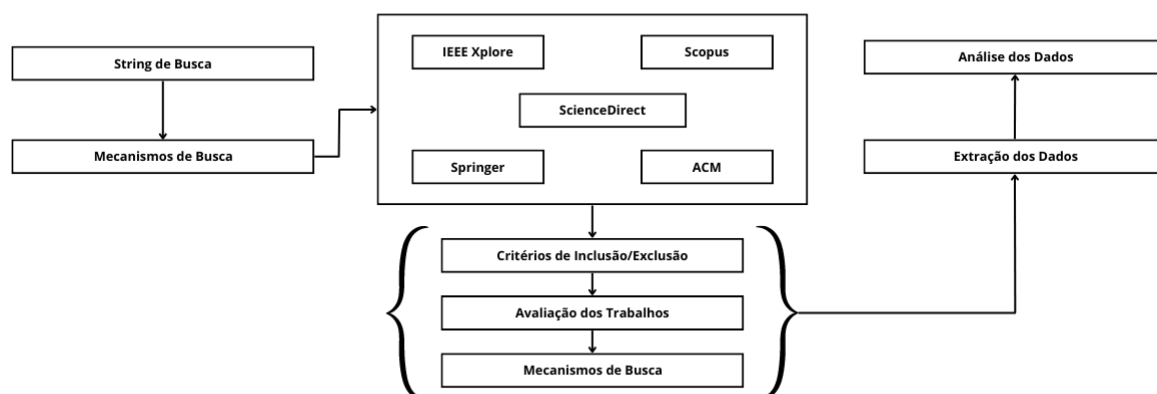


Figura 10 – Metodologia de pesquisa

Fonte: Autor

3.1.1.1 Informações da pesquisa

Para realizar uma cobertura do tema que englobe o maior número de trabalhos relevantes, foi utilizada a estratégia de busca automatizada, considerando as principais bases de dados de Ciência da Computação e Engenharia de Software, sendo elas: ACM Digital Library, IEEE Xplore, Springer Link, ScienceDirect e Scopus. As fontes de dados selecionadas abrangem a maioria dos trabalhos presentes em Conferências, workshops e periódicos da área de Ciência da Computação e Engenharia de Software.

3.1.2 Estratégia de Pesquisa

Para a definição da string de busca, foram aplicadas duas estratégias diferentes: a primeira consiste em analisar o objetivo e questões da pesquisa. Como exposto anteriormente, o objetivo da pesquisa é investigar o uso da tecnologia *Blockchain* em Plataformas de streaming, como essa tecnologia pode auxiliar os criadores de conteúdos dessas plataformas e os principais modelos arquiteturais utilizados nessas aplicações. Desse objetivo, foram extraídos os termos: *Blockchain*, *Streaming Platforms* e *Architecture*. A segunda etapa foi a realização da pesquisa piloto, seu objetivo é realizar uma pesquisa em pequena escala capaz de definir meios e métodos para a coleta de dados definitiva [Shamseer et al. 2015]. A pesquisa piloto foi realizada no Google acadêmico, usando os termos-chave obtidos na primeira etapa, para identificar outros termos relevantes utilizados no mesmo segmento. Os termos, foram localizados nos títulos, resumos, palavras-chave e conclusões dos artigos selecionados. Para os dois termos chaves extraídos e características que podem auxiliar os criadores de conteúdo, foram encontrado os seguintes

conjuntos de termos:

- **Blockchain:** Distributed Ledger e Smart Contract;
- **Características:** Privacidade, Segurança, Transparência e Pagamentos;
- **Plataformas de streaming:** na busca piloto não foi possível identificar termos com sentidos semelhantes ao mesmo.
- **Arquitetura:** na busca piloto não foi possível identificar termos com sentidos semelhantes ao mesmo.

Com base nos termos encontrados, foi criada a string de busca conectando os termos identificados com operadores booleano. Assim, foi utilizado o operador AND para conectar termos chaves da pesquisa, e o OR para conectar termos que estavam na mesma categoria. Desse modo, a string de busca ficou da seguinte forma: (Blockchain OR "Distributed Ledger" OR "Smart Contract") AND (Privacy OR Security OR Transparency OR Payments) AND ("Streaming Platforms") AND ("Architecture"). Para melhor visualização, a Figura 11 mostra a string de busca com os termos utilizados e sua conexão lógica.

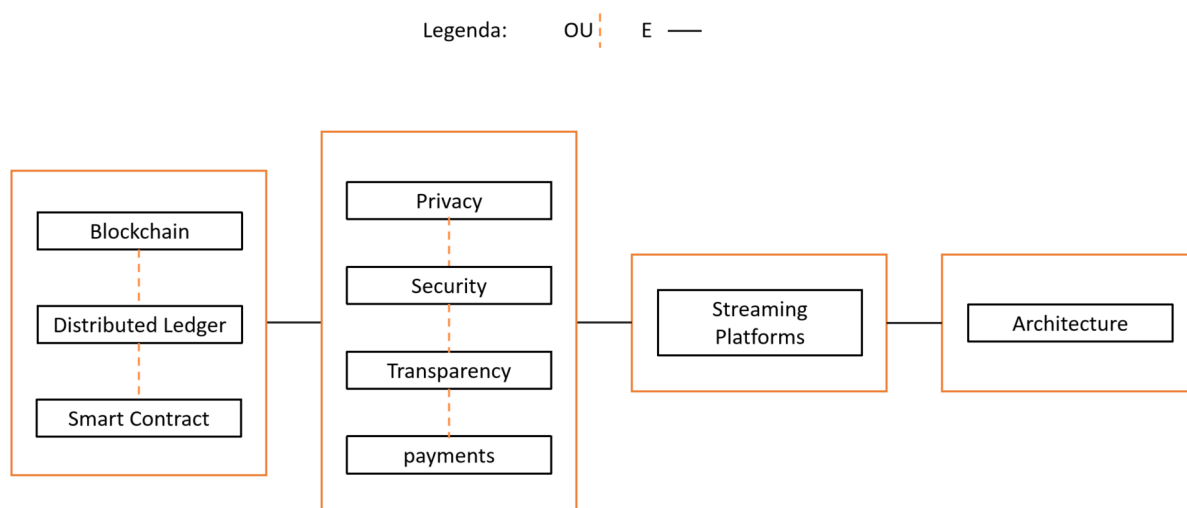


Figura 11 – String de Busca

Fonte: Autor

Esta string de busca foi projetada para encontrar o número máximo de trabalhos que investiguem o uso da tecnologia *Blockchain* em plataformas de streaming, no entanto, alguns artigos podem não ser identificados por essa string de busca por usar terminologias diferentes para descrever o tema selecionado. Além disso, foi adicionado ao filtro das bases de dados e

buscadores de trabalhos acadêmicos as restrições de data de publicação com intervalo entre os anos de 2017 a 2022, antes de iniciar o processo de busca em cada fonte de dados. Em consequente à etapa anterior, foram definidos os critérios de exclusão e inclusão especificados nas tabelas 1 e 2. A utilização desses critérios é necessária para obtenção de resultados com maior qualidade no final dessa pesquisa.

Descrição dos critérios de Exclusão
Artigos Duplicados
Artigos Publicados até 2016
Artigos Sem Disponibilidade do Texto Completo
Artigos com Tamanho Menor de 5 Páginas (<i>Short Papers</i>)
Estudos não Relacionados com a Pesquisa

Tabela 1 – Critérios de Exclusão

Fonte: Autor

Descrição dos critérios de Inclusão
Artigos Publicados Entre 2017 a 2022
Artigos Escritos em Inglês
Artigos Publicados em Periódicos ou Revistas
Artigos que Abordam o Uso da Tecnologia <i>Blockchain</i> em Plataformas de Streaming

Tabela 2 – Critérios de Inclusão

Fonte: Autor

Concluindo as estratégias de busca, foi especificada a estratégia de extração de dados. Decidiu-se que se daria em forma de leitura total dos trabalhos que fossem incluídos na revisão, para não perder nenhum detalhe presente no mesmo. Relacionado à estratégia adotada, foi desenvolvido um formulário de extração de dados contendo as informações a seguir: Nome do artigo, Nome do(s) autor(es), Data da extração, Ano de publicação, Fonte do artigo, Objetivo do estudo, além das seguintes questões: "Qual o uso da tecnologia *Blockchain*?", "quais *Stakeholders* estão presentes no estudo?", "Os smart contracts fazem parte da pesquisa?" e "o modelo arquitetural é definido na pesquisa?" A tabela 3 mostra o formulário desenvolvido com os dados do estudo e suas descrições. Por fim, foi desenvolvida uma categorização de todos os trabalhos selecionados.

3.1.3 Dados do estudo

Os resultados obtidos na revisão da literatura foram inseridos da ferramenta Mendeley [Mendeley 2023]. Essa ferramenta consiste em um aplicativo web ou desktop que permite realizar o gerenciamento de bibliografias. Para isso, foram criadas pastas individuais para cada base de dados (Pasta Scopus, pasta IEEE, etc.). Por fim, foi criada uma pasta para os trabalhos

Dados do Estudo	Descrição
Nome do Artigo	-
Nome do(s) Autor(es)	-
Data de Extração	-
Ano de Publicação	-
Objetivo do Estudo	-
Qual o uso da tecnologia <i>Blockchain</i> ?	-
Quais <i>Stakeholders</i> estão presentes no estudo?	(Plataforma de streaming, Anunciante, Criador de conteúdo, Telespectador)
Os smart contracts fazem parte da pesquisa?	(Sim, Não)
O modelo arquitetura é definido na pesquisa?	(Sim, Não)

Tabela 3 – Formulário de Extração de Dados

Fonte: Autor

não selecionados, na qual, cada artigo foi acompanhado de uma nota na seção de observações explicando o motivo de sua exclusão da pesquisa.

A etapa de processamento de seleção é responsável por extrair da literatura os trabalhos identificados com a string de busca. Em razão das diferentes bases de dados utilizadas para coleta dos artigos, o mesmo estudo pode vir de fontes diferentes, diante disso, foi realizada a exclusão de trabalhos duplicados. Em seguida, foi realizada a seleção dos artigos levando em consideração somente o título. Logo após, foi realizada a leitura dos resumos dos artigos para garantir que o estudo atende aos critérios da pesquisa. Por fim, para garantir que os trabalhos não selecionados não atendem aos critérios de inclusão, os mesmos passaram por mais uma etapa de seleção, que consiste na leitura da conclusão do estudo para atestar que o mesmo não foi removido da pesquisa acidentalmente.

3.1.4 Coleta e avaliação dos dados

Ao finalizar o processamento de seleção, foi iniciado o processamento de coleta de dados. A coleta dos dados se deu em forma de leitura total dos artigos selecionados, com o objetivo de analisar minuciosamente cada trabalho. Para realizar a classificação adequada do artigo quanto à questão principal de pesquisa, serão utilizados parâmetros de análise do estudo acerca de suas metodologias e resultados. Os dados coletados foram inseridos no formulário de extração de dados (Tabela 3). Ao finalizar a fase de coleta de dados, foi iniciada a fase de avaliação dos trabalhos. Nesta fase, utilizaram-se os formulários de extração para comparar os resultados encontrados. O método de análise qualitativo foi utilizado para identificar os principais avanços e oportunidades identificados no uso da tecnologia *Blockchain* voltadas para o desenvolvimento de pesquisa no ecossistema das plataformas de streaming.

3.2 ANÁLISE DO ESTADO DA ARTE

Esta seção tem como objetivo apresentar os principais trabalhos acadêmicos que abordam sobre plataformas de streaming com *Blockchain*, visando fornecer uma visão geral sobre o assunto e contribuir para o avanço da pesquisa nessa área. Serão apresentados estudos que exploram diferentes aspectos do uso de *Blockchain* em plataformas de streaming, desde a proteção dos direitos autorais até a melhoria da experiência do usuário.

3.2.1 Arquiteturas de software

Diversos trabalhos são encontrados no estado da arte, os quais propõem soluções computacionais para o domínio das plataformas de streaming. Nessa subseção, consideraram-se os trabalhos que evidenciaram soluções computacionais que fazem uso de *Blockchain* ou definem modelos arquiteturais nessas plataformas.

Para um melhor entendimento do assunto apresentado a seguir, serão definidos brevemente os conceitos de *Application Programming Interface* (API) e *InterPlanetary File System* (IPFS). Uma API é um conjunto de instruções e padrões de programação para acessar um aplicativo de software baseado na web. Por exemplo, uma empresa pode disponibilizar uma API para desenvolvedores integrarem seus produtos ou serviços em seus próprios aplicativos [Cotton e Jr 1968, Fielding 2000]. Já o IPFS é um sistema distribuído, composto por um conjunto de nós que trabalham juntos para compartilhar e armazenar dados. Cada nó é auto soberano e pode armazenar, compartilhar e acessar dados de forma independente. [Dias e Benet 2016, Vimal e Srivatsa 2019].

O uso dessas ferramentas é visto em diferentes cenários em aplicações com *Blockchain*, suas características permitem o consumo de informações necessárias para o funcionamento das soluções encontradas no estado da arte. [Wöhler et al. 2021] definem duas arquiteturas de software orientadas a eventos que incorporam uma estrutura de mensagens, vinculada a componentes dedicados para lidar com operações de coleta e mudança de estado de *Blockchain*. Em virtude das particularidades inerentes à tecnologia *Blockchain*, a integração de soluções de software apresenta desafios significativos. Isso decorre do design arquitetônico exigido para conectar e coordenar elementos centralizados, como a lógica de *back-end*, com elementos descentralizados, tais como os contratos inteligentes (*smart contracts*) e o registro distribuído (*ledger*).

As arquiteturas baseadas em eventos foram projetadas para ter o funcionamento assíncrono devido à latência na execução e confirmação das transações, e ao fato de que mudanças ou

operações significativas que ocorrem na *Blockchain* são geradas como eventos. A *Blockchain* pode ser utilizada como plataforma autônoma, implementando toda a lógica da aplicação nos *smart contracts*, ou como uma ferramenta auxiliar. As arquiteturas propostas são fundamentadas no segundo cenário de uso, ou seja, arquiteturas híbridas baseadas em *Blockchain*. As Figuras 12 e 13 ilustra as visões gerais das arquiteturas.

A arquitetura descentralizada é presente em aplicações fundamentadas em plataformas de computação distribuída, e as soluções de software que adotam essa arquitetura são denominadas aplicativos descentralizados (DApp). A estrutura apresentada na Figura 12 representa um aplicativo corporativo composto por diversos serviços fracamente acoplados. O usuário acessa uma aplicação conectada à *Blockchain* por meio do navegador. Nesse contexto, um evento de envio é desencadeado, transmitindo informações para uma API. Este evento, por sua vez, instiga uma solicitação de acesso para armazenamento ou envio de informações na *Blockchain*. O usuário, ao receber uma notificação em sua carteira presente no navegador, responde enviando a chave privada. Após a confirmação, a transação é efetivada. Por fim, as informações são armazenadas tanto em um repositório convencional (SQL ou NoSQL) quanto na rede *Blockchain*.

Na arquitetura híbrida, especialmente em ambientes mais complexos, todas as informações do usuário são centralizadas na camada de API. Essa camada incorpora novos atributos dedicados à gestão de outros microsserviços inseridos na lógica de negócio, além disso, uma nova camada de armazenamento é adicionada para trazer maior segurança das informações. Os dois casos de uso apresentados permitem compensar a desvantagem de latência da tecnologia *Blockchain* com a utilização de uma API para gerenciar informações e eventos em tempo real e fornecer uma ambiente com confiança mútua e colaborativo, com potencial de substituir modelos de negócios existentes.

[Rodríguez-Gil et al. 2017] propõem uma arquitetura com o uso do Redis para entender os requisitos para plataformas interativas de transmissão ao vivo, com foco em laboratórios remotos. Neste trabalho, a arquitetura tem como objetivo ser altamente escalável, tanto com a entrada quanto com a saída de dados, para isso, a arquitetura foi dividida em quatro camadas: "Input", "Redis", "CamServers" e "Client-side". A Figura 14 ilustra a arquitetura proposta pelo autor.

A camada de *input* é composta por webcams, transcodificadores e *Feeders*. As webcams capturam e enviam as informações em diferentes formatos para os *Feeders*. Em alguns casos, os fluxos são transcodificados por meio do FFmpeg, dependendo da combinação de entrada e saída solicitada pelo cliente. Os *Feeders* são responsáveis por enviar as informações para a camada do

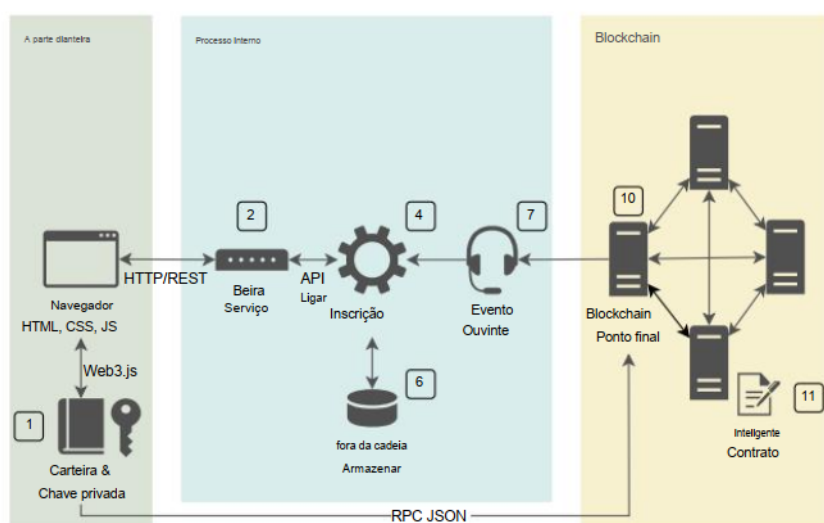


Figura 12 – Visão geral da arquitetura híbrida com back-end

Fonte: Adaptado por WÖHRER (2021)

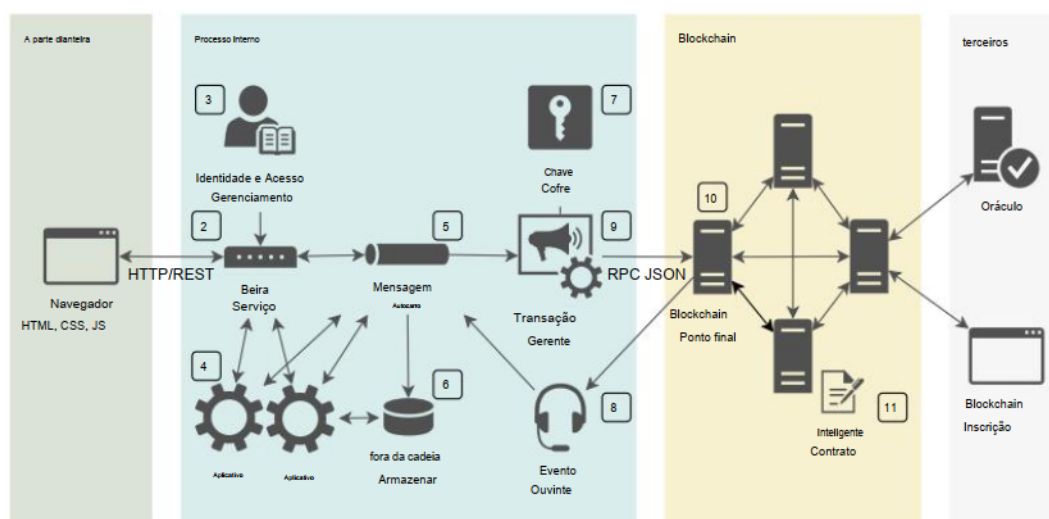


Figura 13 – Visão geral da arquitetura híbrida proposta

Fonte: Adaptado por WÖHRER (2021)

Redis.

A segunda camada é composta por um servidor Redis, responsável por armazenar e fornecer informações com alto nível de escalabilidade. O redis é popular por seu alto desempenho, escalabilidade e flexibilidade, tornando-o adequado para uma variedade de casos de uso, como cache, análise em tempo real, filas de mensagens, entre outros.

As informações armazenadas no Redis são fornecidas para a camada de CamServer responsável por gerenciar os fluxos que os usuários estão solicitando no momento. Por fim, a camada de cliente é composta por uma aplicação web que fornece bibliotecas e widgets que

podem solicitar os fluxos em diferentes formatos e renderizá-los.

