

UNIVERSIDADE FEDERAL DE JUIZ DE FORA
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Raiane Querino Coelho

BlockFlow: Uma Arquitetura Baseada em *Blockchain* para Confiança em
Workflows Científicos Colaborativos Apoiados por uma Plataforma de
Ecossistema de *Software*.

Juiz de Fora
2021

Raiane Querino Coelho

**BlockFlow: Uma Arquitetura Baseada em *Blockchain* para Confiança em
Workflows Científicos Colaborativos Apoiados por uma Plataforma de
Ecosistema de *Software*.**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Juiz de Fora como requisito parcial à obtenção do grau de Mestre em Ciência da Computação.

Orientadora: Dr^a Regina Maria Maciel Braga Villela

Juiz de Fora

2021

Ficha catalográfica elaborada através do Modelo Latex do CDC da UFJF
com os dados fornecidos pelo(a) autor(a)

Coelho, Raiane.

BlockFlow: Uma Arquitetura Baseada em *Blockchain* para Confiança em *Workflows* Científicos Colaborativos Apoiados por uma Plataforma de Ecossistema de *Software*. / Raiane Querino Coelho. – 2021.

118 f. : il.

Orientadora: Regina Maria Maciel Braga Villela

Dissertação (Mestrado) – Universidade Federal de Juiz de Fora, Instituto de Ciências Exatas. Programa de Pós-Graduação em Ciência da Computação, 2021.

1. Blockchain. 2. Computação em Nuvem. 3. Ecossistema de Software Científico. 4. Experimentos Científicos Colaborativos. 5. Confiabilidade. 6. Proveniência de dados. 7. Reprodutibilidade. I. Braga, Regina, orient. II. Título.

Raiane Querino Coelho

BlockFlow: Uma Arquitetura Baseada em *Blockchain* para Confiança em *Workflows* Científicos Colaborativos Apoiados por uma Plataforma de Ecosistema de *Software*.

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Juiz de Fora como requisito parcial à obtenção do grau de Mestre em Ciência da Computação.

Aprovada em 14 de Junho de 2021

BANCA EXAMINADORA

Dr^a Regina Maria Maciel Braga Villela - Orientador
Universidade Federal de Juiz de Fora

Dr. José Maria Nazar David - Membro Interno
Universidade Federal de Juiz de Fora

Dr. Mário Antônio Ribeiro Dantas – Membro Interno
Universidade Federal de Juiz de Fora

Dr. Tadeu Moreira de Classe – Membro Externo
Universidade Federal do Estado do Rio de Janeiro

Dedico este trabalho a todos que eu amo.

AGRADECIMENTOS

Agradeço à Deus por me guiar, abençoar minhas escolhas e por me pegar no colo em todos os momentos difíceis. Quem o conhece sabe a sua maneira de agir e sua infinita bondade. Pois dele, por ele e para ele são todas as coisas. A ele seja a glória para sempre!

Ao meu mozinho, marido Fábio, pelo amor, companheirismo e paciência. Obrigada, por me fazer acreditar em mim mesmo e por estar sempre comigo.

À minha família, que sem sombra de dúvida são a base de tudo. Em especial à minha vó Iracema pelo apoio incondicional durante toda a minha vida e por todo carinho e amor. À minha mãe, Alessandra e aos meus irmãos “Querino’s brothers”, Yago e João Paulo. Obrigada, por todo amor, por serem o meu impulso e por me encorajarem a seguir meus objetivos e serem sempre o meu ombro amigo.

À Helena e a Isadora por segurarem a barra e por colaborarem para que eu pudesse trabalhar e estudar. Obrigada pela compreensão e carinho.

Aos professores do PGCC em especial, à minha orientadora, Regina, ao qual eu tenho um carinho especial (acredito que Deus coloca pessoas no nosso caminho, para que se cumpra nossos propósitos). Obrigada pela confiança, paciência, dedicação e por acreditar no meu potencial.

Aos meus amigos e colegas que fiz no PGCC, obrigada pelas risadas, incentivos, ajudas e parceria.

“Aqueles que semeiam com lágrimas, com cantos de alegria colheirão”. Salmos 126:5-6

RESUMO

Atualmente, os experimentos científicos são realizados de forma colaborativa. Na colaboração científica, o compartilhamento de dados, a troca de ideias e resultados são essenciais para promover o conhecimento e acelerar o desenvolvimento da ciência. Nesse sentido, com atividades cada vez mais complexas, os *workflows* científicos estão se tornando mais intensivos em dados, exigindo ambientes colaborativos, distribuídos e de alto desempenho (HPC), como grades ou nuvens, para sua execução. Esses ambientes em nuvem estão se tornando cada vez mais adotados por cientistas, pois fornecem escalabilidade e provisionamento de recursos sob demanda. Por outro lado, em experimentos científicos colaborativos baseados em dados, a interoperabilidade, a privacidade e a confiança devem ser consideradas. Para isso, dados de proveniência tem sido amplamente reconhecido por fornecer um histórico das etapas da realização de experimentos científicos, auxiliando na reprodutibilidade dos resultados. Além disso, uma das tecnologias que podem melhorar a colaboração, rastreabilidade e confiança nos resultados científicos, com o objetivo de reprodutibilidade, é *blockchain*. Nesse sentido, este trabalho propõe uma arquitetura baseada em *blockchain*, proveniência e infraestrutura em nuvem para trazer confiança na execução de experimentos científicos colaborativos. A arquitetura permite que os pesquisadores criem ambientes distribuídos e confiáveis para a experimentação científica colaborativa, apoiando a coleta e análise de dados de *workflows* científicos. A solução oferece um ambiente distribuído, que privilegia a interoperabilidade, a privacidade e a confiança em dados de fontes heterogêneas, para permitir a reprodutibilidade dos resultados obtidos na experimentação científica colaborativa.

Palavras-chave: Blockchain. Computação em Nuvem. Ecossistema de Software Científico. Experimentos Científicos Colaborativos. Confiabilidade. Proveniência de dados. Reprodutibilidade.

ABSTRACT

Currently, scientific experiments are carried out collaboratively. In scientific collaboration, data sharing, the exchange of ideas and results are essential to promote knowledge and accelerate the development of science. In this sense, with increasingly complex activities, scientific workflows are becoming more data-intensive, requiring collaborative, distributed, and high-performance environments (HPC), such as grids or clouds, for its execution. Cloud environments are becoming increasingly adopted by scientists as they provide scalability and provisioning of resources on demand. On the other hand, in collaborative scientific