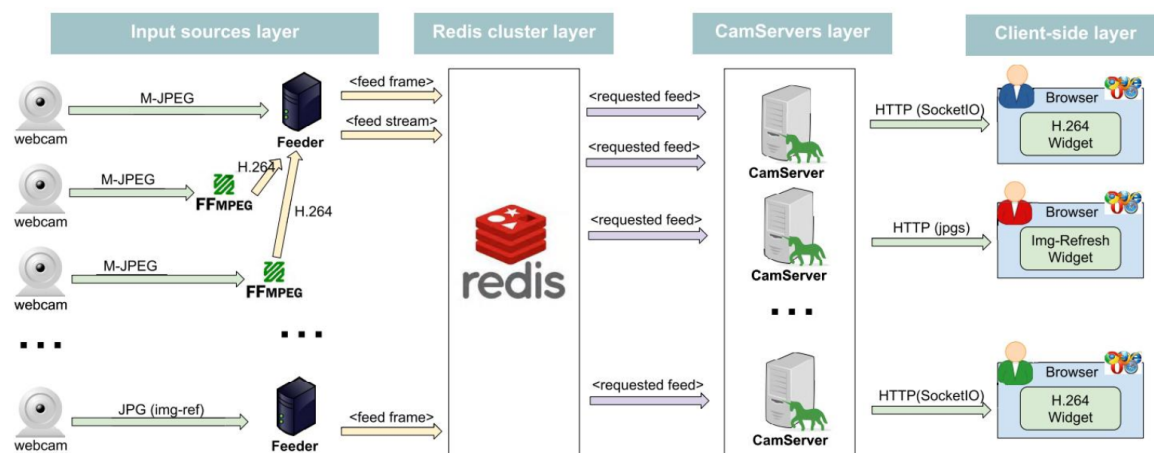


Figura 14 – Visão geral da arquitetura de plataformas interativas de transmissão ao vivo com redis

Fonte: Adaptado por RODRÍGUEZ-GIL (2017)

[Zhang 2022] propôs uma arquitetura orientada a serviços (SOA) que faz uso da tecnologia de computação em borda em uma plataforma de streaming. Com a arquitetura SOA e a computação em borda, é possível obter um processamento e transmissão de dados com maior qualidade e eficiência.

A arquitetura baseada em serviços é fundamentada nos conceitos de componentes, serviços e orientação a serviços. Esta arquitetura permite que aplicativos de diferentes fontes e plataformas se comuniquem por meio de uma camada orientada a serviços, que normalmente é composta por serviços web.

A computação de borda é um paradigma de computação distribuída que aproxima a computação e o armazenamento de dados das fontes de dados. A computação de borda é uma arquitetura, e não uma tecnologia específica, e uma forma de computação distribuída sensível à topologia e à localização. Devido a essas características, espera-se que isso melhore os tempos de respostas e economize largura de banda no uso de seus recursos.

Deste modo, [Zhang 2022] desenvolveu uma plataforma de streaming como uma prova de conceito para implementar a arquitetura proposta. O algoritmo de agrupamento orientado a SOA, fundamentado no algoritmo de agrupamento de nó de transmissão, é empregado para otimizar o fluxo de transmissão. Isso é alcançado por meio do uso de dois canais de comunicação independentes e da administração dos nós de transmissão entre a fonte e o destino das informações.

Para se adaptar ao desenvolvimento contínuo da rede, a estrutura e o roteamento da

plataforma foram realizados com a tecnologia openFlow. Esta tecnologia permite separar a camada de controle da camada de dados, gerenciando centralmente o equipamento de rede subjacente. Mediante o switch openFlow foi realizada a transmissão, processamento e controle dos dados para envio das informações de decisão de controle da aplicação. Os experimentos apresentados no trabalho demonstraram maior eficiência, precisão e menor latência na arquitetura SOA com computação em borda comparada ao método tradicional.

3.2.2 Plataformas de streaming com *Blockchain*

Esta subseção explora os efeitos do armazenamento descentralizado, imutabilidade e rastreabilidade no modelo de negócios das plataformas de streaming. Para alcançar esse objetivo, foram analisados estudos relacionados ao desenvolvimento e validação de soluções computacionais utilizando a tecnologia *Blockchain*. Essa análise considerou critérios como o tipo de rede, descrição do armazenamento de dados, sistema de pagamento por meio de *Blockchain* e transparência nas métricas de audiência usando *smart contracts*. A escolha da rede, por exemplo, influencia diretamente na viabilidade financeira da solução, enquanto o critério de armazenamento de dados afeta o poder computacional necessário para processar transações. Além disso, o sistema de pagamento por meio de carteiras digitais redefine a tradicional abordagem transacional, e a transparência nas métricas de audiência proporciona confiança e segurança nas transações.

As pesquisas recentes sobre a tecnologia *Blockchain* em plataformas de streaming concentraram esforços no desenvolvimento de aplicativos para fornecer maior transparência e controle sobre os serviços de mídia criados pelos usuários finais. Neste campo de pesquisa, [Liu et al. 2018] especificaram uma estrutura usando computação de borda na qual o tamanho do bloco na rede é adaptável para aplicativos de streaming de vídeo. Seguindo nessa mesma linha, [Naik et al. 2020, Barua e Talukder 2020] propõem a criação de uma plataforma de streaming descentralizada que faz uso do protocolo *InterPlanetary File System* (IPFS) e usa um sistema criptográfico de incentivo de recompensa para criadores de conteúdo com base na popularidade de seus vídeos.

Para reduzir o custo de armazenamento dos conteúdos audiovisuais, [Naik et al. 2020] realiza apenas o armazenamento da *hash* única do vídeo na *Blockchain*. Deste modo, quando um vídeo é acessado em sua plataforma, uma requisição de busca com a *hash* do vídeo é enviada à rede *Blockchain* e em seguida uma nova requisição é enviada para a plataforma IPFS que retorna o vídeo solicitado ao usuário.

A plataforma de streaming apresentada por [Barua e Talukder 2020] faz uso de um sistema de copyright e criptografia dos arquivos enviados em sua plataforma. O criador de conteúdo realiza *upload* do vídeo, em seguida, o arquivo é submetido a vários métodos de proteção a *copyright*, como, por exemplo a marca d'água digital. A marca d'água digital é um processo de incorporação de informações em conteúdo multimídia digital que não pode ser visto ou ouvido pelo usuário final. As informações incorporadas podem ser usadas para várias finalidades, incluindo proteção contra cópia, autenticação, verificação de propriedade e rastreamento de conteúdo. Ao finalizar o processo de *copyright* da aplicação, o vídeo é criptografado com criptografia assimétrica antes do *upload* do mesmo na plataforma IPFS. A chave privada é enviada para um gerenciador de acesso que é responsável por salvar a chave privada do vídeo na rede *Blockchain* e gerenciar o acesso ao conteúdo audiovisual da plataforma.

Em [Esmaili e Javidan 2020], uma arquitetura de compartilhamento de vídeo distribuído é apresentada na qual as questões de transparência e imutabilidade dos dados são abordadas usando a tecnologia *Blockchain*. No esquema proposto, a rede *Blockchain* se separou do armazenamento. Em razão do armazenamento de vídeo em *Blockchain* resulta em mineração dispendiosa nos nós da rede. O contrato inteligente da Ethereum é responsável por manter a validade, transparência, imutabilidade, integridade e segurança das transações, além de controlar o acesso ao vídeo. Do outro lado da arquitetura, há um armazenamento de arquivos distribuído compatível com Ethereum chamado Swarm para armazenamento de vídeo.

Atualmente, a audiência dos serviços de streaming responde por uma parcela significativa dos dados processados na internet. É fundamental garantir a entrega de conteúdo e fornecer mecanismos de integridade para serviços digitais para manter os usuários em uma plataforma. Como resultado, a avaliação de aplicativos de streaming baseados na tecnologia *Blockchain* é uma área de pesquisa que requer mais investigação, e algumas pesquisas foram realizadas nessa área. [Doan et al. 2020] conduziram um estudo no qual compararam aplicativos de compartilhamento de vídeo descentralizados com aplicativos de streaming tradicionais. O estudo em questão abordou a conectividade e o desempenho dos aplicativos avaliados, e os resultados mostraram que o aplicativo descentralizado possui um desempenho promissor apesar de apresentar uma latência maior no carregamento de seus vídeos do que o aplicativo tradicional.

A tecnologia *Blockchain* tem sido utilizada em conjunto com outras técnicas computacionais, e [Sathish et al. 2019] apresentam uma plataforma de streaming que utiliza aprendizado de máquina para monitorar transações e indicar conteúdos relacionados armazenados na rede *Blockchain*. A solução computacional apresentada pelos autores possui dois subsistemas de

processamento: i) processo On-Chain e ii) processo off-chain. O primeiro processo é responsável por gerenciar as transações, tokens, execução dos *smart contracts*, o processo de entrega e recebimento de conteúdo e execução do back-end com *Blockchain*. O segundo processo lida com todo o processo de gerenciamento e armazenamento nas bases de dados, controle de usuários e informações de conteúdos. O aprendizado de máquina é utilizado neste trabalho para analisar a correlação dos conteúdos presentes nos vídeos, fazendo uso de análise semântica de textos e falas por intermédio do uso de *Auto Speech Recognizer* para converter o fluxo de fala em texto, processar e analisar o seu conteúdo e sugerir para os usuários vídeos que mais se assemelham as suas características.

Na mesma linha, [Liu et al. 2018] propõem o uso de aprendizado por reforço profundo e redes neurais em uma plataforma de vídeo com *Blockchain* para melhorar os requisitos de entrega de conteúdo, enquanto [Yunqi 2021] apresentam uma plataforma de streaming de vídeo baseada em *Blockchain* que usa um WebTorrent de compartilhamento de arquivos e recompensa os participantes com moeda virtual. Nesse sistema, os usuários obtêm moedas com base em suas contribuições nos *uploads* realizados. Um *smart contract* é encarregado de aprovar a solicitação de recebimento da moeda, e a transação é executada por meio do uso da *Blockchain*.

O modelo de negócios da indústria fonográfica tem sido influenciado pelas tecnologias digitais disponibilizadas aos usuários finais pelas plataformas de streaming. Enquanto o consumo de música e podcasts proporciona à sociedade comodidade e customização dos serviços de streaming, o estado da arte relata problemas como atrasos nos pagamentos dos artistas, falta de transparência e centralização no modelo de negócios. Para abordar essas questões importantes, é proposto em [Yahya e Habbal 2021] um novo sistema de pagamento de royalties baseado na tecnologia *Blockchain*, no qual três atores principais (coletores, pagadores e receptores) são responsáveis por coletar, transferir e receber royalties via *Blockchain*.

A nova abordagem possui um novo modelo de funcionamento, nele as entidades responsáveis por cobrar os valores devidos não atuam com os meios de pagamentos, sendo responsáveis somente por fornecer as informações de cobranças para os clientes ou responsáveis financeiros dos pagamentos do royalties. Esses pagamentos são executados através da rede *Blockchain* mediante a execução das cláusulas contidas nos *smart contracts*.

[Chavan et al. 2019] desenvolveram uma plataforma de streaming de música baseada em *Blockchain* que usa um modelo de negócios pay-per-play (PPP) com preços fixos ou definidos pelo artista. O artista pode adicionar outros beneficiários à receita de seu produto musical na plataforma proposta. Nesse caso, o valor será distribuído com base no percentual estabelecido,

resultando em maior transparência e confiabilidade entre as partes interessadas.

Da mesma forma, [Awuor 2020] especifica uma plataforma de streaming na qual os usuários pagam royalties diretamente aos artistas pelas músicas que ouvem. Finalmente, [Kim e Kim 2020] criaram um modelo de distribuição de música baseado em contrato inteligente centrado no gerenciamento de direitos autorais. Os autores relatam que é possível fornecer todas as informações de serviços e transações relacionadas ao mercado musical com segurança e transparência utilizando a solução proposta.

No trabalho de [Darwich et al. 2023], é proposta uma estrutura de armazenamento distribuído empregando Blockchain em Plataformas de Streaming. Nesta abordagem, os vídeos são recebidos, particionados e armazenados em uma rede Blockchain distribuída, assegurando segurança, transparência e rastreabilidade dos conteúdos.

De acordo com as pesquisas apresentadas nesta seção, a tecnologia *Blockchain* oferece às plataformas de streaming segurança, rastreabilidade, transparência nas regras de negócios e a capacidade de fazer pagamentos sem o uso de uma instituição financeira. Embora os avanços no estado da arte sejam consistentes, há oportunidades de pesquisa que precisam ser exploradas.

A Figura 15 apresenta a categorização dos trabalhos investigados nesta dissertação. A seção seguinte discute os critérios de pesquisa utilizados neste trabalho, bem como uma análise das oportunidades identificadas.

Artigos	Blockchain				
	Rede Pública	Rede Privada	Descrição do Armazenamento	Transparência com Smart Contract	Pagamento com Blockchain
Mengting Liu et.al (2019)			✓	✓	✓
Mengting Liu et.al (2018)		✓			✓
Hemendra M. Naik et.al (2020)	✓		✓	✓	✓
Sailesh Kumar Sathish (2019)	✓		✓	✓	✓
Ahyoung Kim et.al (2020)		✓	✓		✓
Molud Esmaili et.al (2020)	✓				✓
MA Yunqi et.al (2021)	✓			✓	✓
Trinh Viet Doan et.al (2020)	✓				
Almigdad Yahya et.al (2021)		✓	✓	✓	✓
Sudarshan Chavan et.al (2019)	✓		✓	✓	✓
Fredrick Elder Awuor (2020)	✓		✓	✓	✓
Suvadra Barua et.al (2020)	✓			✓	✓
Mahmoud Darwich et.al (2023)	✓		✓		

Figura 15 – Categorização dos trabalhos investigados nessa seção

Fonte: Autor

3.3 DISCUSSÃO DOS RESULTADOS

Com base nos trabalhos correlatos apresentados neste capítulo, será realizada uma análise das características, comparando-as com os critérios de busca estabelecidos para este estudo. Durante essa avaliação, serão destacadas algumas lacunas abordadas ao longo desta dissertação.

Conforme mostrado na Figura 15, a maioria dos trabalhos mencionados faz uso da rede pública *Blockchain* para o armazenamento e distribuição de suas informações de aplicativos. Isso se deve ao livre acesso, transparência e auditabilidade presentes nesse tipo de rede, o que é fundamental para os criadores de conteúdo das plataformas de streaming. Aplicando as descobertas ao mercado, foi descoberto que as principais plataformas de streaming de vídeo e música carecem de regras claras para monetizar a mídia produzida pelo público dos criadores de conteúdo. A disponibilidade de recursos computacionais que permitem a transparência, integridade e auditabilidade de regras e informações de geração de receita são recursos necessários em plataformas de streaming que a tecnologia *Blockchain* pode fornecer.

Compreensivelmente, algumas empresas e criadores de conteúdo hesitam em tornar públicos os dados de receita e transação. Como resultado, alguns trabalhos usaram uma rede *Blockchain* privada em suas plataformas de streaming. A rede privada oferece mecanismos de controle que podem ser estabelecidos por seus participantes ou por uma autoridade reguladora para acesso e visualização de informações consideradas sensíveis por seus *stakeholders*. É possível manter as características de transparência, integridade e auditabilidade encontradas nas redes públicas neste caso. Um exame mais aprofundado desse critério revelou que poucos trabalhos descreveram o tipo de rede utilizada na solução desenvolvida. O tipo de rede foi classificado para estes estudos com base nas características e detalhes de operação da solução proposta. A falta de justificativa na rede utilizada pode indicar que os aspectos de governança da rede *Blockchain* e a viabilidade financeira do processamento das transações não foram considerados nos estudos desenvolvidos, o que pode inviabilizar a utilização da solução proposta no mercado.

A abordagem de armazenamento usada em uma solução de computação *Blockchain* tem um impacto direto no poder computacional da rede e no custo financeiro do processamento de cada transação. Poucas soluções nos trabalhos analisados forneceram detalhes sobre a abordagem utilizada para armazenar dados na rede *Blockchain*. A utilização do protocolo *InterPlanetary File System* (IPFS) para armazenar e compartilhar arquivos de áudio e vídeo em um sistema de arquivos distribuído foi identificada neste contexto. No entanto, a indicação de requisitos de

dados armazenados na rede *Blockchain* não foi identificada nas soluções avaliadas. Além disso, não há indicação de como os estudos examinados aqui mediram a audiência de conteúdo de vídeo e/ou áudio e calcularam a geração de receita para criadores de conteúdo.

A falta de informações precisas, fluxos bem definidos e regras pactuadas entre as partes envolvidas justificam o interesse por ferramentas computacionais que tragam mais transparência nas regras de monetização e audiência nas plataformas de streaming. Como resultado, foi possível constatar que a maioria das aplicações analisadas utiliza contratos inteligentes por dois motivos. Garantir a transparência na implementação das regras acordadas entre as partes envolvidas e permitir que os aplicativos realizem pagamentos por meio de carteiras digitais sem o uso de instituições financeiras tradicionais como intermediários.

O uso de uma arquitetura de software bem projetada pode ter um impacto significativo na qualidade e no sucesso de um sistema. As contribuições científicas analisadas no estado da arte trouxeram diversos benefícios com as novas arquiteturas de softwares para plataformas de streaming. O desenvolvimento e gerenciamento de novas camadas em uma arquitetura, a utilização de novas estratégias e requisitos computacionais são responsáveis por resultados, como: avanços na performance de armazenamento das informações, menor latência na transmissão dos conteúdos, plataformas totalmente descentralizadas, controle no gerenciamento de receitas, transparência e auditabilidade nas regras de negócios entre as plataformas e criadores de conteúdo, foram identificados nos trabalhos.

A pesquisa conduzida neste estudo representa um avanço significativo no estado da arte e demonstra a contribuição da tecnologia *Blockchain* na indústria do entretenimento e nas plataformas de streaming. De acordo com os resultados encontrados, o modelo de negócio das plataformas de streaming pode proporcionar maior transparência das regras de negócio entre as partes envolvidas, pagamento sem intermediação de instituições financeiras e rastreabilidade nas métricas de audiência e pagamentos efetuados.

No entanto, percebe-se que as soluções propostas visam criar e manter novas plataformas de streaming, incorporando todas as funcionalidades que compõem o ecossistema do serviço de streaming. Nesse caso, faltam estudos que avaliem a viabilidade financeira e de governança de manter todas as funcionalidades de uma plataforma utilizando a tecnologia *Blockchain*. Como contribuição a este ponto, entende-se que pesquisas atuais devem abordar o desenvolvimento de serviços computacionais que integrem plataformas de streaming no mercado com serviços de audiência, recursos de controle e monetização baseados na tecnologia *Blockchain*. O objetivo principal é manter toda a estrutura de operação das plataformas de streaming como está hoje

e criar serviços intermediários que permitam receber informações das plataformas e gerenciar apenas funcionalidades entendidas como essenciais na rede *Blockchain*.

3.4 CONSIDERAÇÕES FINAIS

Este capítulo investigou como a tecnologia *Blockchain* influenciou e modificou os modelos de negócios das plataformas de streaming. Para atingir este objetivo, investigou-se no estado da arte, pesquisas que abordaram o desenvolvimento e validação de soluções computacionais com tecnologia *Blockchain* levando em consideração os seguintes critérios: i) tipo de rede, ii) descrição do armazenamento de dados, iii) sistema de pagamento baseado na tecnologia *Blockchain*, iv) transparência nas métricas de audiência usando contratos inteligentes, e v) arquiteturas de software utilizadas em plataformas de streaming e *Blockchain*. Os critérios investigados são significativos porque abordam o poder computacional para processamento de transações, pagamento via carteiras digitais e a capacidade de fornecer transparência e rastreabilidade sobre serviços de streaming.

A pesquisa investigada neste estudo representa um avanço significativo na literatura e demonstra a contribuição da tecnologia *Blockchain* no setor de entretenimento e plataformas de streaming. No entanto, notou-se a falta de estudos avaliando a viabilidade financeira e de governança de manter a funcionalidade completa de uma plataforma de streaming usando a tecnologia *Blockchain*.

Além disso, uma lacuna no estado da arte a respeito de serviços computacionais que integrem plataformas de streaming utilizadas no mercado com recursos de audiência, monetização, transparência e auditabilidade baseados na tecnologia *Blockchain*, direcionou o desenvolvimento desta pesquisa. Deste modo, o objetivo desta dissertação é desenvolver uma arquitetura de software para plataformas de Streaming que mantém toda a estrutura operacional existente e a criação de serviços intermediários que permitam o recebimento de informações das plataformas e realize o controle de audiência e monetização baseadas na tecnologia *Blockchain*.

4 ARQUITETURA DE SOFTWARE DA SOLUÇÃO PROPOSTA

A era digital tem trazido uma revolução na forma como as pessoas consomem mídias digitais, com isso os streamings de vídeo e áudio se tornaram uma forma cada vez mais popular de entretenimento. No entanto, as atuais plataformas de streaming não fornecem mecanismos que garantam total transparência sobre as regras de monetização e audiência utilizadas na plataforma, deste modo os criadores de conteúdo enfrentam desafios relacionados à segurança, privacidade e transparência dos dados sobre os seus conteúdos fornecidos ao público, como visto no Capítulo 3. Para resolver a lacuna identificada no Capítulo 3, este trabalho propõe uma arquitetura de software nomeada de StreamingChain, um design de software para plataformas de streaming baseada em tecnologia *Blockchain* que faz uso de um middleware de controle de audiência e pagamentos com *Blockchain* e mantém a estrutura operacional existente na plataforma de streaming.

A arquitetura StreamingChain combina a segurança e a descentralização da *Blockchain* com a escalabilidade e a eficiência já utilizada pelas plataformas, fazendo uso de uma camada de *Blockchain* para disponibilizar maior confiabilidade para os seus criadores de conteúdo. Ela funciona como uma ponte entre as plataformas de streaming e os criadores de conteúdo, fornecendo métricas de audiência precisas e justas que são registradas em contratos inteligentes e podem ser comparadas com as informações disponibilizadas em telas de dashboard das plataformas de streaming.

Este modelo de software proposto permite que os criadores de conteúdo recebam uma remuneração adequada e justa pelo seu trabalho, sem a necessidade de intermediários que possam prejudicar o processo de pagamento. Além disso, StreamingChain também oferece uma camada adicional de segurança aos dados de audiência dos vídeos, prevenindo manipulações ou falsificações dessas informações, o que é fundamental para garantir a precisão das métricas de audiência e para o sucesso dos criadores de conteúdo.

Neste capítulo, é explorada a arquitetura e as vantagens de StreamingChain, bem como suas aplicações para a indústria de streaming. A seguir, a Seção 4.1 descreve um cenário funcional para exemplificar um contexto no qual a proposta desta pesquisa se aplica. A seção 4.2 apresenta os requisitos funcionais e não funcionais identificados na arquitetura, enquanto na seção 4.3 é apresentado em detalhes a arquitetura de software. Por fim, a última seção apresenta as considerações finais deste capítulo.

4.1 CENÁRIO FUNCIONAL

Nesta seção é apresentado um cenário funcional, que serve como base para a identificação dos requisitos considerados na proposta de arquitetura de software discutida neste capítulo. Nesse cenário, foi utilizada a plataforma de streaming vídeo YouTube como ambiente de estudo para o desenvolvimento da arquitetura. O YouTube é uma plataforma de compartilhamento de vídeos que permite aos usuários visualizar, carregar e compartilhar vídeos com a comunidade online. Neste caso, os usuários podem ser criadores de conteúdo e/ou usuários comuns.

A arquitetura do YouTube é uma combinação de tecnologias e sistemas que permite o armazenamento, a transmissão e a exibição de vídeos na plataforma. O YouTube usa uma arquitetura de cluster baseada em servidores para lidar com a grande quantidade de vídeos e tráfego de usuários na plataforma. Os vídeos são armazenados em servidores de vídeo distribuídos geograficamente para garantir alta disponibilidade e velocidade de carregamento. A plataforma também usa tecnologias de cache e transcoding para fornecer aos usuários uma experiência de reprodução suave, mesmo com diferentes larguras de banda de internet e dispositivos. Sistemas avançados de recomendação e pesquisa são empregados pelas plataformas para auxiliar os usuários na descoberta e visualização dos vídeos desejados. Os principais protocolos de comunicação utilizados são o HTTP (Hypertext Transfer Protocol) e HTTPS (Hypertext Transfer Protocol Secure) para a transferência de dados de vídeo e outros conteúdos e o protocolo RTSP (Real-Time Streaming Protocol) é utilizado para transmissões de vídeo em tempo real.

Para melhor entendimento do cenário funcional identificado, serão apresentados dois casos de uso presente nesse modelo de negócio. O primeiro caso de uso será o do criador de conteúdo, no qual realiza o upload do vídeo e recebe pagamentos baseados no seu consumo. O segundo caso consiste do usuário comum, que fornece o tempo de consumo do conteúdo presente na plataforma.

O processo de upload de vídeo começa com o criador de conteúdo que usa o front-end da plataforma de streaming para enviar o arquivo de vídeo para o servidor da plataforma. O front-end, também conhecido como "interface do usuário", é a parte da aplicação que o usuário vê e interage com o sistema. A comunicação é geralmente feita por meio de APIs REST. O protocolo de comunicação usado para essa transferência de arquivos é o protocolo HTTP, que é amplamente utilizado na web para transferência de dados.

Depois de enviar o vídeo, ele é recebido pelo back-end da plataforma de streaming, que é composto por vários servidores e camadas de tecnologia, incluindo bancos de dados, sistemas de

armazenamento de arquivos e aplicativos de codificação de vídeo. O back-end ou "servidor", é a parte da aplicação responsável por gerenciar as requisições do front-end, processá-las e retornar as respostas apropriadas ao usuário. A partir desse ponto, o vídeo é processado e codificado para que possa ser exibido corretamente no site da plataforma.

Os processos de tratamento e configurações realizados com vídeo no back-end não serão abordados nesta pesquisa por não fazerem parte do objetivo da mesma. A arquitetura proposta não modificará o processo de recebimento e entrega dos conteúdos das plataformas.

O sistema de pagamento para os criadores de conteúdo do YouTube é baseado em publicidade. Quando um usuário assiste a um anúncio em um vídeo, o anunciante paga ao YouTube uma taxa, e uma parte desse valor é destinada ao criador do conteúdo. Ao processar pagamentos, a plataforma leva em consideração fatores como a política de monetização, regulamentos fiscais e questões de privacidade, para garantir que os pagamentos sejam realizados.

Uma vez elegíveis, os criadores de conteúdo são pagos pelo YouTube a cada mês. O YouTube usa uma combinação de cookies e tecnologias de reconhecimento de publicidade para determinar quantos anúncios foram exibidos em cada vídeo e, portanto, quanto dinheiro deve ser pago a cada criador. O pagamento é feito através de instituições financeiras, por transferência bancária ou por uma conta em uma empresa parceira.

A visualização de um vídeo no YouTube envolve a comunicação entre o front-end e o back-end da plataforma. O protocolo de comunicação utilizado pode ser o HTTP. Quando um usuário clica em um vídeo para visualizá-lo, o front-end envia uma solicitação ao back-end solicitando informações sobre o vídeo, quando a requisição do usuário chega ao back-end, ele é primeiro direcionado ao serviço de gerenciamento de vídeos. Este serviço verifica se o vídeo está disponível e se o usuário tem permissão para visualizá-lo. Em seguida, o serviço de entrega de conteúdo é acionado para enviar o vídeo ao usuário. O YouTube coleta as informações de consumo do vídeo, tais como: tempo de visualização, número de anúncios assistidos, sistema de avaliação do vídeo, comentários, entre outras métricas. A qualidade do vídeo pode ser ajustada automaticamente para atender às condições da rede do usuário.

A Figura 16 apresenta os principais fluxos de disponibilização de conteúdo e camadas da arquitetura do YouTube envolvidos na visualização de um vídeo. A combinação de protocolos de comunicação, arquitetura de microsserviços e outras tecnologias permite que o YouTube ofereça aos usuários uma experiência de visualização de vídeos suave e eficiente.

Dessa forma, como apresentado na Figura 17, a arquitetura proposta tem como objetivo coletar dados sobre o consumo de vídeo gerados pelos usuários e incorporá-los a uma nova

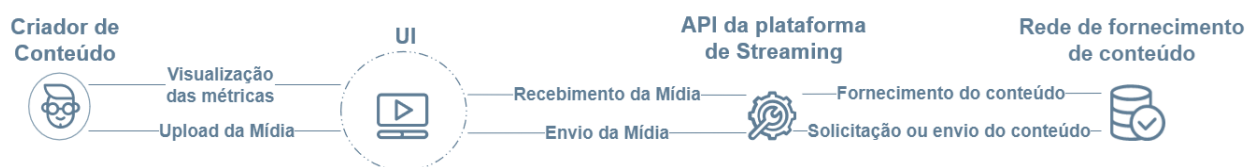


Figura 16 – Principal fluxo de disponibilização de conteúdo da arquitetura do youtube

Fonte: Autor

camada com o uso de *Blockchain*. A partir dos contratos inteligentes, os criadores de conteúdo terão acesso aos dados gerados pelo seu vídeo, bem como às regras estabelecidas para monetização de seu conteúdo.

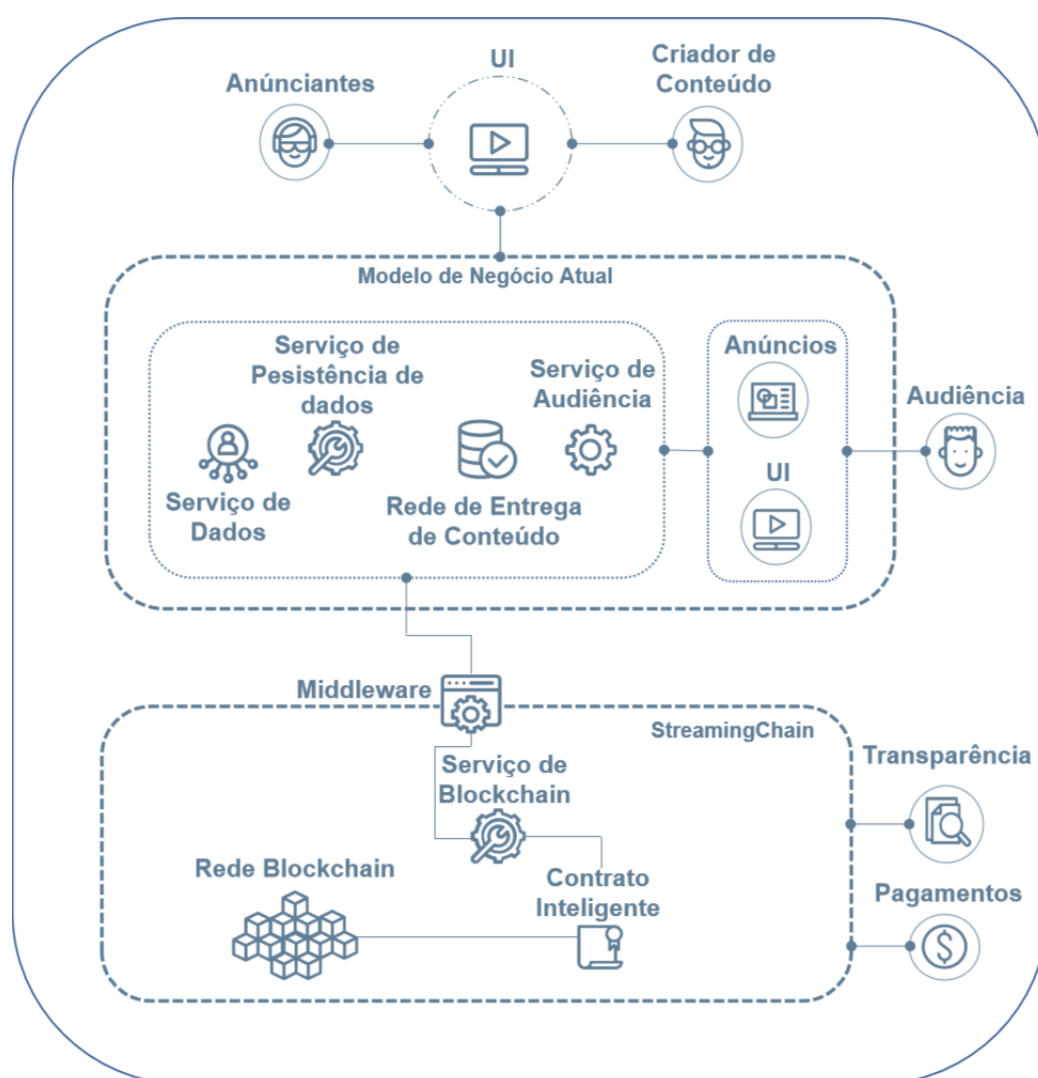


Figura 17 – Arquitetura Proposta

Fonte: Autor

4.2 REQUISITOS DE SOFTWARE

Nesta seção, apresentam-se os requisitos funcionais e não funcionais, que devem ser considerados e atendidos pela arquitetura proposta. Como apresentado na Seção 2.4 requisitos funcionais são aqueles que descrevem o que o sistema deve fazer. Sendo geralmente baseados nas necessidades dos usuários e no contexto do sistema. Já os requisitos não funcionais são aqueles que descrevem como o sistema deve ser, e não o que ele deve fazer. São os requisitos responsáveis por descrever os critérios que o sistema de software deve atender para que os requisitos funcionais sejam satisfeitos.

4.2.1 Requisitos Funcionais e Não Funcionais da Arquitetura StreamingChain

Os Requisitos Funcionais (RF) são descrições de serviços ou de resultados que um sistema de software deve fornecer para atender às necessidades do usuário. Desse modo, para atingir o propósito da arquitetura desenvolvida nesta dissertação, os requisitos funcionais devem prover capacidades que assegurem segurança, transparência e possibilidade de auditoria aos produtores de conteúdo em relação às mídias disponibilizadas nas plataformas de streaming.

Para atender aos requisitos necessários, é fundamental que todas as regras relativas à monetização, à audiência e às informações do conteúdo armazenado na plataforma estejam disponíveis em um mecanismo que assegure a execução das regras estabelecidas entre as partes e a integridade das informações, garantindo ainda que elas não tenham sofrido qualquer tipo de alteração.

Nesta perspectiva, a implementação de contratos inteligentes para assegurar a implementação das regras acordadas entre os envolvidos e proporcionar aos criadores de conteúdo uma compreensão sobre as normas de monetização de seus materiais, bem como a garantia da inalterabilidade das métricas de audiência, podem ser alcançados através da utilização de contratos inteligentes.

Partindo desse pressuposto, foram identificadas funcionalidades que permitem esse controle. Portanto, a solução proposta fornece aos criadores de conteúdo acesso a métricas precisas e imutáveis sobre seus conteúdos, através da utilização da tecnologia *Blockchain*. Além disso, a independência de intermediários financeiros permite aos criadores de conteúdo receber pagamentos de maneira mais eficiente, e a disponibilidade das regras de monetização em smart contracts permite a elaboração de estratégias de marketing eficazes. Adicionalmente, a auditoria completa das ações realizadas em seus conteúdos e a distribuição justa dos pagamentos entre sócios são outros benefícios promovidos por essa solução. Desta forma, os RF identificados para

esta arquitetura foram:

- **RF1:** Gerar contratos inteligentes individuais para cada conteúdo disponibilizado na plataforma. Com isso, no momento em que a mídia é enviada para a plataforma de streaming, um smart contract é gerado com informações do criador de conteúdo, tais como nome de usuário ou nome do canal, além da chave pública da conta do usuário. O contrato também conterá informações da plataforma de streaming, incluindo a chave pública da conta da plataforma, o valor inicial armazenado no contrato para pagamentos futuros e as métricas de audiência, com valores iniciados em zero. A Figura 18 apresenta este RF.

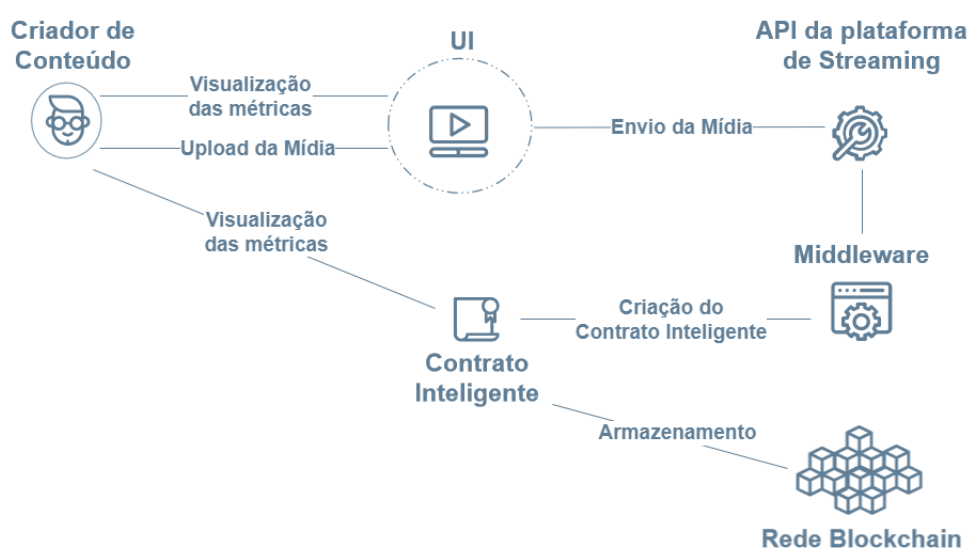


Figura 18 – Fluxo de upload de mídia da plataforma de streaming

Fonte: Autor

- **RF2:** Atualizar as informações do vídeo com base no consumo do conteúdo. Esta funcionalidade é responsável por atualizar as métricas de audiência do conteúdo disponibilizado na plataforma, as quais são baseadas no consumo do usuário final (telespectador). Durante a execução da mídia, informações como tempo assistido, número de interações com o vídeo, avaliações positivas e negativas, número de anúncios assistidos e outras informações relevantes são coletadas e atualizadas.
- **RF3:** Realizar transferências monetárias para os criadores de conteúdo com base na regra de negócio estabelecida. O smart contract da mídia disponibilizada na plataforma será responsável por verificar as condições estabelecidas na regra de negócio, como a quantidade de visualizações ou engajamento do conteúdo, e transferir automaticamente o valor correspondente para a carteira do criador de conteúdo.

- **RF4:** Permitir a visualização das métricas de audiência da mídia. Essa funcionalidade viabilizará o acesso às métricas de audiência de seus conteúdos, bem como acompanhar o desempenho dos mesmos ao longo do tempo. Para isso, o smart contract será acessível através da plataforma de streaming utilizada ou por meio da rede *Blockchain* na qual foi armazenado, oferecendo assim maior comodidade e facilidade de acesso aos usuários.
- **RF5:** Realizar a verificação da origem das ações na plataforma de streaming, garantindo que apenas a plataforma de streaming possa realizar ações específicas, tais como remuneração do criador de conteúdo e atualização das métricas de audiência.
- **RF6:** Realizar a verificação da origem das ações do criador de conteúdo para que somente ele e de pessoas autorizadas possam ter acesso às métricas e regras de negócios presente no contrato.
- **RF7:** Permitir a divisão de pagamentos entre parceiros do canal. A funcionalidade permite aos criadores de conteúdo estabelecer parcerias com outros usuários da plataforma, definindo regras no smart contract para a distribuição do valor monetário gerado pelo conteúdo.

Adicionalmente aos requisitos mencionados, foram identificados dois requisitos funcionais para monitoramento de recursos da aplicação.

- **RF8:** Permitir o monitoramento de recursos de hardware da aplicação. Essa funcionalidade consiste em coletar informações sobre o uso dos recursos de hardware pelos componentes da aplicação em tempo real. Essas informações incluem o uso da CPU, memória e disco. Esse recurso é capaz de enviar alertas em caso de sobrecarga ou falhas em algum dos recursos, permitindo que a equipe de operações possa identificar rapidamente o problema e realizar ações corretivas.
- **RF9:** Permitir o monitoramento de requisições interceptadas pela camada do Streaming-Chain. Essa funcionalidade possibilita a coleta de informações relevantes, tais como: status code da resposta da requisição, latência, rota da requisição, custo em gás¹ do armazenamento na *Blockchain*, quantidades de requisições com sucesso e falha, nome da rede *Blockchain* utilizada para armazenamento. Esses dados serão utilizados para análise de

¹ Uma unidade de gás é uma medida utilizada para determinar a quantidade de trabalho computacional necessária para executar uma transação ou contrato inteligente na rede Ethereum.

desempenho da plataforma, bem como para fins de auditoria, garantindo a transparência e segurança das transações realizadas na rede *Blockchain*.

Diante do exposto, identificamos as principais funcionalidades necessárias para atender aos objetivos propostos na aplicação. Vale ressaltar que, na construção de um sistema de software, os Requisitos Não Funcionais (RNF) também desempenham um papel crucial, definindo as características de qualidade que o software deve possuir, como desempenho, usabilidade, segurança, entre outros. No entanto, neste trabalho, os requisitos não funcionais não foram considerados, tendo em vista que essas considerações não foram estudadas na pesquisa.

4.3 FORMALIZAÇÃO DA ARQUITETURA DE SOFTWARE COM O PADRÃO C4 MODEL

Nesta seção, será apresentada a arquitetura proposta utilizando o modelo C4 para definir os componentes do middleware responsável pelo gerenciamento e transparência dos conteúdos fornecidos para as plataformas de streaming. A escolha do C4 Model como a metodologia de modelagem arquitetural para este projeto foi feita com base em várias considerações fundamentais que o tornaram a opção mais adequada.

Primeiramente, o C4 Model é uma abordagem de modelagem arquitetural que se destaca por sua capacidade de representar de forma clara e concisa a arquitetura de sistemas complexos. Ele oferece um conjunto de diagramas de alto nível que permite a representação hierárquica dos componentes do sistema, desde uma visão geral até os detalhes de implementação. Isso é particularmente relevante para um projeto de desenvolvimento de um software, já que essas aplicações geralmente consistem em uma variedade de componentes interconectados, cada um desempenhando um papel específico.

O C4 Model promove a clareza na comunicação entre os membros da equipe de desenvolvimento e outros *stakeholders* envolvidos no projeto, facilitando o compartilhamento de conhecimento e a compreensão da arquitetura proposta. Sua abordagem simplificada e a ênfase na criação de diagramas legíveis e compreensíveis desempenham um papel fundamental nesse aspecto. Além disso, o C4 Model é altamente flexível e pode ser adaptado para atender às necessidades específicas do projeto, o que se torna crucial quando se lida com uma arquitetura sujeita a evoluções ao longo do tempo, à medida que novos requisitos surgem ou à medida que a plataforma de streaming se expande.

4.3.1 Níveis da Arquitetura

A arquitetura foi especificada com base no levantamento dos requisitos funcionais na Seção 4.2.1. Em seguida, os componentes da arquitetura foram especificados com o propósito de fornecer as capacidades requeridas para atender a esses requisitos de forma eficaz. Para registrar esta arquitetura, adotamos as orientações estabelecidas no modelo C4 apresentado na Seção 2.5. Consequentemente, a arquitetura foi delineada por meio de diagramas que representam quatro camadas distintas: contexto, contêiner, componentes e código. Nas seções subsequentes, forneceremos uma descrição detalhada dessas camadas.

4.3.1.1 Nível de Contexto

Conforme ilustrado na Figura 19, o nível de contexto da arquitetura proposta é composto por dois ambientes (interface gráfica e internet), três tipos de pessoas (criadores de conteúdos, anunciantes e audiência) e três tipos de *softwares* (*front-end*, *back-end* e *middleware*), os quais são descritos a seguir:

- A interface gráfica representa o ambiente no qual os *stakeholders* têm acesso à plataforma de *streaming*;
- O *front-end* da plataforma de *streaming* é o sistema que possui as regras de negócios responsáveis por gerenciar a interface gráfica do usuário e realizar a comunicação com o *back-end* da plataforma de *streaming*;
- Criador de conteúdo é uma pessoa que disponibiliza conteúdo multimídia para ser consumido na plataforma de *streaming*;
- Anunciante é uma pessoa ou empresa que realiza pagamentos para a plataforma de *streaming* exibir seu produto ou serviço em sua aplicação;
- Audiência é a pessoa responsável por consumir o conteúdo disponibilizado pela plataforma de *streaming* e eventualmente consumir os produtos ou serviços anunciados na plataforma;
- Internet é um ambiente virtual acessado por uma rede de computadores;
- O *back-end* da plataforma de *streaming* é um software encarregado de executar a adaptação dos componentes de software, garantindo seu funcionamento eficiente e a entrega de conteúdo de maneira fluída;

- O *Middleware Blockchain* é o *software* encarregado de interceptar as requisições direcionadas ao *back-end* da plataforma de *streaming*, executando comandos nos contratos inteligentes relacionados ao conteúdo acessado no *front-end*.

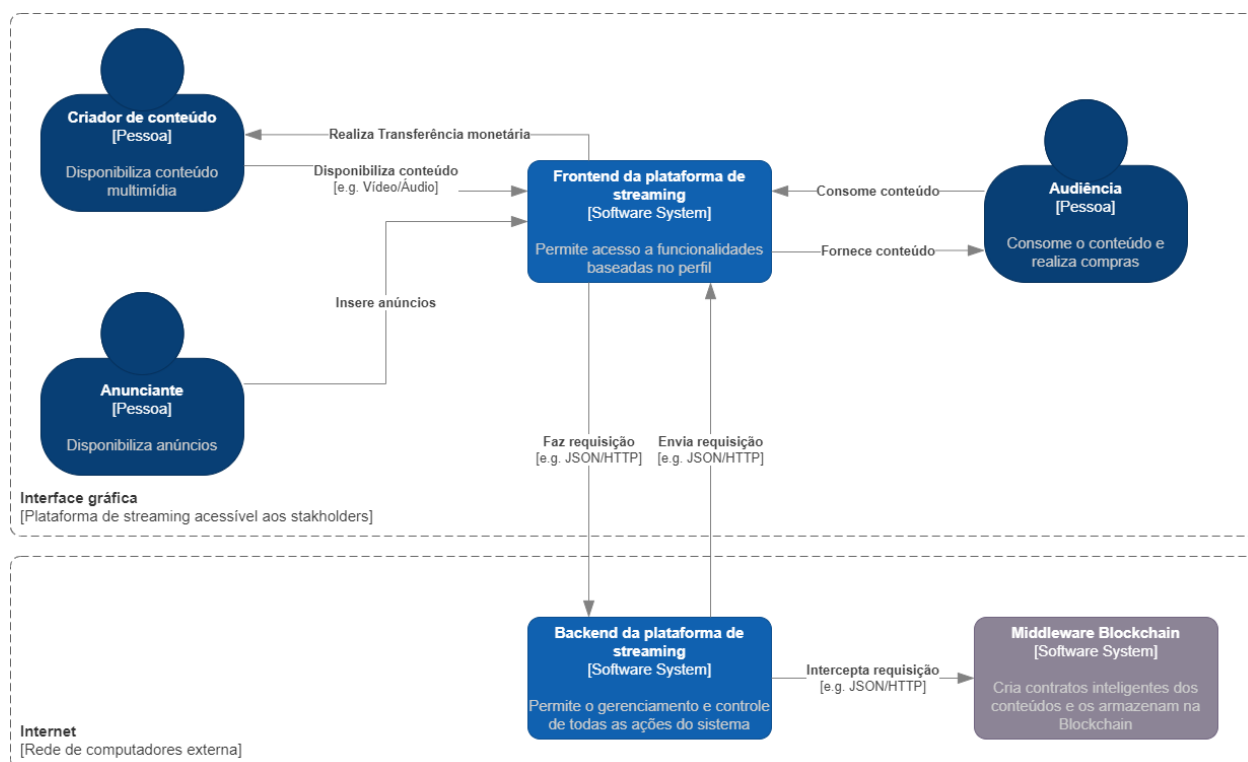


Figura 19 – Nível de contexto da arquitetura proposta: pessoas, software, modelo de dados e suas relações

Fonte: Autor

A relação entre os elementos mencionados se divide em três cenários distintos de uso, são eles: a interação entre o criador de conteúdo e a plataforma de *streaming*, a relação entre a audiência e a plataforma de *streaming*, e o envolvimento do anunciante com a plataforma de *streaming*. No contexto da interação entre o criador de conteúdo e a plataforma de *streaming*, o procedimento se inicia com o criador de conteúdo disponibilizando seu material multimídia no *front-end* da plataforma de *streaming*. Subsequentemente, o *streaming* encaminha as informações fornecidas pelo criador de conteúdo para o *back-end* da plataforma de *streaming*, o qual, por sua vez, procede ao armazenamento do conteúdo disponibilizado em seu banco de dados. Consequentemente, o *middleware* identifica que a solicitação implica na criação de um novo conteúdo, gerando assim um contrato inteligente que associa o conteúdo disponibilizado à conta do criador de conteúdo.

O segundo cenário de uso diz respeito a disponibilização de anúncios relacionados a produtos e/ou serviços oferecidos pelos anunciantes na plataforma de *streaming*. É importante observar que esse contexto de utilização não se enquadra no âmbito da pesquisa, cujo foco principal é a análise da interação entre o criador de conteúdo e a plataforma de *streaming*.

Por fim, destaca-se a relação entre a audiência e a plataforma de *streaming*. A audiência efetua a sua interação por meio do consumo do conteúdo disponibilizado pela plataforma de *streaming*, sendo que as métricas de consumo geradas por essa audiência são interceptadas pelo *front-end* da plataforma de *streaming* e, subsequentemente, encaminhadas ao *back-end* para a realização do tratamento e armazenamento das informações relevantes. Adicionalmente, o *middleware* assume o papel de interceptar e atualizar as métricas relacionadas ao conteúdo multimídia, efetuando as devidas atualizações no contrato inteligente associados a essas métricas.

4.3.1.2 Nível de Contêiner

O nível de *contêiner* da arquitetura proposta proporciona uma abordagem mais aprofundada ao *Middleware Blockchain*. O *Middleware Blockchain* é apresentado por meio da subdivisão em quatro contêineres distintos, a saber: um destinado ao roteamento e análise de requisições, outro designado para as configurações relacionadas ao detentor do conteúdo, um terceiro responsável pelo controle dos contratos e, por último, um *contêiner* dedicado à implementação do contrato inteligente.

- **Roteamento e Análise de Requisições:** Esse contêiner é responsável por analisar as requisições interceptadas, com o intuito de identificar a rota correspondente. Sua implementação é influenciada pela tecnologia utilizada no *middleware* e no *framework*, que neste contexto foi utilizado o Node Js. Este contêiner é composto por dois elementos essenciais: o "*Router Engine*", responsável por direcionar as requisições com base em diversas informações, como URL, métodos HTTP e cabeçalhos, e o "*Request Parser*", encarregado de examinar o conteúdo das requisições para extrair informações pertinentes, tais como o tipo de rota e seus parâmetros;
- **Configurações do dono do conteúdo:** Esse contêiner foi desenvolvido com a responsabilidade de permitir que o proprietário do vídeo estabeleça os parâmetros necessários para transferir seus ganhos para outras contas. Sua implementação é baseada em uma interface de configuração integrada ao *middleware*. Este contêiner consiste principalmente

na "*Configuration Interface*", a qual oferece ao dono do vídeo uma interface para definir as configurações relacionadas à transferência de ganhos;

- **Controlador dos contratos:** É um contêiner central na arquitetura, encarregado de interagir com o contrato inteligente e executar suas funcionalidades essenciais. Ele é composto por vários componentes especializados, cada um desempenhando funções específicas, garantindo uma gestão eficiente das operações relacionadas ao contrato inteligente, como atualizações e busca das informações do contrato.

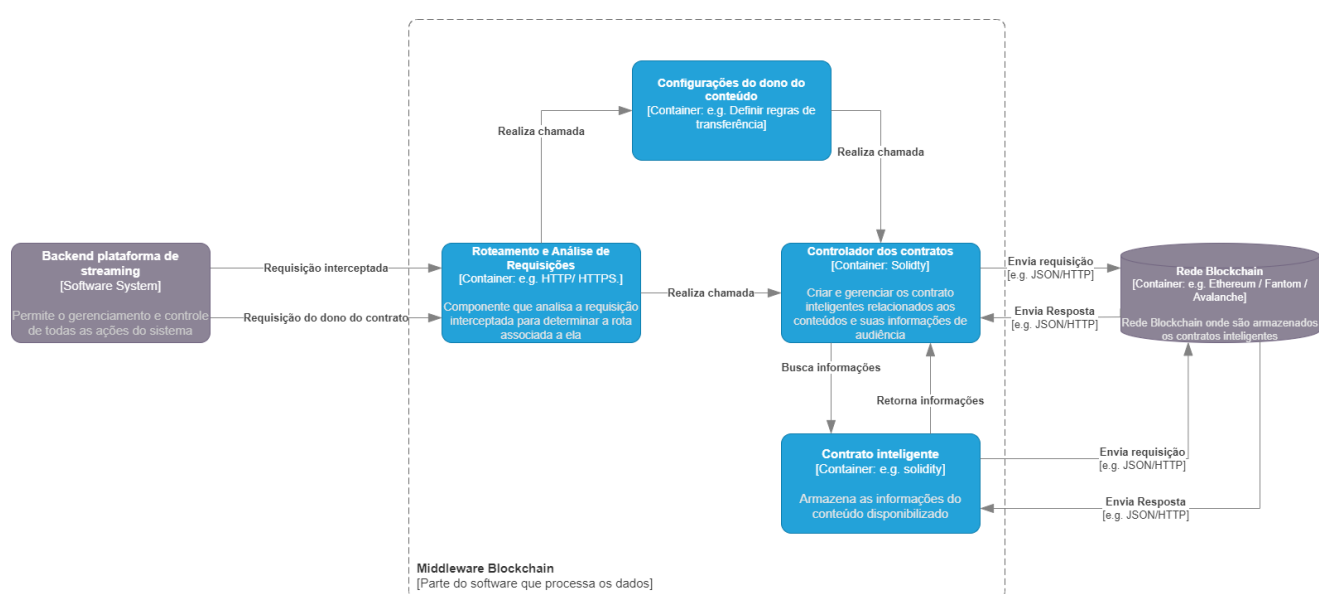


Figura 20 – Nível de contêiner da arquitetura proposta e fluxo de dados entre os contêineres e componentes.

Fonte: Autor

O fluxo de dados ilustrado na Figura 20 descreve um processo que inicia com uma solicitação originada pelo proprietário do contrato ou por uma solicitação interceptada do back-end da plataforma de streaming. Essas solicitações são encaminhadas para o componente de roteamento e análise de requisições, onde são processadas e encaminhadas para o componente de configuração do proprietário do conteúdo ou para o controlador do contrato. O controlador do contrato, por sua vez, possui acesso a todas as funções do contrato e é encarregado de estabelecer conexões com a rede Blockchain. Além disso, o controlador do contrato pode realizar buscas de informações contidas no contrato e ativar suas funções conforme necessário.

4.3.1.3 Nível de *Componente*

Esse nível apresenta com mais detalhes a especificação da arquitetura proposta para o *Middleware Blockchain*, conforme descrito a seguir.

A Figura 21 ilustra a arquitetura da camada de componentes do *Middleware Blockchain* em sua totalidade, porém, a fim de aprimorar a compreensão, sua apresentação será subdividida em três fases distintas: Roteamento, Controladores e Contrato Inteligente, e Rede *Blockchain*.

Roteamento: A Figura 22 apresenta a representação visual dos elementos envolvidos no processo de recebimento e roteamento de requisições. O procedimento de recebimento das requisições é composta pela interação com quatro componentes distintos, definidos a seguir:

- *Ouvinte de Requisição:* O componente em questão assume a responsabilidade de interceptar todas as requisições efetuadas no ambiente onde o *middleware* foi implantado.
- *Engenharia de rotas:* Este componente assume a responsabilidade de receber a URL da requisição e encaminhá-la ao analisador de rotas. Posteriormente, ao receber as informações de roteamento, procede com a chamada do controlador apropriado, encarregado de executar a funcionalidade associada à requisição em questão.
- *Analisador de rotas:* Este componente recebe os parâmetros da requisição e efetua uma validação, verificando a conformidade das informações fornecidas com os critérios especificados na lista de contratos associada. Quando todas as informações estão em conformidade, os dados são encaminhados para a invocação dos controladores correspondentes às rotas em questão.
- *Manipulador de Erros:* Este componente é responsável por gerenciar erros associados a uma funcionalidade específica, aplicando tratamentos adequados de acordo com o tipo de erro identificado. Ele desempenha um papel crucial na detecção e resolução de erros durante a execução da funcionalidade, garantindo uma resposta controlada e apropriada para manter a estabilidade e confiabilidade do sistema.

O fluxo de dados, conforme representado na Figura 22, descreve o processo no qual o componente denominado *ouvinte da requisição* inicialmente intercepta todas as requisições dentro do ambiente no qual o *middleware* está implantado. Subsequentemente, este componente encaminha as informações da URL interceptada à *engenharia de rotas*, cuja responsabilidade é realizar a análise e direcionamento das requisições ao *analisador de rotas*. O analisador de rotas, por sua vez, recebe os parâmetros da requisição e efetua uma validação para determinar se as informações fornecidas são consistentes e válidas. Caso se constate a inviabilidade dessas informações, são direcionadas ao *Manipulador de Erros* para a devida implementação do tratamento de erros correspondente. No entanto, caso as informações sejam consideradas válidas, o analisador de rotas encaminha as informações necessárias de volta à *engenharia de rotas*, a fim de acionar o controlador responsável pela funcionalidade associada à requisição em questão.

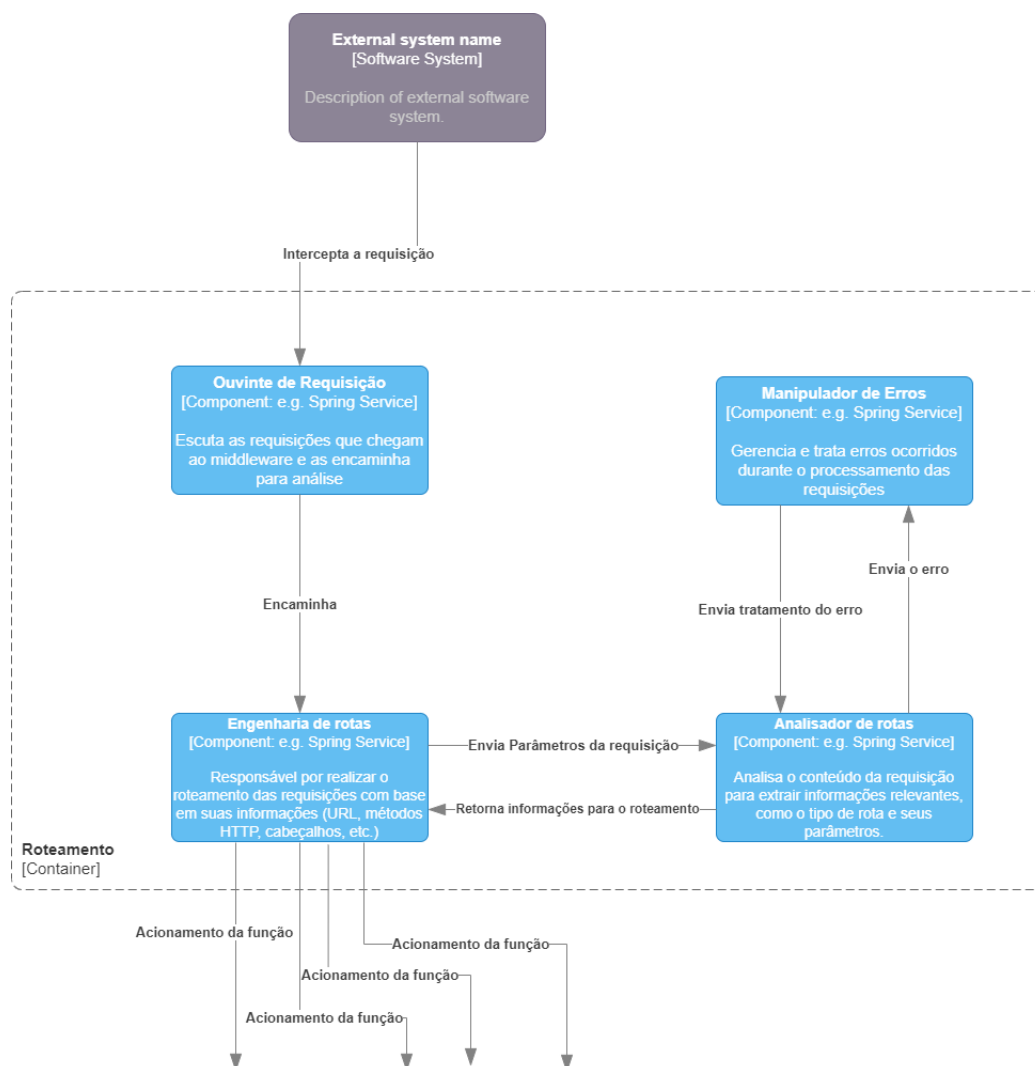


Figura 22 – Nível de componente do *Roteamento* e suas interações.

Fonte: Autor

Controladores: A Figura 23 representa o contêiner denominado *Controladores e Controladores do contrato* que juntos possuem quatro componentes, cuja responsabilidade é estabelecer a comunicação com as funcionalidades inerentes ao contrato de cada vídeo. A definição de cada é definida como:

- *Configuração de transferência monetária:* Este componente detém a responsabilidade de gerenciar os ganhos advindos do conteúdo oferecido na plataforma de streaming. O detentor do conteúdo tem a capacidade de vincular contas de terceiros e especificar a quantia ou a porcentagem que tais contas receberão da receita gerada pelo conteúdo. Posteriormente à recepção dessas informações de transferência, elas são encaminhadas ao componente responsável pela atualização das informações presente no contrato inteligente;

- *Atualiza informações*: Este componente desempenha a função de acionar todas as funções contratuais que efetuam modificações em seu estado atual. Os parâmetros são recebidos, sendo a responsabilidade deste componente a ativação da função apropriada;
- *Buscar informações*: Este componente assume a responsabilidade de ativar todas as funcionalidades de busca de informações presentes no contrato. A ação de busca é iniciada de acordo com o parâmetro recebido;
- *Criar contrato*: Este componente desempenha a função de criar um novo contrato relativo ao conteúdo fornecido. Seu uso é restrito ao momento em que o criador de conteúdo disponibiliza um novo conteúdo em sua conta. Todas as métricas de audiência são inicializadas com valores nulos (zero), enquanto a transferência financeira dos rendimentos se inicia com 100% destinados à conta do criador de conteúdo.

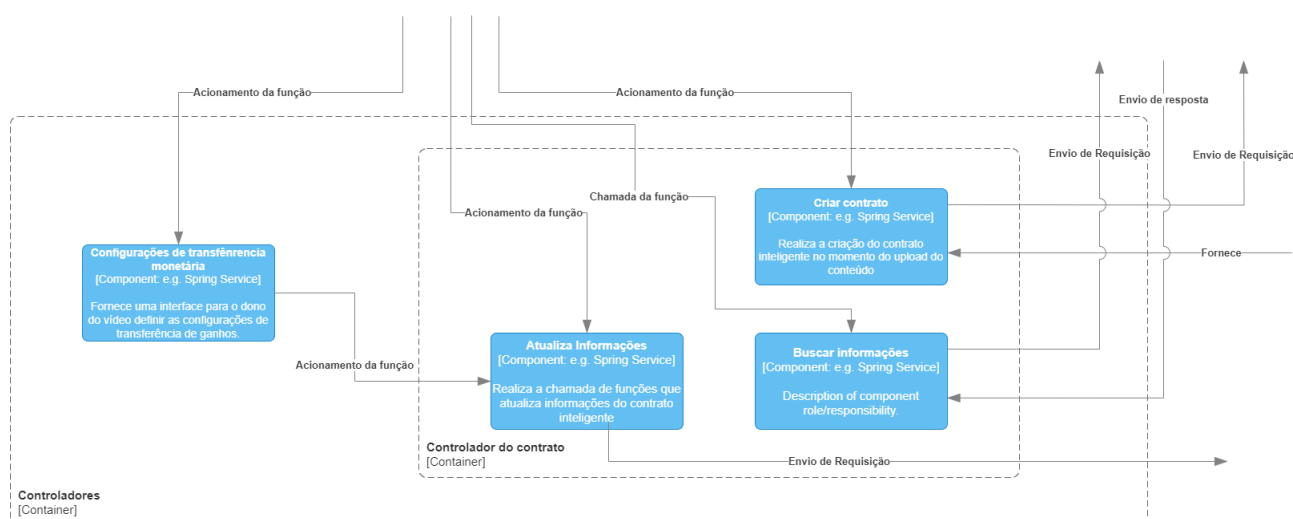


Figura 23 – Nível de componente dos *Controlers* e suas interações

Fonte: Autor

A descrição do fluxo de dados representado na Figura 23 inicia-se com a ativação do componente denominado *Engenharia de Rotas*. Este componente possui a capacidade de transmitir os parâmetros necessários aos quatro componentes envolvidos. Para acionar o componente *Criar Contrato*, a *Engenharia de Rotas* envia os parâmetros requeridos, os quais são recebidos e preparados para aplicação no contrato inteligente quando estiver pronto. Após esse processamento, as informações são encaminhadas à rede *Blockchain* para efetuar o armazenamento do contrato. O processo de ativação do componente *Busca de Informações* segue uma abordagem semelhante à anterior, em que a *Engenharia de Rotas* transmite os parâmetros necessários ao

componente de busca de informações. Após o recebimento e validação dessas informações, uma solicitação é enviada à rede *Blockchain*, que, por sua vez, retorna as informações, caso existam, tornando-as disponíveis ao usuário.

O componente *Configuração de Transferência Monetária* também recebe parâmetros da *Engenharia de Rotas*, submetendo esses dados a um processo de tratamento antes de transferi-los ao componente *Atualiza Informações*. Este último identifica qual funcionalidade deve ser executada e envia uma chamada de função à rede *Blockchain*. Vale destacar que o componente *Atualiza Informações* pode igualmente ser acionado diretamente pela *Engenharia de Rotas*, a qual realiza o envio dos parâmetros necessários para sua ativação.

Contrato inteligente: A Figura 24 representa os componentes presentes em um contrato inteligente, composto por cinco elementos, descritos a seguir:

- *Estrutura do contrato:* Este componente detém a responsabilidade de disponibilizar a estrutura do contrato em formato de *bytecode* e a *Interface Binária da Aplicação* (ABI) ao componente encarregado da criação do contrato;
- *Deposito de valores:* Este componente assume a responsabilidade de efetuar depósitos de valores monetários no contrato, sendo esses recursos destinados ao pagamento das contas registradas no mesmo.
- *Monitoramento de audiência:* Este componente detém a responsabilidade de atualizar as métricas de audiência do contrato, com base nos parâmetros recebidos da rede *Blockchain*;
- *Transferência monetária:* Este componente assume a responsabilidade de emitir as ações de transferência para as contas previamente cadastradas;
- *Gerenciamento de pagamentos:* Este componente verifica as contas registradas para o recebimento de pagamentos e executa a geração das transferências conforme as configurações estabelecidas pelo proprietário do contrato.

A maior parte do fluxo de dados desses componentes é iniciada através da Rede *Blockchain*, a qual é ativada pelos componentes presentes nos *Controladores*. O componente responsável pelo depósito de valores recebe uma requisição da Rede *Blockchain* contendo os valores monetários a serem armazenados no contrato, sendo esta ação executada pela plataforma de *streaming*. Para efetuar a atualização das informações de audiência no contrato, a Rede *Blockchain* emite uma requisição direcionada ao contrato, transmitindo os parâmetros especificados pelo

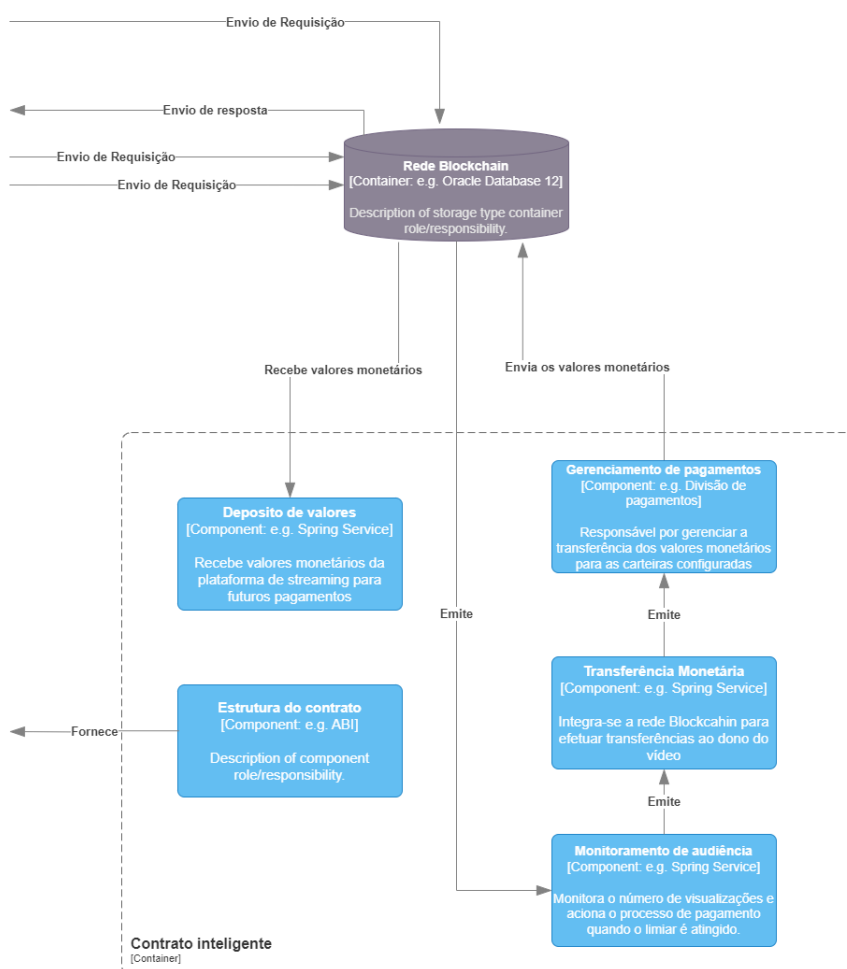


Figura 24 – Nível de componente do *Contrato inteligente* do conteúdo armazenado na plataforma de streaming

Fonte: Autor

Controlador responsável pela chamada. Caso os critérios de pagamento sejam cumpridos no momento da atualização das informações, o componente emite as transferências monetárias para as contas cadastradas. A ação de transferência percorre um último componente encarregado de verificar as contas registradas para o recebimento de pagamentos. Ao término desse processo, uma requisição é enviada do contrato inteligente para a Rede *Blockchain*, contendo os valores a serem transferidos para as contas cadastradas. Por fim, o último componente é a Estrutura de Contrato, que se comunica diretamente com o componente de Criação de Contrato, fornecendo toda a estrutura necessária para a criação de novos contratos inteligentes

4.3.1.4 Nível de *Código*

Na presente dissertação, a escolha feita para representar o nível de código consiste na apresentação de algoritmos e na exposição das implementações dos componentes que constituem

a arquitetura de software proposta

Os algoritmos desenvolvidos como prova de conceito para a construção desta arquitetura estão acessíveis por meio do endereço eletrônico <https://github.com/Rendrikson98/repositorio-streamingChain>. Neste repositório, esses algoritmos são apresentados como um projeto independente na pasta denominada "*middleware*", enquanto o exemplo de caso de uso é disponibilizado na pasta "*back-end*". O contrato inteligente utilizado no âmbito deste estudo encontra-se disponível na pasta "*Smart contract*", e o *front-end* desenvolvido para simular o acesso a uma plataforma de *streaming* pode ser acessado na pasta "*front-end*".

4.4 CONSIDERAÇÕES FINAIS

Neste capítulo, foi proposta uma arquitetura de software para plataformas de streaming que incorpora a tecnologia *Blockchain*, preservando a estrutura de operação existente. Essa arquitetura foi concebida com o objetivo de proporcionar transparência, segurança e autonomia em relação às instituições financeiras para a transferência de monetária. Além disso, oferece recursos de auditabilidade e gerenciamento de mídias disponíveis nas plataformas de streaming através dos registros na rede *Blockchain* e pelo acesso às funcionalidades do smart contract do conteúdo disponibilizado na plataforma.

No início, foram identificados os cenários e requisitos para a arquitetura. O cenário funcional foi apresentado na Seção 4.1, servindo como base para o desenvolvimento dos requisitos funcionais descritos na Seção 4.2.1. A Seção 4.3 serviu para apresentar a arquitetura proposta, documentando os níveis de contexto, contêiner e componentes conforme o modelo C4. Por fim, diversas modalidades de interação entre os elementos arquiteturais projetados foram descritas por meio das subseções contidas na Seção 4.3.

5 AVALIAÇÃO DA SOLUÇÃO PROPOSTA

Este capítulo apresenta as avaliações realizadas na arquitetura de software proposta nesta pesquisa. O principal propósito dessas avaliações é analisar a eficácia da arquitetura proposta em um ambiente de execução simulado, avaliar a opinião dos criadores de conteúdo e desenvolvedores de software com experiência em tecnologia *Blockchain* por meio de formulários eletrônicos e compará-las com outras arquiteturas de software existentes no mercado. Para atingir esses objetivos, foram conduzidos testes computacionais e foi submetido e aprovado pelo Comitê de Ética e Pesquisa o projeto para realização de questionários com criadores de conteúdo, a fim de coletar dados e opiniões sobre a arquitetura proposta e outras existentes no mercado.

Este capítulo é organizado em seções que apresentam avaliações específicas da arquitetura proposta. A Seção 5.1 descreve a avaliação do serviço de geração de transações *Blockchain*. Na Seção 5.2, é apresentada a análise do sistema de monitoramento de aplicações utilizando a tecnologia *Blockchain*. Já a Seção 5.3 aborda uma avaliação que combina aspectos quantitativos e qualitativos referentes ao modelo de negócio envolvendo criadores de conteúdo e desenvolvedores de software. Por último, a conclusão deste capítulo engloba as considerações finais sobre as avaliações conduzidas.

5.1 AVALIAÇÃO DO SERVIÇO DE GERAÇÃO DE TRANSAÇÕES *BLOCKCHAIN*

A avaliação de arquiteturas de software possui muitos desafios, uma vez que envolve a análise de diversos aspectos que influenciam diretamente na qualidade e eficiência do sistema. Para realizar uma avaliação precisa, é necessário definir critérios e métricas que permitam mensurar esses aspectos de forma objetiva, além de estabelecer procedimentos e ferramentas que auxiliem na coleta e análise dos dados. Desse modo, uma alternativa viável para a avaliação da arquitetura de software é a abordagem baseada em cenários. Nessa abordagem, busca-se identificar os requisitos mais críticos e as propriedades de qualidade mais relevantes da arquitetura.

Nesta dissertação, foram empregados dois cenários de avaliação para examinar a efetividade da arquitetura proposta. O primeiro cenário, abordado nesta seção, tem como objetivo avaliar todos os requisitos funcionais mencionados no capítulo anterior. Por sua vez, o segundo cenário tem como propósito avaliar qualitativamente a arquitetura e a solução proposta, mediante a realização de entrevistas com criadores de conteúdos de plataformas de streaming já existentes no mercado.

A fim de validar a solução proposta no primeiro cenário de avaliação, foi desenvolvida uma aplicação de software que tem por objetivo dar suporte ao novo modelo de negócio proposto para as plataformas de streaming. Nesta avaliação, foi simulada uma plataforma de compartilhamento de vídeos, no qual os criadores de conteúdos disponibilizam mídias digitais para consumo dos usuários finais. Neste cenário, nos interessa investigar os seguintes itens de pesquisa: i) a integração dos metadados das plataformas de streaming com a solução proposta, ii) a transparência das regras de negócio dos serviços de streaming por meio de *smart contracts*, iii) o custo financeiro para processar as transações na rede *Blockchain*.

5.1.1 Infraestrutura

Com base no cenário de compartilhamento de vídeo em plataformas de streaming, criou-se um ambiente computacional para avaliar as principais funcionalidades da StreamingChain. A respeito da infraestrutura computacional utilizada, foi empregada uma estação de trabalho em uma rede local com a seguinte configuração: processador I5 7400U 2.5 GHz quad-core, 8 GB de RAM DDR4, sistema operacional Windows 10 Pro e memória ROM SSD 500GB.

Sobre as ferramentas computacionais, foram instaladas na estação de trabalho local o seguinte conjunto de ferramentas: Gerenciador de pacotes *NPM*, *Truffle Suite*, *Ganache*, *Visual Studio Code* e *Insomnia*. Essas ferramentas foram escolhidas porque permitem o desenvolvimento de software com *Blockchain*. Em relação às linguagens de programação e tecnologias empregadas no desenvolvimento da solução proposta, incluem: JavaScript, TypeScript, Node Js, React Js, Knex, Postgresql, Solidity, Jest e GIT.

Cada tecnologia e linguagem de programação utilizada na solução proposta tiveram um papel específico no desenvolvimento do projeto. O JavaScript foi utilizado como a principal linguagem de programação para desenvolvimento front-end e back-end, enquanto o TypeScript foi adotado exclusivamente para fornecer tipagem estática e aumentar a segurança e a escalabilidade do código. O Node Js foi utilizado como o ambiente de tempo de execução para o back-end, permitindo a execução de código JavaScript fora do navegador. O React Js foi empregado como biblioteca JavaScript para desenvolvimento de interfaces de usuário. O Knex foi escolhido como uma ferramenta para construção de consultas SQL e migração de banco de dados. O Postgresql foi utilizado como o sistema de gerenciamento de banco de dados relacional para simular o armazenamento convencional em ferramentas do mercado. O Solidity foi empregado como linguagem de programação para a criação de *smart contracts* na rede *Blockchain*. O Jest foi utilizado como framework de testes unitários. O *GIT* foi escolhido como sistema de controle de

versão para gerenciar o código-fonte do projeto.

5.1.2 Primeira Etapa: Construção do Smart Contract e Middleware

A primeira etapa do desenvolvimento do middleware consistiu na criação do *Smart Contract*, que tem como principal objetivo receber todas as regras de negócio relacionadas à monetização das mídias disponibilizadas pelos criadores de conteúdo na plataforma de streaming. Para isso, foram inseridas no contrato informações essenciais como nome e descrição da mídia armazenada, nome do usuário, nome do canal, chave pública da carteira do criador de conteúdo, da plataforma de streaming e de terceiros, caso haja divisão de lucro. As principais funcionalidades presentes no *Smart Contract* incluem a atualização das métricas de audiência sempre que um conteúdo for consumido, o pagamento ao criador de conteúdo quando o número de visualizações atingir o valor definido, a realização do pagamento para cada parceiro conforme estabelecido pelo proprietário da mídia, a adição de valores monetários ao contrato pela plataforma de streaming, funcionalidades de segurança para garantir que apenas a plataforma de streaming ou o proprietário do conteúdo armazenado possam chamar as ações do contrato, e a visualização das métricas dos conteúdos. A Figura 25 mostra as principais funcionalidades do *Smart Contract*. Cada uma dessas funcionalidades foi desenvolvida utilizando a linguagem de programação Solidity.

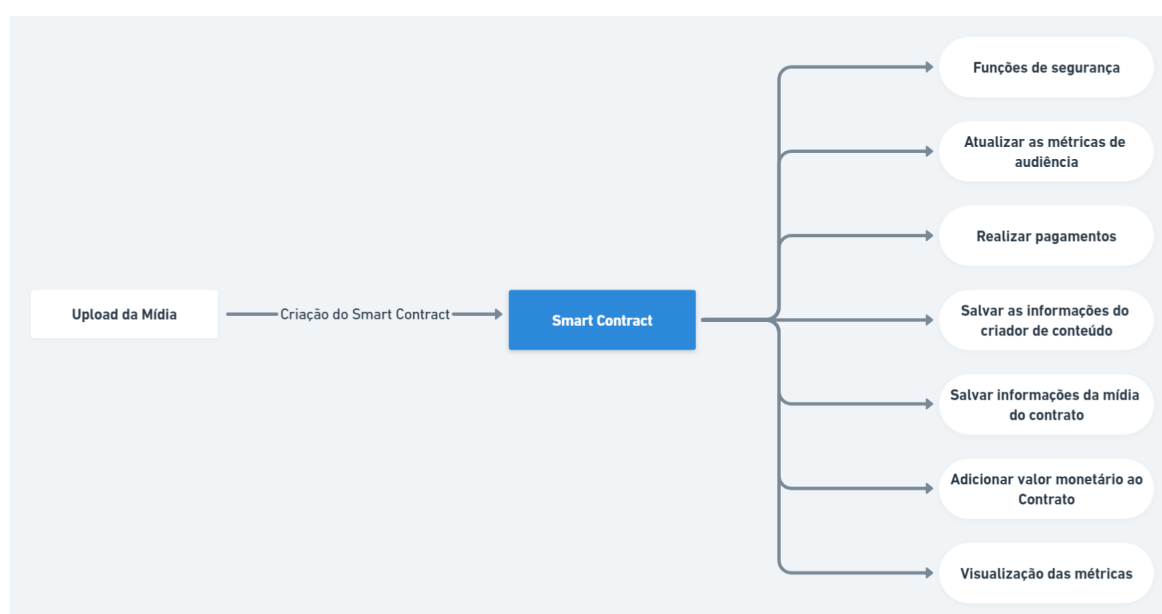


Figura 25 – Principais Funções do Smart Contract

Fonte: Autor

Após a criação da estrutura do *Smart Contract*, deu-se início ao desenvolvimento do

middleware, o qual tem como principal responsabilidade interceptar as requisições e gerenciar as informações de acordo com as funcionalidades presentes no contrato. A escolha das tecnologias utilizadas no desenvolvimento do middleware foi o TypeScript e o Node.js. Com a implementação do middleware, foi possível criar um novo *smart contract* exclusivamente para cada mídia enviada para a plataforma de streaming, contendo todas as regras de negócio e de monetização. Dessa forma, sempre que um telespectador consome uma mídia, as informações de consumo são coletadas e enviadas para o contrato. O contrato, por sua vez, realiza o pagamento das informações quando o número estabelecido de visualizações é atingido. Além disso, é possível visualizar as informações do contrato através da passagem do endereço do contrato no endpoint de visualização presente no middleware.

5.1.3 Segunda Etapa: Construção do protótipo de plataforma de streaming

Para avaliar a arquitetura proposta, foi preciso criar um protótipo de plataforma de streaming que simule as principais ações realizadas pelos usuários e criadores de conteúdo. Essa simulação permitiu a implementação do middleware mencionado na seção anterior no back-end da aplicação.

O back-end do protótipo da plataforma de streaming foi desenvolvido utilizando tecnologias como Typescript, Node.js, Knex e Postgresql. O objetivo principal do back-end é fornecer um conjunto de funcionalidades para que os usuários possam interagir com a plataforma. O sistema de login para criadores de conteúdo permite que eles acessem a plataforma de streaming e gerenciem suas contas e canais. Além disso, foi implementado um sistema de autenticação baseado em tokens para garantir a segurança dos dados dos usuários.

O sistema de upload de mídia permite que os criadores de conteúdo adicionem seus vídeos e outros arquivos de mídia à plataforma, tornando-os disponíveis para os usuários. O sistema de controle de visualização é responsável por rastrear as visualizações dos usuários e manter um registro de todas as atividades realizadas na plataforma.

Vale ressaltar que, devido à natureza do protótipo, somente as principais funcionalidades do back-end foram desenvolvidas para a avaliação do middleware. No entanto, o sistema é facilmente escalável e pode ser expandido com novas funcionalidades no futuro. O uso do Typescript e Node.js permite que o desenvolvimento seja ágil e escalável, enquanto o Knex e o Postgresql são utilizados para gerenciar o banco de dados de forma eficiente e confiável.

Por fim, foi implementado um protótipo com interface gráfica de usuário (GUI) utilizando as tecnologias React e TypeScript, que foi integrado ao backend. Embora o protótipo tenha

sido integrado ao backend, a avaliação de desempenho e custo se concentrou apenas na API. O protótipo de GUI foi criado apenas para fins de demonstração aos criadores de conteúdo entrevistados. Isso permitiu que eles pudessem ver que o uso da plataforma de streaming permaneceu inalterado, apesar da adoção da nova arquitetura.

5.1.4 Avaliação em diferentes redes *Blockchain*

Para os testes realizados nesta subseção, foi utilizada a rede privada do Ganache, e as redes públicas da Fantom e Avalanche. O Ganache é uma ferramenta de *Blockchain* de código aberto que permite criar ambientes locais de *Blockchain* para emulação e teste. Já a Fantom é uma rede pública de *Blockchain* com alta velocidade de transação e baixo custo, enquanto a Avalanche é uma plataforma de contratos inteligentes com alta escalabilidade e interoperabilidade. Para a execução dos testes, a plataforma de streaming e as contas dos criadores de conteúdo receberam uma carteira digital nas três redes de testes. StreamingChain, o sistema desenvolvido, usa *smart contracts* escritos em *Solidity* para registrar informações e termos acordados entre as partes, além de gerenciar transações de dados e pagamentos por visualizações de vídeos. A tecnologia *Blockchain* fornece transparência, permitindo que os *stakeholders* envolvidos no modelo de negócio consultem as transações realizadas e confirmem se estão em conformidade com o acordo estabelecido.

Para avaliar a funcionalidade de StreamingChain com as três redes *Blockchain*, foram inseridos nove vídeos por meio da aplicação web desenvolvida, e em um deles, foi simulado o número de dez mil visualizações. As regras de negócio especificadas nos *smart contracts* garantem que os termos acordados entre as partes envolvidas sejam cumpridos. Para isso, as informações que caracterizam as partes envolvidas são registradas no *smart contract*, e em seguida, os termos e cláusulas são codificados em funções de programação. Com base no número de visualizações, tempo médio assistido por vídeo e tempo total de visualizações, as transações são gerenciadas e os pagamentos são efetuados. Para a avaliação dos pagamentos realizados por meio dos *smart contracts*, consideramos o número de quinhentas visualizações por vídeo para que o criador de conteúdo receba o valor de um criptativo (Ether, FTM ou AVAX) referente à rede em execução. Com o vídeo em questão atingindo o total de dez mil visualizações, o criador de conteúdo recebeu vinte pagamentos em sua carteira digital. A Figura 26 mostra o vídeo objeto de estudo desta avaliação e nela é possível identificar alguns dos componentes de software importantes para o funcionamento de StreamingChain. O *smart contract* desenvolvido na linguagem *Solidity* é ilustrado na Figura 26, juntamente com os endereços das carteiras

digitais do criador de conteúdo e da plataforma de streaming.

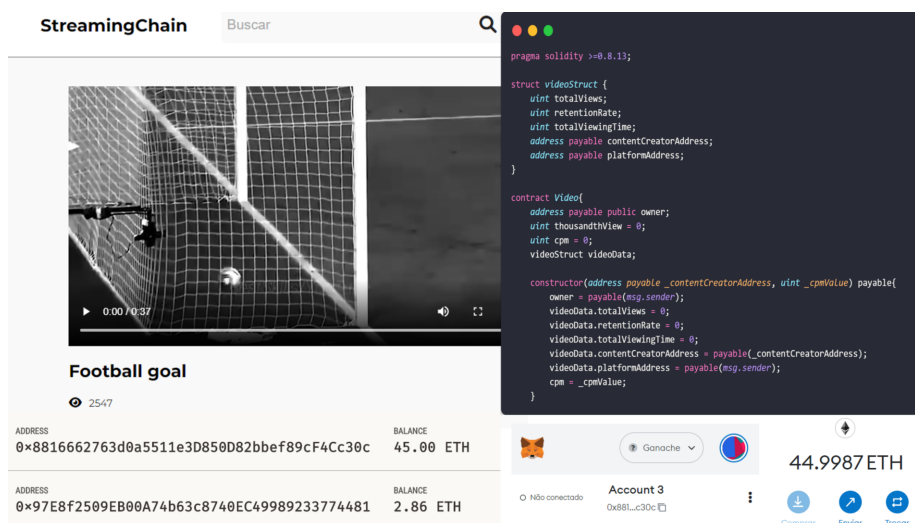


Figura 26 – Plataforma de Streaming baseada em Tecnologia *Blockchain*

Fonte: Autor

Durante os testes de funcionalidades, também foram realizados testes de custo para execução e armazenamento das informações na *Blockchain*. Foi utilizado um *script* que simulava o acesso ao conteúdo, enviando requisições randômicas com métricas de consumo para a API do backend da aplicação. Esse script enviou várias requisições por segundo para a API e registrou o número de requisições bem-sucedidas e com falha durante o processo. Além disso, foram avaliados o custo total das 10 mil transações e o custo médio por transação.

Em meio ao processo de avaliação, foi observado um custo maior por transação após a transação de pagamento ao criador de conteúdo. No entanto, observou-se que na rede Avalanche, não era possível obter informações detalhadas sobre os custos por tipo de transação. Desse modo, somente as redes Fantom e Ganache tiveram avaliações de custo por tipo de transação. Assim, foram coletados os custos médios de transação para atualização de métricas de audiência, realização do pagamento ao criador de conteúdo e a primeira transação após o pagamento.

Quadro 1 – Custo por tipo de transação

Custo por Transação			
Redes	Atualização de Métricas de Audiência	Pagamento ao Criador de Conteúdo	Após o Pagamentos
Ganache	US\$ 1,6947	US\$ 2,2800	US\$4,1998
Fantom	US\$ 0,0010	US\$ 0,0009	US\$ 0,0013

Fonte: Autor

O Quadro 1 apresenta os resultados das avaliações de custo por transação das duas principais funcionalidades do contrato mencionado anteriormente, bem como da transação seguinte ao pagamento do criador de conteúdo, considerando as redes Ganache e Fantom, que possuem criptomoedas distintas. Na rede Ganache, durante a avaliação das métricas mencionadas, o preço pago por unidade de gás para o processamento das transações foi de 20 gwei, e o valor do ETH em dólares foi de US\$1951,00. Vale lembrar que 1 gwei equivale a 1 bilhão de unidades de gás na Ethereum, e que o preço do gwei em dólares foi de \$0,00003902. O custo máximo de processamento das transações com transferência monetária foi de US\$2,28, enquanto que para a requisição seguinte, o maior custo foi de US\$ 4,19. Já o custo médio por transação de atualização de métricas de audiência foi de US\$ 1,69.

Por outro lado, na rede Fantom, a criptomoeda utilizada foi o FTM, cuja cotação no mercado financeiro era de US\$ 0,37. O custo máximo de processamento das transações com transferência monetária foi de US\$ 0,00099, enquanto que para a requisição seguinte, o maior custo foi de US\$ 0,0013. Por fim, o custo médio por transação de atualização de métricas de audiência foi de US\$ 0,0010.

O resultado final da avaliação financeira com a carga de dados de 10 mil transações nas redes Ganache, Fantom e Avalanche é exposto no Quadro 2. Verificou-se que o custo total da carga de dados na rede Ganache foi de US\$ 33.941,51, enquanto a rede Fantom apresentou um custo de US\$ 5,18, com todas as requisições realizadas com sucesso, tendo um custo médio de US\$ 0,000518 por transação. Já a rede Avalanche teve um custo de US\$ 129,72, com um custo médio de US\$ 0,012 por transação.

Quadro 2 – Custo por 10 Mil Transações em Diferentes Redes *Blockchain*

Custo por 10 Mil Transação		
Redes	Custo das 10 mil Transações	Custo Médio por Transação
<i>Ganache</i>	US\$ 33.941,51	US\$ 3,39
<i>Fantom</i>	US\$ 5,18	US\$ 0,000518
<i>Avalanche</i>	US\$ 129,72	US\$ 0,012

Fonte: Autor

Vale ressaltar que, das 10 mil transações realizadas, tanto na rede Ganache quanto na Avalanche, todas as requisições foram executadas com sucesso, ao passo que na rede Fantom, 99% das requisições tiveram êxito e 1% das transações apresentaram falha. Esses resultados revelam aspectos relevantes acerca da eficiência e do desempenho dessas três redes *Blockchain*,

os quais podem ser úteis para futuras avaliações e implementações de soluções em ambiente de *Blockchain*.

Após a análise do custo financeiro para o processamento das transações na rede Ganache, Fantom e Avalanche, é possível concluir que os resultados obtidos não inviabilizam a utilização do StreamingChain como solução para o gerenciamento da audiência e controle de pagamentos em plataformas de streaming. É importante ressaltar que a retirada das taxas de instituições financeiras e a transparência proporcionada pela solução, podem trazer redução significativa nos custos operacionais e aumento da confiança dos stakeholders envolvidos no ecossistema.

Dessa forma, o custo encontrado pode ser incorporado ao modelo de negócio proposto como forma de tornar transparente o controle da audiência e dos pagamentos entre as partes envolvidas. Consequentemente, a solução proposta pode trazer benefícios significativos para as plataformas de streaming e seus usuários, por meio de um modelo de negócio mais eficiente e transparente.

5.2 AVALIAÇÃO DE MONITORAMENTO DE RECURSOS DA APLICAÇÃO

O propósito desta seção é apresentar uma avaliação do sistema de monitoramento de recursos de aplicações *Blockchain*, o qual engloba o monitoramento contínuo de recursos de hardware e requisições da aplicação, com o intuito de produzir relatórios analíticos dessas informações. O sistema objetiva garantir a correta operação da aplicação, bem como aprimorar o uso de recursos de hardware, reduzindo despesas financeiras. Para atingir esse objetivo, foi desenvolvida uma ferramenta de monitoramento capaz de coletar métricas da aplicação e das transações *Blockchain*, além de oferecer interfaces de conexão com diferentes redes *Blockchain*.

A ferramenta em questão foi nomeada como MonitorChain e consiste em um sistema computacional que realiza o monitoramento de recursos de hardware e a avaliação de solicitações de processamento de transações em redes *Blockchain*. O monitoramento de recursos de hardware calcula o uso de RAM (em inglês, *Random Access Memory*), a pilha de processamento e as métricas de uso do processador do recurso de computação executando o sistema de avaliação proposto. Os valores descobertos para as métricas acima são atualizados a cada segundo e os resultados são gravados em um arquivo de log. O componente de monitoramento de solicitação cliente-servidor pode avaliar transações HTTP ou HTTPS. Como resultado do monitoramento da solicitação, as seguintes informações são medidas e gravadas em um arquivo de log: o endereço da rota usada na solicitação, os códigos de status da resposta (HTTP ou HTTPS), a quantidade de gás utilizado em transações *Blockchain*, custo total estimado em dólar, a cotação da criptomoeda

em dólar utilizada nas transações, a criptomoeda utilizada, o preço do gás, o tempo de resposta em milissegundos e o número total de solicitações bem-sucedidas e com falha.

Para permitir a conexão com diferentes redes *Blockchain*, foi desenvolvido um componente que gerencia as interfaces de comunicação com as principais plataformas DApps disponíveis no mercado. As interfaces disponíveis no MonitorChain possuem parâmetros de conexão, assinaturas de método de persistência e contratos de assinatura para classes de aplicativos de software da camada de negócios que precisam se comunicar com diferentes redes *Blockchain*. MonitorChain possui interfaces de comunicação com as redes Fantom, Avalanche e Ethereum, e se necessário novas interfaces de comunicação com outras redes podem ser desenvolvidas seguindo o padrão implementado na arquitetura proposta. Por fim, o componente de carga de trabalho permite a configuração da carga de dados que será processada nas redes *Blockchain* disponíveis no MonitorChain.

Para realizar a avaliação da solução proposta, foi realizada a implementação do MonitorChain no protótipo de plataforma de streaming mencionado na seção 5.1.3. Essa avaliação foi conduzida em um ambiente computacional com uma rede local, composta por estações de trabalho equipadas com 16 GB de RAM DDR4 e processador I7 1165G7. Após a configuração do ambiente computacional, as seguintes etapas foram executadas: i) A rede Ethereum, executada localmente por meio do Ganache, foi selecionada como a rede *Blockchain* de conexão. ii) Foram configuradas três diferentes cargas de dados, totalizando 10.000 transações. A escolha da rede Ethereum foi motivada pela disponibilidade de contas com quantidades significativas de criptoativos para realizar as transações. As cargas de dados foram distribuídas da seguinte maneira: i) 100 transações para a criação de contratos. ii) 6.000 transações para atualizações de informações de contrato. iii) 3.900 transações para consultas de informações de contrato.

Como resultado da avaliação, foi possível monitorar os recursos consumidos durante a execução das cargas de dados por meio da interface gráfica do usuário disponibilizada pelo MonitorChain. A Figura 27 exibe informações sobre as solicitações feitas após a conclusão da avaliação. Esses dados incluem o número total de solicitações, o número de solicitações bem sucedidas e com falha e a quantidade de gás usada para todas as transações, o custo total e a criptomoeda utilizada. Um gráfico também mostra o tempo de execução das últimas dez transações realizadas. Por outro lado, a Figura 28 mostra informações sobre os recursos de hardware utilizados durante o processo de avaliação. Essas informações incluem o uso médio do processador, uso de RAM e pilha de processamento. Os gráficos ilustram o uso médio da capacidade do processador e a quantidade média de RAM usada nos últimos 10 segundos.

entre abertas e fechadas. As perguntas abertas buscaram uma compreensão aprofundada das opiniões dos criadores, enquanto as perguntas fechadas utilizaram a escala de Likert, oferecendo métricas sobre as percepções em relação às normas estabelecidas pela plataforma de streaming. A escala de Likert apresenta alternativas de resposta, a saber: i) discordo totalmente, ii) discordo, iii) não discordo nem concordo, iv) concordo e v) concordo totalmente.

A segunda seção do formulário foi direcionada à coleta de percepções sobre a proposta de arquitetura apresentada. Para tal fim, três questões foram formuladas, abrangendo tanto abordagens fechadas quanto abertas. Essas questões exploraram a arquitetura de software proposta, previamente apresentada em um vídeo de três minutos incorporado ao formulário. O objetivo dessa seção foi colher feedback sobre a solução proposta, enfocando a perspectiva de um público não técnico que utiliza a aplicação como ferramenta de trabalho no cotidiano.

Simultaneamente, o segundo questionário, destinado a desenvolvedores de software com expertise em *Blockchain*, concentrou-se na avaliação da viabilidade técnica do modelo de negócio proposto. Com o intuito de alcançar tal objetivo, elaboraram-se seis perguntas, compreendendo tanto questões fechadas quanto abertas, seguindo a mesma estrutura adotada no questionário destinado aos criadores de conteúdo. Essas perguntas foram direcionadas para explorar aspectos como viabilidade, transparência, riscos, eficiência e integração com as plataformas de streaming já existentes.

Para a execução dessas tarefas, seguiu-se o protocolo estabelecido pelo projeto de número 0170922.3.0000.5013, com aprovação do Comitê de Ética e Pesquisa (CEP). Essa aprovação reforçou a integridade e os padrões éticos na coleta de dados, consolidando a credibilidade da pesquisa e garantindo a confiabilidade dos resultados obtidos.

A partir dessa metodologia adotada, foi possível realizar a coleta e análise das respostas em ambos os questionários. Para proporcionar uma compreensão mais clara e organizada dos resultados, estes serão discutidos em subseções separadas. As metodologias complementares serão examinadas minuciosamente, buscando uma compreensão abrangente do impacto do modelo de negócio proposto, incorporando tanto as nuances subjetivas quanto os dados quantificáveis.

5.3.1 Pesquisa com criadores de conteúdo

Esta subseção encontra-se subdividida nas seções referentes a questões quantitativas e qualitativas. Essa segmentação na análise e interpretação das respostas foi efetuada com o propósito de aprimorar a compreensão dos resultados obtidos.

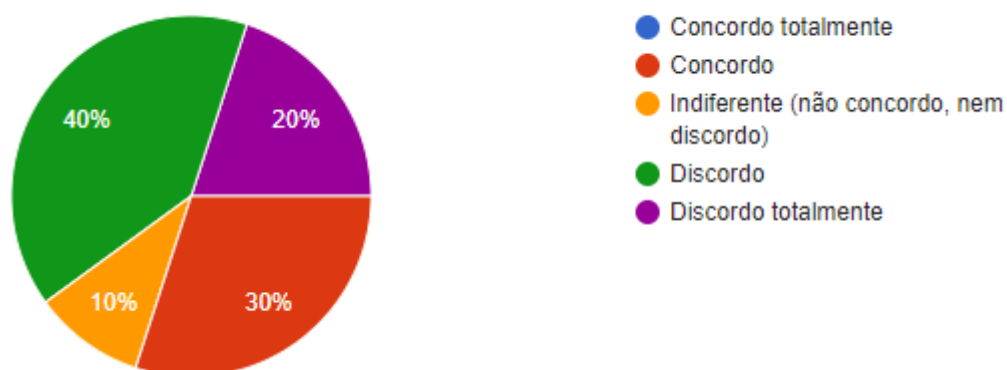


Figura 29 – Clareza das regras de monetização

5.3.1.1 Pesquisa quantitativa

Conforme previamente mencionado, o questionário destinado aos criadores de conteúdo consistia em seis questões quantitativas que empregavam a escala de Likert. Todas as questões desta categoria obtiveram um total de 10 respostas. A análise dos resultados sobre as regras de monetização do YouTube (Figura 29) revelou que 20% dos criadores de conteúdo expressaram plena convicção de que as diretrizes fornecidas não são suficientemente claras para compreenderem o processo de monetização e os critérios necessários para alcançar ganhos financeiros com seus conteúdos. Além disso, 40% dos participantes também discordam, em menor grau, que essas informações são totalmente claras, enquanto 10% indicaram uma posição neutra, sem concordar nem discordar. Por fim, 30% dos participantes afirmaram acreditar que as regras fornecidas são suficientemente claras.

Em continuidade à investigação, os participantes foram indagados acerca da disponibilidade de documentação que concretize o processo de cálculo de monetização em seus respectivos canais. Como resultado, observou-se que 20% dos participantes discordaram integralmente dessa proposição. Em uma perspectiva semelhante, 60% dos criadores de conteúdo manifestaram discordância quanto à disponibilidade de qualquer documento que apresentasse detalhes sobre o referido processo. Adicionalmente, 20% dos participantes adotaram uma posição neutra, indicando que não concordam nem discordam. Notavelmente, nenhuma resposta foi registrada nas categorias de concordância ou concordância total em relação à existência de algum tipo de documentação por parte da plataforma que abrangesse as características mencionadas. A Figura 30 apresenta os resultados mencionados de forma gráfica.

Com o intuito de aprofundar a compreensão das normas de remuneração de conteúdo, foi

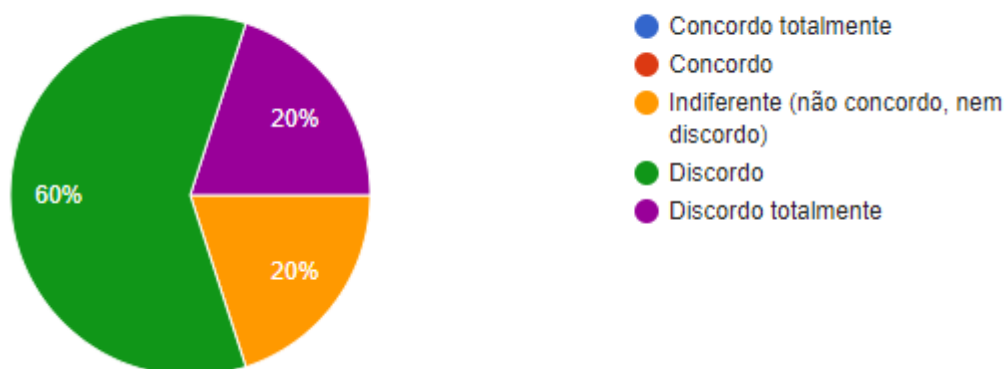


Figura 30 – Fornecimento de documentação sobre o cálculo da monetização

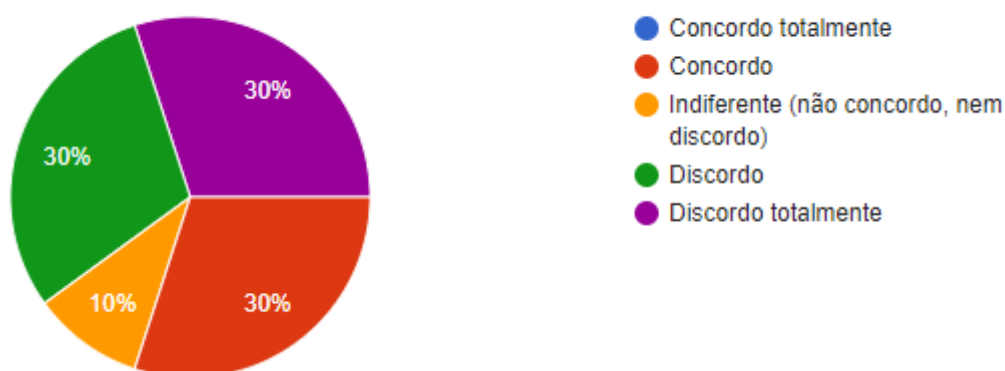


Figura 31 – Pagamentos diretamente ligados ao número específico de visualizações

indagado aos criadores de conteúdo se os pagamentos são efetuados com base em um número específico de visualizações. Nesse contexto, observou-se que 30% dos participantes concordaram com a existência dessa regra para a realização dos pagamentos, enquanto 10% adotaram uma posição neutra. Adicionalmente, 30% expressaram discordância em relação à ideia de que os pagamentos são efetuados com base em um número específico de visualizações, e outros 30% discordaram integralmente dessa regra. Cabe ressaltar que não foram registradas respostas na categoria de concordância total nesse questionamento. Para proporcionar uma visualização mais clara dos resultados obtidos, a Figura 31 apresenta uma representação gráfica dessas respostas.

Com o propósito de avaliar a acessibilidade das regras de restrições de conteúdo para os criadores de conteúdo, devido ao seu potencial impacto direto nos ganhos desses profissionais quando infringidas, os participantes foram indagados sobre a disponibilidade dessas normas. Os resultados indicaram que 10% dos participantes concordam plenamente que tais regras

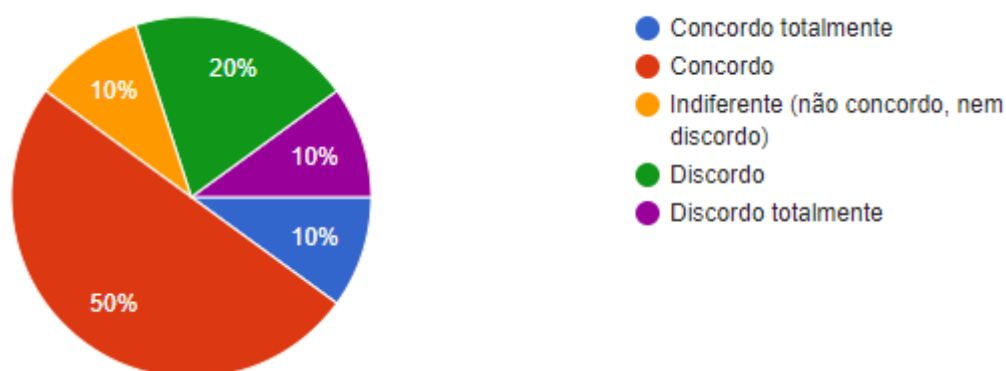


Figura 32 – Acessibilidade as informações de restrição de conteúdo da plataforma

são facilmente acessíveis, enquanto 50% manifestaram concordância em uma escala menor. Um contingente de 10% adotou uma posição neutra em relação a essa questão, ao passo que 20% discordaram da ideia de que essas informações são de fácil acesso. Adicionalmente, 10% discordaram integralmente dessa afirmativa. A Figura 32 apresenta esses resultados de forma gráfica.

Além das perguntas quantitativas destinadas a aprofundar a compreensão do ecossistema do YouTube do ponto de vista dos criadores de conteúdo, foram conduzidas duas perguntas qualitativas específicas com o intuito de coletar informações acerca da arquitetura proposta nesta dissertação, sob a perspectiva desses criadores. Após a exposição da proposta de arquitetura, os participantes foram indagados quanto à eficácia percebida da solução proposta para abordar a problemática da falta de transparência entre os criadores de conteúdo e a plataforma. Os resultados revelaram uma distribuição diversificada de opiniões, onde 30% dos participantes indicaram concordância total com a eficácia da proposta no contexto apresentado. Adicionalmente, outro grupo de 30% manifestou concordância, enquanto 20% adotaram uma posição neutra em relação à eficácia da solução. Por outro lado, 20% dos participantes discordaram da eficácia da proposta como solução para o problema em questão. Destaca-se que, na totalidade das respostas, não foram registradas opiniões na categoria de discordância total, conforme ilustrado na Figura 33.

Por último, os participantes foram indagados acerca da viabilidade percebida da implementação da nova arquitetura, com base em suas opiniões. Os resultados indicaram que 30% dos participantes concordam totalmente com a viabilidade da proposta no ambiente da plataforma de streaming, enquanto 40% expressaram concordância com essa viabilidade. Por outro lado, 10% dos participantes mantiveram uma postura neutra em relação à questão, e 20% discordaram da viabilidade da proposta. Assim como na questão anterior, não foram registradas respostas

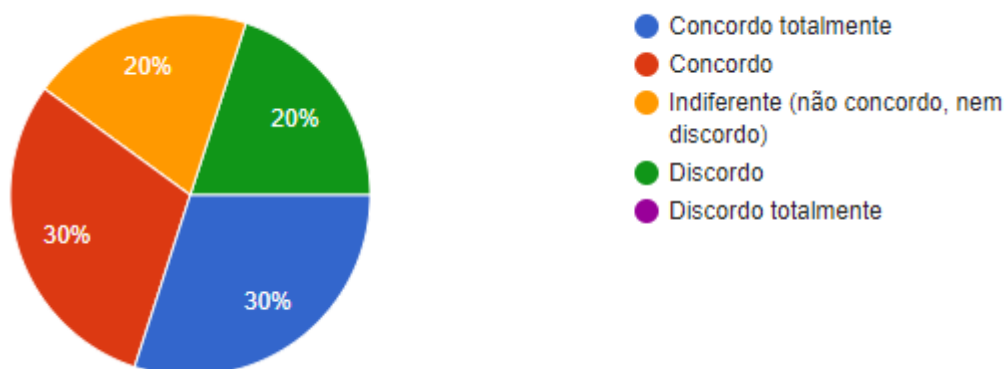


Figura 33 – Eficácia da arquitetura proposta para solucionar a falta de transparência entre criadores de conteúdo e a plataforma de streaming

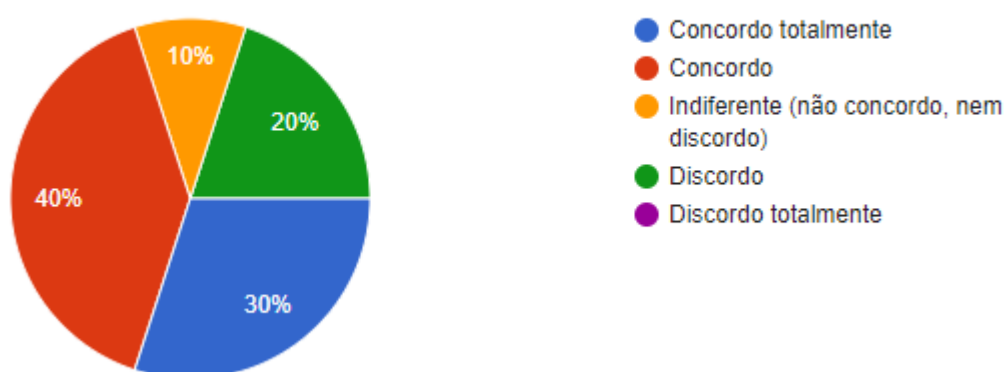


Figura 34 – Viabilidade de implantação da arquitetura proposta

na categoria de discordância total. A Figura 34 apresenta de forma gráfica os resultados dessas respostas.

5.3.1.2 Pesquisa qualitativa

As questões abertas, caracterizadas como questões qualitativas nesta pesquisa, consistiram em um conjunto de cinco perguntas. Estas foram aplicadas de forma opcional, resultando em um número de respostas que não corresponde exatamente ao total de participantes da pesquisa. Para analisar os dados coletados, empregou-se o processo de codificação aberta, axial e seletiva.

Na codificação aberta, identificam-se códigos de forma exploratória nas respostas dos participantes, permitindo a emergência de padrões e temas. A codificação axial organiza e relaciona esses códigos, buscando conexões mais estruturadas. Já a codificação seletiva identifica e refina temas-chave para apresentar conclusões de forma clara, muitas vezes por meio de

modelos gráficos.

Importa ressaltar que não se efetuou qualquer juízo de valor acerca das opiniões dos entrevistados. Dessa forma, as percepções dos participantes sobre o que consideram como consequente ou desempenho foram expressas sem interferências. Contudo, há situações em que se torna necessária a interpretação das respostas dos entrevistados. Por vezes, observa-se que, mesmo diante de uma pergunta específica, os entrevistados fornecem respostas distintas. Nesses casos, a realocação dos códigos para os termos apropriados se faz necessária.

O propósito das questões abertas consistia em alcançar uma compreensão aprofundada das opiniões dos criadores de conteúdo acerca de sua experiência na plataforma, bem como obter um feedback direto sobre a arquitetura proposta nesta pesquisa. Durante o processo de análise das respostas coletadas, emergiu um tema central que foi identificado em ambas as seções de perguntas, o qual diz respeito à transparência. Esse tema abrange a falta de transparência percebida e a necessidade expressa pelos participantes de implementar transparência nos processos presentes no ecossistema analisado.

Uma das primeiras indagações dirigidas aos participantes abordou os métodos empregados pela plataforma de streaming em análise para a monetização dos conteúdos oferecidos pelos criadores. Observa-se que a monetização de vídeos está sujeita a diversos fatores que impactam diretamente a remuneração do conteúdo, conforme evidenciado por um dos participantes que destacou: "A monetização de um conteúdo depende de muitas coisas como: - categoria do vídeo - tempo que o pessoal ficou assistindo o vídeo - taxa de clicks na propaganda antes do vídeo - conteúdo do vídeo...". Além disso, outro participante mencionou a possibilidade de oferecer assinaturas mensais para acesso a conteúdos exclusivos como uma nova fonte de receita.

Contudo, ao lado das descrições dos meios de monetização fornecidos, os participantes indicaram a ausência de detalhes específicos sobre as regras de monetização aplicadas nos referidos métodos, como expresso no seguinte trecho: "...como são calculados os valores e o quanto é pago pela visualização ou não de anúncio, não há uma especificação clara e de fácil acesso na plataforma...". Outro participante acrescentou: "...E esse valor é bem arbitrário. Não sei o quanto recebo por membro...". Assim, tornou-se evidente uma categorização central nas regras de monetização e plataformas, caracterizada pela falta de transparência.

De acordo com a pergunta anterior, os participantes foram indagados acerca dos mecanismos de transparência oferecidos pela plataforma em relação às informações de audiência e pagamentos recebidos. As respostas dos participantes destacaram a ênfase nos mecanismos de transparência voltados para as métricas de visualizações de seus conteúdos. Ficou evidente que a

plataforma não disponibiliza de maneira transparente tais informações aos criadores de conteúdo, obrigando-os a buscar fontes externas para obtê-las, conforme afirmou um dos participantes: "...todas essas questões foram sabidas a partir de páginas avulsas da internet."

Aqueles que não buscam informações externas à plataforma observaram uma ausência desses dados, como indicado nas seguintes declarações: "...não temos acesso do por que um vídeo deu mais que o outro detalhadamente" e em outro trecho: "...não fornece nenhum meio para verificar a origem das métricas disponibilizadas.". Alguns participantes mencionaram a métrica CPM¹, fornecida pela plataforma, que está diretamente relacionada ao valor recebido pelo criador de conteúdo para cada milhar de visualizações.

Outros participantes informaram que o YouTube disponibiliza um painel de controle (dashboard) com algumas métricas dos canais e de seus conteúdos, no entanto, não esclarece detalhes sobre a origem dessas métricas ou qualquer recurso de auditoria, conforme mencionado nos trechos a seguir: "O YouTube disponibiliza uma tela com as informações e métricas dos vídeos e a monetização, mas não fornece nenhum meio para verificar a origem das métricas disponibilizadas."

Diante do contexto dinâmico e criativo que caracteriza a produção de conteúdo, tornou-se essencial examinar as principais dificuldades enfrentadas pelos criadores de conteúdo no ambiente do YouTube. As respostas dos participantes abordaram diversas questões, abrangendo desde a presença excessiva de anúncios nos vídeos, potencialmente impactando a retenção do público, até problemas técnicos relacionados às notificações durante transmissões ao vivo, influenciando diretamente o número de visualizações simultâneas. Além disso, alguns participantes destacaram preocupações em relação aos tipos de conteúdo presentes na plataforma, considerando alguns como fúteis e carentes de criatividade, conforme expresso na citação: "Concorrência com conteúdos fúteis e falta de criatividade".

Observou-se que tais desafios podem estar vinculados às épocas em que os espectadores consomem determinados tipos de conteúdo, conhecidos como "ondas" ou vídeos virais, resultando em uma saturação de criadores de conteúdo produzindo materiais semelhantes. No entanto, dentro das respostas de alguns participantes, temas como a falta de transparência e as dificuldades para monetizar seus conteúdos foram destacados nas codificações, emergindo como a categoria central dessa questão.

Um participante relatou que, frequentemente, ao solicitar a monetização, enfrenta recusas sem explicações suficientes devido à falta de informações detalhadas fornecidas pelo YouTube.

¹ O custo por mil impressões (CPM) é uma métrica que representa quanto dinheiro os anunciantes estão gastando para veicular anúncios no YouTube.

Como ilustrado no trecho: "As principais dificuldades em meu tipo de conteúdo residem na falta de informações detalhadas fornecidas pelo YouTube. Por exemplo, ao solicitar a monetização, muitas vezes a recusa é comunicada sem esclarecimentos suficientes...". Outro participante mencionou que, mesmo ao monetizar, não possui clareza sobre o valor gerado pelo seu conteúdo, conforme expresso na citação: "O YouTube não deixa de forma clara o valor que será recebido pelo conteúdo disponibilizado...".

Considerando as adversidades identificadas pelos criadores de conteúdo, elaborou-se uma pergunta com o propósito de colher suas opiniões sobre possíveis aprimoramentos na plataforma, visando facilitar a criação de novos materiais. Nas respostas obtidas, a sugestão de implementar novas funcionalidades que proporcionassem maior controle sobre os anúncios veiculados antes e durante o conteúdo foi destacada. Em particular, foi sugerida a criação de uma funcionalidade que permitisse aos criadores listarem empresas indesejadas, evitando a exibição de anúncios incompatíveis com seus materiais. Outros criadores de conteúdo mencionaram a necessidade de novos mecanismos de segurança para prevenir a perda de contas hackeadas.

Por fim, observou-se que a maioria das respostas enfatizava a transparência nas informações e a imposição de regras claras sobre os conteúdos. Essas sugestões de melhoria incluíam a necessidade de transparência nos valores, conforme evidenciado em trechos como: "Exatamente o tema discutido no tópico anterior: a transparência das informações", "Melhorar a transparência sobre os valores das coisas...", "A disponibilização de regras claras e mecanismos de segurança dos canais" e "Transparência e praticidade". Essas proposições indicam um desejo coletivo por uma plataforma mais transparente e eficaz para aprimorar a experiência dos criadores de conteúdo.

As duas últimas perguntas formuladas se complementaram, estabelecendo uma correlação entre as respostas dos participantes e permitindo a materialização de suas preocupações e aspirações. Desse modo, foi criada a Figura 35, uma nuvem de palavras que destaca os termos mais citados nas respostas.

A análise revela que "transparência" foi o termo mais frequente, tanto no contexto da falta de transparência quanto na necessidade de mais transparência na plataforma. Termos como "informações" e "ausência" também se destacaram, relacionados ao ambiente tecnológico dos criadores de conteúdo. Essa representação visual sintetiza as principais preocupações dos participantes, destacando a importância atribuída à transparência no ecossistema da plataforma.

Por fim, com o objetivo de coletar um feedback direto dos criadores de conteúdos sobre a solução proposta nesta pesquisa, foi solicitada a opinião dos participantes sobre os pontos



Figura 35 – Nuvem de Palavras das respostas sobre dificuldades e melhorias na Plataforma de Streaming

positivos e negativos da solução proposta. Algumas opiniões dos participantes refletem ceticismo em relação à garantia de transparência dos dados sobre os vídeos por meio de uma camada *Blockchain* integrada à plataforma de streaming, conforme exposto no trecho: "...Ter uma camada *Blockchain* não necessariamente garante que teremos a transparência dos dados sobre o vídeo...". Contrariamente, a maioria dos participantes acredita que a implementação de uma nova camada utilizando a tecnologia *Blockchain*, juntamente com as funcionalidades apresentadas, poderá proporcionar maior transparência e controle sobre seus conteúdos, como indicado nos seguintes trechos das entrevistas: "... Aparentemente fornecerá mais recursos de segurança e controle sobre os vídeos...", "Mais transparência e a exclusão de instituições financeiras sempre agradam..." e "Como mencionei anteriormente, a transparência é fundamental."

Diante das contribuições dos participantes, ao analisar as respostas, percebe-se a presença de implicações e conexões relevantes sobre os temas discutidos. Notavelmente, a transparência emerge como um tema central entre as percepções compartilhadas. Algumas das sugestões apresentadas pelos participantes sugerem soluções aparentemente de baixa complexidade e viáveis para implementação nas plataformas de streaming.

Entre essas sugestões, destaca-se a proposta de aprimorar a apresentação de informações sobre os requisitos de monetização, proporcionando detalhes mais abrangentes sobre as receitas geradas pelos conteúdos. A criação de um campo específico, de fácil acesso, consolidando todas as regras e restrições, surge como uma solução para mitigar a necessidade de buscar informações em fontes não oficiais sobre a plataforma.

No contexto de transparência, a análise das respostas dos criadores de conteúdo destaca consistentemente a falta de transparência como uma preocupação significativa. Essa falta de clareza abrange desde os requisitos de monetização até as informações referentes às decisões, impactando a compreensão dos criadores sobre suas receitas, especialmente no contexto dos membros. A incerteza gerada pela falta de transparência intensifica frustrações, possivelmente resultando na migração para outras plataformas e na redução da produção de conteúdos.

A busca por transparência se estende a diversas áreas, indicando o anseio dos criadores por um entendimento claro e controle sobre a plataforma. No contexto geral, a busca pela transparência emerge como uma preocupação central, destacando a importância de melhorias nessa área para fortalecer a confiança entre criadores e plataforma.

5.3.2 Pesquisa com desenvolvedores *Blockchain*

O questionário elaborado para os desenvolvedores de *Blockchain* tem como finalidade analisar as vantagens e desvantagens da arquitetura proposta, visando avaliar a viabilidade de implementação sob uma perspectiva técnica. No total, foram contabilizados a participação de 20 desenvolvedores, permitindo a coleta de opiniões e sugestões relacionadas à arquitetura. No que diz respeito aos aspectos positivos e negativos da arquitetura proposta, observou-se que, segundo os participantes, a transparência fornecida pela arquitetura mostra-se tecnicamente viável e será de grande contribuição para os criadores de conteúdos e anunciantes presentes na plataforma.

Trechos das respostas dos participantes como "a solução se propõe a deixar o ambiente transparente, isso agrega valor para a marca do serviço que venha a utilizar...", "A transparência proporcionada ao utilizar a tecnologia *Blockchain* é o maior ganho da arquitetura proposta...", "A solução destacou aspectos extremamente positivos, especialmente no que diz respeito à transparência nas plataformas de streaming" e "Principal ponto positivo é a transparência dos dados para geradores de conteúdo" ressaltaram essa característica positiva da arquitetura.

Da mesma forma, a remoção de instituições financeiras foi apontada pelos participantes de duas perspectivas. Uma delas, mais positiva, destaca a redução de custos e a descentralização dos meios de pagamento, conforme evidenciado nos trechos: "inserir um contrato inteligente no processo de pagamento é viável", "a eliminação da necessidade de uma instituição financeira, o que irá reduzir os custos operacionais da transação..." e "a ideia de fazer divisão de pagamentos pode ser muito útil em casos de reivindicação de direitos autorais...".

Por outro lado, a remoção das instituições financeiras gerou preocupações, especialmente devido à mudança na forma de recebimento, que passaria a ocorrer por meio de criptomoedas

(tokens). Essa alteração, segundo os participantes pode representar uma barreira para a aceitação por parte dos criadores de conteúdo, conforme indicado em trechos como "pode criar uma certa barreira para os criadores de conteúdos não familiarizados com a web3.0", "...remover as instituições financeiras desse fluxo de pagamentos, os criadores de conteúdo serão obrigados a utilizar alguma forma de token como meio de recebimento dos valores." e "cria uma dificuldade que é a conversão de tokens/criptomoedas em moedas reais".

Outro ponto abordado pelos participantes diz respeito à escalabilidade do uso da tecnologia *Blockchain* para armazenar todas as informações de audiência, uma vez que cada atualização resultará na geração de um novo bloco na rede. Essa preocupação foi expressa nos seguintes trechos: "...a escalabilidade disso pode ser um desafio pois existe muito conteúdo sendo gerado/consumido" e "...acredito que irá gerar muitos blocos na rede...". Além disso, outro ponto identificado foi a aceitação da nova arquitetura pela plataforma de streaming, como evidenciado nas seguintes citações: "Possível rejeição das plataformas de streaming...", "...convencer o YouTube a investir alguns recursos adicionais para cada interação de usuário na plataforma" e "Uma ameaça externa à pesquisa é a não aceitação pelas plataformas de streaming, visto que a arquitetura representa uma mudança no atual processo de negócios".

Na perspectiva de aprimoramentos futuros na solução apresentada, conduziu-se uma segunda indagação, na qual os participantes foram solicitados a oferecer sugestões consideradas pertinentes para implementação na arquitetura proposta. Com isso, identificou-se que alguns desenvolvedores propuseram a descentralização completa da plataforma ou do armazenamento de vídeos, com o intuito de fornecer uma estrutura sem um agente central que exercesse controle sobre os conteúdos. Tal abordagem foi identificada em trechos como: "...acredito que o poder de uma arquitetura descentralizada e distribuída está em promover mais liberdade para os pequenos agentes..." e "...Uma possível forma de resolver seria criar uma plataforma de vídeos do zero, totalmente em blockchain."

No contexto da descentralização, emergiram sugestões para a implementação de mecanismos de incentivo, como ganhos de criptomoedas para os espectadores, visando integrá-los aos nós da rede, com o intuito de otimizar a escalabilidade do aplicativo. Outra consideração dos participantes foi relativa à adesão dos criadores de conteúdo ao uso de carteiras digitais de criptoativos. Nesse sentido, foi proposta a criação de um mecanismo de conversão das criptomoedas em moedas locais, que poderiam ser transferidas para contas convencionais, ou a disponibilização do uso de instituições financeiras para o recebimento dos criptoativos, com a adição dos custos associados a essa operação. Este ponto foi observado nos trechos: "Sugiro

explicar melhor como o dinheiro pago em moeda local do anunciante chegará até a conta corrente do criador de conteúdo."e "Manter a opção do uso de uma instituição financeira, para deixar claro para os criadores que há uma opção de receber os créditos sem necessariamente precisarem ter uma 'digital Wallet', embora tenha um custo envolvido na operação."

Outra indagação levantada pelos participantes envolveu a dificuldade de adesão das plataformas de streaming à arquitetura proposta, atribuída à falta de transparência nas regras de monetização e à alteração no modelo de negócio com a exclusão das instituições financeiras. Como apresentado no trecho: "O principal desafio é por provocar uma mudança no processo de negócio das plataformas de streaming"

Deste modo, foi possível observar que as principais implicações e conexões identificadas nas respostas estão relacionadas à introdução da tecnologia *Blockchain* nas plataformas. A transparência mostrou ser uma das principais vantagens presentes na solução proposta. A eliminação de intermediários financeiros reduz os custos, mas impõe a adoção de carteiras digitais, o que exigirá um período de adaptação por parte dos usuários. Entretanto, essa transição não parece representar uma barreira significativa, considerando que essa tecnologia está se tornando cada vez mais comum. A dificuldade na conversão de tokens, conforme identificada nas respostas dos participantes, pode ser amenizada pelo emprego de tokens pareados a moedas já existentes, como o dólar e o real, ou mesmo através do uso de tokens oficiais de moedas existentes. Por fim, os aspectos relacionados à escalabilidade devem ser cuidadosamente considerados durante a fase de implementação.

5.4 CONSIDERAÇÕES FINAIS

No presente capítulo, foram realizadas avaliações da arquitetura proposta, por meio de testes em diferentes redes *Blockchain*, com o intuito de verificar a viabilidade técnica e financeira da solução proposta para uma plataforma de streaming. Para tal, foi implementado uma prova de conceito que simulou uma plataforma de streaming e integrou o middleware do StreamingChain. Os resultados obtidos permitiram a identificação da viabilidade técnica e financeira da implementação da arquitetura de software proposta.

Por fim, a análise da solução proposta foi conduzida de maneira quantitativa e qualitativa, envolvendo criadores de conteúdo e desenvolvedores de software especializados em *Blockchain*, por meio da utilização de formulários eletrônicos. Os resultados obtidos permitiram a obtenção de informações relevantes sobre o desempenho da plataforma de streaming em consideração, além de proporcionar avaliações distintas da arquitetura proposta por esses dois públicos, cada

um com perspectivas específicas de suas respectivas áreas. Essa abordagem visou garantir não apenas a eficácia técnica, mas, crucialmente, a aceitação favorável pelos usuários finais.

6 CONCLUSÃO

Este capítulo apresenta as considerações finais sobre os tópicos abordados nesta dissertação, abordando as principais contribuições alcançadas, as limitações deste estudo e, por fim, fornecendo indicações para trabalhos futuros.

6.1 CONSIDERAÇÕES FINAIS

Ao explorar o cenário do consumo de serviços multimídia em plataformas de streaming, deparamo-nos com uma transformação impulsionada pelo aumento do uso de smartphones, avanço das redes móveis e de banda larga, e aprimoramento das tecnologias de software e armazenamento de dados. Essas plataformas, agora consolidadas como fontes globais de entretenimento, alcançaram números expressivos de audiência e receita, com o uso de estratégias de monetização, desenvolvidas e adaptadas para atender às diversas demandas dos consumidores, demonstrando a flexibilidade desse ecossistema dinâmico.

Na busca por soluções computacionais que proporcionassem transparência e segurança na manipulação de dados, a tecnologia *Blockchain* emergiu como resposta, especialmente na indústria de entretenimento digital, onde a falta de transparência na implementação de serviços de audiência e cálculos de métricas de monetização representava um desafio. Este estudo explorou as potencialidades do *Blockchain*, identificando desafios específicos no ecossistema das organizações de entretenimento digital.

Por meio da análise do estado da arte, observaram-se diversas soluções na indústria do entretenimento voltadas para a promoção de transparência entre os criadores de conteúdo. Contudo, nota-se que a literatura predominantemente destaca a criação de novas plataformas, com a tecnologia *Blockchain* centralizando o gerenciamento de informações e armazenamento de conteúdos em diversos setores de entretenimento. No entanto, identificou-se uma lacuna no que concerne à implementação da *Blockchain* na estrutura operacional de plataformas de streaming já existentes. Nessa perspectiva, propõe-se a utilização da *Blockchain* como uma camada adicional responsável por armazenar e gerenciar regras de monetização e audiência, atuando como uma segunda fonte de informação e mecanismo de auditoria. Além disso, vislumbra-se a viabilidade de realizar o gerenciamento de pagamentos financeiros por meio da rede *Blockchain*, excluindo as instituições financeiras da dinâmica operacional das plataformas de streaming.

No contexto apresentado, foram identificados os requisitos funcionais necessários para criar uma arquitetura de software que preenche a lacuna identificada. Com base nisso, elaborou-se

uma arquitetura de software usando o modelo C4 para definir os diferentes elementos do middleware responsável por gerenciar e tornar transparentes os conteúdos fornecidos nas plataformas de streaming.

Com a arquitetura devidamente especificada, foi iniciada a avaliação da solução proposta nos âmbitos técnico e financeiro, envolvendo criadores de conteúdo e desenvolvedores de software com experiência em tecnologia *Blockchain*. A análise técnica e financeira revelou a viabilidade da arquitetura proposta em várias redes *Blockchain*, alcançando baixos custos, da ordem de frações de centavos de dólar, para a execução de transações em uma das redes testadas. Paralelamente, implementou-se um sistema de monitoramento capaz de rastrear informações de hardware, solicitações e custos das transações em redes *Blockchain*.

Adicionalmente, foram empregados formulários eletrônicos para examinar o panorama das plataformas de *streaming* do ponto de vista dos criadores de conteúdo, coletando *feedbacks* sobre a arquitetura proposta com criadores de conteúdos e especialistas técnicos. Este método proporcionou uma compreensão mais abrangente das necessidades e percepções dos usuários finais, contribuindo para a avaliação da arquitetura proposta.

Desse modo, foi possível desenvolver uma arquitetura de software, baseada na tecnologia *Blockchain*, com o objetivo de viabilizar a transparência no gerenciamento de audiência e pagamentos em plataformas de *streaming*. Ao empregar a *Blockchain* como uma camada adicional de armazenamento e gerenciamento de informações, propõe-se não apenas promover a integridade dos dados, mas também eliminar intermediários financeiros, potencializando a eficiência operacional das plataformas de *streaming*.

Ao longo do estudo, os requisitos funcionais foram considerados, culminando na elaboração de uma arquitetura com viabilidade financeira e computacional para implementação no cenário proposto. A avaliação técnica e financeira da solução proposta evidenciou sua viabilidade em diversas redes *Blockchain*, destacando sua capacidade de execução de transações com custos mínimos. Ademais, a incorporação do *feedback* dos criadores de conteúdo e especialistas técnicos proporcionou resultados valiosos para o refinamento contínuo da arquitetura, reforçando seu potencial para promover a transparência e aprimorar a experiência dos usuários nas plataformas de *streaming*.

6.2 LIMITAÇÕES

A presente pesquisa, inserida nesta dissertação, ofereceu contribuições significativas para o estado da arte, por meio da identificação de descobertas e do desenvolvimento de uma

arquitetura de software. Esta arquitetura visa proporcionar maior transparência para a indústria de entretenimento digital. No entanto, durante o estudo, foram identificadas limitações específicas relacionadas ao monitoramento e à integração com aplicações de mercado, como discutido adiante.

No âmbito dos recursos computacionais destinados ao processamento de transações, pode-se destacar:

- **Infraestrutura robusta:** A ausência de uma infraestrutura robusta para realizar testes experimentais em várias redes *Blockchain* impactou a avaliação da robustez da arquitetura, especialmente em cenários de alta demanda de dados e execuções prolongadas. Essa limitação trouxe desafios para a análise da capacidade de resposta e eficácia da arquitetura.
- **Interoperabilidade:** A ausência de padrões consolidados na implementação da *Blockchain* pode resultar em desafios significativos de interoperabilidade entre diversas plataformas e redes, representando uma limitação devido a falta de testes abordando esse aspecto específico.

No contexto das integrações com plataformas já existentes no mercado, ressalta-se a seguinte limitação:

- **Implementação em ambiente externo:** A não implementação da arquitetura proposta em uma plataforma de streaming já existente no mercado apresenta uma limitação para verificar sua aplicabilidade no contexto. A ausência de aplicação prática impede uma análise completa do desempenho da arquitetura no cenário específico das plataformas de streaming, enfatizando a importância de futuras oportunidades para validação prática.

6.3 TRABALHOS FUTUROS

Como perspectivas de pesquisa futura, propõe-se a condução de novos estudos nas diversas dimensões e funcionalidades contempladas na arquitetura, detalhadas a seguir.

- **Desenvolvimento de mecanismos de transparência e auditoria para anunciantes:** Explorar a criação de mecanismos adicionais que proporcionem transparência e permitam auditorias específicas para anunciantes, fortalecendo a confiança e a compreensão do processo.

- **Avaliações quantitativas e qualitativas ampliadas com audiência e anunciantes:** Expandir as avaliações existentes, tanto quantitativas quanto qualitativas, promovendo uma compreensão mais abrangente das percepções e expectativas tanto da audiência quanto dos anunciantes.
- **Implementação de novas funcionalidades de controle de conteúdo para criadores de conteúdo:** Introduzir funcionalidades adicionais de controle de conteúdo que capacitem os criadores a gerenciar e personalizar efetivamente o material apresentado, promovendo maior autonomia e adequação às suas necessidades.
- **Incorporação de funcionalidade de aprendizado de máquina no sistema de monitoramento *Blockchain*:** Investigar a viabilidade e benefícios da integração de recursos de aprendizado de máquina no sistema de monitoramento *Blockchain*, visando identificar anomalias, análise de padrões, otimização de recursos, melhorias na eficácia na análise de dados e no fornecimento de insights relevantes.
- **Validação da arquitetura proposta em uma plataforma de mercado:** Conduzir uma validação prática da arquitetura em uma plataforma de streaming já consolidada no mercado, possibilitando uma avaliação mais contextualizada e identificação de desafios específicos de implementação.
- **Validação de escalabilidade do armazenamento dos contratos inteligentes:** Realizar uma avaliação prática da capacidade de armazenamento escalável de contratos inteligentes em uma rede blockchain, considerando a quantidade crescente de conteúdo disponibilizado em uma plataforma de streaming.

Além disso, em virtude das leis de proteção de dados impostas no Brasil e no mundo, um trabalho futuro pertinente é a implementação dessas normativas, visando alinhar a arquitetura proposta nesta dissertação aos requisitos estabelecidos pelos critérios de segurança e proteção de dados.

REFERÊNCIAS

- ALHARBY, M.; MOORSEL, A. V. Blockchain-based smart contracts: A systematic mapping study. **arXiv preprint arXiv:1710.06372**, 2017.
- AWUOR, F. M. Towards creating transparency in loyalty payments in music streaming platforms. p. 1–8, 2020.
- BAHGA, A.; MADISETTI, V. K. Blockchain platform for industrial internet of things. **Journal of Software Engineering and Applications**, Scientific Research Publishing, v. 9, n. 10, p. 533–546, 2016.
- BARMAN, N.; DEEPAK, G.; MARTINI, M. G. Blockchain for video streaming: Opportunities, challenges, and open issues. **Computer**, IEEE, v. 53, n. 7, p. 45–56, 2020.
- BARUA, S.; TALUKDER, D. A blockchain based decentralized video streaming platform with content protection system. p. 1–6, 2020.
- BIGCHAINDB. 2022. Disponível em: <<https://www.bigchaindb.com/>>.
- BOWMAN, N.; JOHNSON, M. R.; WOODCOCK, J.; PARTIN, W.; JACKSON, N. Sociality and monetization on live streaming platforms. **AoIR Selected Papers of Internet Research**, 2020.
- BROWN, S. **The C4 model for visualising software architecture**. 2022. <<https://c4model.com/>>. Acesso em 10 de Junho de 2023.
- CHAVAN, S.; WARKE, P.; GHUGE, S.; DEOLEKAR, R. V. Music streaming application using blockchain. p. 1035–1040, 2019.
- CORDA. 2022. Disponível em: <<https://corda.net/>>.
- COTTON, I. W.; JR, F. S. G. Data structures and techniques for remote computer graphics. p. 533–544, 1968.
- DARWICH, M.; KHALIL, K.; ISMAIL, Y.; BAYOUMI, M. Blockchain-enhanced distributed storage for cloud-based video streaming: Enhancing security and scalability. In: **SPRINGER. Proceedings of the Future Technologies Conference**. [S.l.], 2023. p. 426–439.
- DIAS, D.; BENET, J. **Distributed web applications with IPFS, tutorial**. [S.l.], 2016. 616–619 p.
- DOAN, T. V.; PHAM, T. D.; OBERPRIELER, M.; BAJPAI, V. Measuring decentralized video streaming: A case study of dtube. p. 118–126, 2020.
- ELANGO, D. Social media video creators monetization and business on youtube. **Available at SSRN 3320253**, 2019.
- ERL, T. **SOA: principles of service design**. [S.l.]: Prentice Hall, 2008.
- ESMAILI, M.; JAVIDAN, R. A distributed blockchain-based video sharing system with copy-right, integrity, and immutability. p. 86–92, 2020.
- FERNANDES, E. M.; MAIA, Â. Grounded theory. Universidade do Minho. Centro de Estudos em Educação e Psicologia (CEEP), 2001.

FIELDING, R. T. **Architectural styles and the design of network-based software architectures**. [S.l.]: University of California, Irvine, 2000.

GARLAN, D. **Software architecture**. Wiley, 2001.

GMI, B. **YouTube statistics 2024 [users by country + demographics]**. 2024. Disponível em: <<https://www.globalmediainsight.com/blog/youtube-users-statistics/#stat>>.

GÖTTING, M. C. **Topic: Music streaming services worldwide**. 2023. Disponível em: <<https://www.statista.com/topics/11066/music-streaming-services-worldwide/#topicOverview>>.

HYPERLEDGER. 2022. Disponível em: <<https://www.hyperledger.org/>>.

JOHNSTON, D.; YILMAZ, S.; KANDAH, J.; BENTENITIS, N.; HASHEMI, F.; GROSS, R.; WILKINSON, S.; MASON, S. The general theory of decentralized applications, dapps. github—. 2014.

KIM, A.; KIM, M. A study on blockchain-based music distribution framework: focusing on copyright protection. p. 1921–1925, 2020.

LAURA, W. **Global Music Streaming Market 2020-2027: Revenues to Grow at a CAGR of 7.4%, Landing the Market Size at \$46.9 Billion by 2027 - ResearchAndMarkets.com**. 2020. Disponível em: <<https://www.businesswire.com/news/home/20200813005306/en/Global-Music-Streaming-Market-2020-2027-Revenues-to-Grow-at-a-CAGR-of-7.4-Landing-the-Market-Size-at-46.9-Billion-by-2027---ResearchAndMarkets.com>>.

LEE, Y.-C.; YEN, C.-H.; CHIU, P.-T.; KING, J.-T.; FU, W.-T. Tip me! tipping is changing social interactions on live streams in china. In: **Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems**. [S.l.: s.n.], 2018. p. 1–6.

LIU, M.; YU, F. R.; TENG, Y.; LEUNG, V. C.; SONG, M. Deep reinforcement learning (drl)-based transcoder selection for blockchain-enabled video streaming. p. 1–6, 2018.

LIU, M.; YU, F. R.; TENG, Y.; LEUNG, V. C.; SONG, M. Distributed resource allocation in blockchain-based video streaming systems with mobile edge computing. **IEEE Transactions on Wireless Communications**, IEEE, v. 18, n. 1, p. 695–708, 2018.

MAJEED, U.; KHAN, L. U.; YAQOOB, I.; KAZMI, S. A.; SALAH, K.; HONG, C. S. Blockchain for iot-based smart cities: Recent advances, requirements, and future challenges. **Journal of Network and Computer Applications**, Elsevier, v. 181, p. 103007, 2021.

MARÉCHAUX, J.-L. Combining service-oriented architecture and event-driven architecture using an enterprise service bus. **IBM developer works**, v. 12691275, 2006.

MENDELEY. 2023. <<https://www.mendeley.com>>. Acesso em 8 de fevereiro de 2023.

MOORE, F. Ifpi global music report 2019. **International Federation of the Phonographic Industry**, p. 1–23, 2020.

MURKO, D. 2023. Disponível em: <<https://streamscharts.com/news/2023-q1-global-livestreaming-landscape>>.

NAIK, H. M.; VAITY, S. A.; D'MELLO, S. N.; PATIL, S. D. Decentube: A decentralized video streaming platform (dapp) using ethereum and ipfs. **International Journal of Next-Generation Computing**, p. 178–190, 2020.

- NAKAMOTO, S. A peer-to-peer electronic cash system. **Bitcoin**.—URL: <https://bitcoin.org/bitcoin.pdf>, v. 4, p. 2, 2008.
- NEMOTO, T.; BEGLAR, D. Likert-scale questionnaires. In: **JALT 2013 conference proceedings**. [S.l.: s.n.], 2014. p. 1–8.
- NIELSEN, N. **Consumo total equivalente de álbuns NOS Estados Unidos Aumentou 23% EM 2018**. 2022. Disponível em: <<https://www.nielsen.com/pt/insights/2018/us-music-mid-year-report-2018/>>.
- OSETSKYI, V. What is smart contracts blockchain and its use cases in business. **Medium**, 2018.
- PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de software-9**. [S.l.]: McGraw Hill Brasil, 2021.
- ROCHA, D. F.; BITTENCOURT, I. I.; DERMEVAL, D.; ISOTANI, S. Uma revisão sistemática sobre a educação do surdo em ambientes virtuais educacionais. v. 25, n. 1, p. 1263, 2014.
- RODRÍGUEZ-GIL, L.; GARCÍA-ZUBIA, J.; ORDUNA, P.; IPIÑA, D. Lopez-de. An open and scalable web-based interactive live-streaming architecture: The wilsp platform. **IEEE Access**, IEEE, v. 5, p. 9842–9856, 2017.
- ROWSE, C. S. P. **Building Infrastructure Platforms**. 2022. <<https://martinfowler.com/articles/building-infrastructure-platform.html>>. Acesso em 09 de Junho de 2023.
- SATHISH, S. K.; PATANKAR, A. A.; KHANNA, H. Aurum: A blockchain based decentralized video streaming platform. p. 1–8, 2019.
- SHAMSEER, L.; MOHER, D.; CLARKE, M.; GHERSI, D.; LIBERATI, A.; PETTICREW, M.; SHEKELLE, P.; STEWART, L. A. Preferred reporting items for systematic review and meta-analysis protocols (prisma-p) 2015: elaboration and explanation. **Bmj**, British Medical Journal Publishing Group, v. 349, 2015.
- SOARES, R.; ARAÚJO, A.; RODRIGUES, G.; ALENCAR, C. Streaming platforms based on blockchain technology: A business model impact analysis. In: SPRINGER. **International Conference on Information Technology-New Generations**. [S.l.], 2023. p. 143–149.
- SOLIDITY. 2022. Disponível em: <<https://solidity.readthedocs.io/en/v0.4.24/>>.
- SOMMERVILLE, I. **Engenharia de software**. Pearson Prentice Hall, 2011. ISBN 9788579361081. Disponível em: <<https://books.google.com.br/books?id=H4u5ygAACAAJ>>.
- TIAN, F.; LIANG, P.; BABAR, M. A. Relationships between software architecture and source code in practice: An exploratory survey and interview. **Information and Software Technology**, Elsevier, v. 141, p. 106705, 2022.
- TORBENSEN, A.; CIRIELLO, R. Tuning into blockchain: Challenges and opportunities of blockchain-based music platforms. In: **Twenty-Seventh European Conference on Information Systems (ECIS2019), Stockholm-Uppsala, Sweden**. [S.l.: s.n.], 2019.
- TURNER, M.; BUDGEN, D.; BRERETON, P. Turning software into a service. **Computer**, IEEE, v. 36, n. 10, p. 38–44, 2003.

- VIMAL, S.; SRIVATSA, S. A new cluster p2p file sharing system based on ipfs and blockchain technology. **Journal of Ambient Intelligence and Humanized Computing**, Springer, p. 1–7, 2019.
- WANG, S.; YUAN, Y.; WANG, X.; LI, J.; QIN, R.; WANG, F.-Y. An overview of smart contract: architecture, applications, and future trends. p. 108–113, 2018.
- WAYNE, M. L. Global streaming platforms and national pay-television markets: a case study of netflix and multi-channel providers in israel. **The Communication Review**, Taylor & Francis, v. 23, n. 1, p. 29–45, 2020.
- WÖHRER, M.; ZDUN, U.; RINDERLE-MA, S. Architecture design of blockchain-based applications. p. 173–180, 2021.
- XIAO, Y.; WATSON, M. Guidance on conducting a systematic literature review. **Journal of planning education and research**, SAGE Publications Sage CA: Los Angeles, CA, v. 39, n. 1, p. 93–112, 2019.
- XU, B.; XU, L. D.; WANG, Y.; CAI, H. A distributed dynamic authorisation method for internet+ medical & healthcare data access based on consortium blockchain. **Enterprise Information Systems**, Taylor & Francis, v. 16, n. 12, p. 1922757, 2022.
- YAHYA, A.; HABBAL, A. Music royalty payment scheme using blockchain technology. p. 539–545, 2021.
- YUNQI, M. Distributed rewarding for browser-based p2p video streaming with ethereum blockchain. p. 484–486, 2021.
- ZHANG, H. Online platform construction of soa architecture and streaming media technology under edge computing. **International Journal of System Assurance Engineering and Management**, Springer, p. 1–10, 2022.
- ZHENG, Z.; XIE, S.; DAI, H.-N.; CHEN, X.; WANG, H. Blockchain challenges and opportunities: A survey. **International journal of web and grid services**, Inderscience Publishers (IEL), v. 14, n. 4, p. 352–375, 2018.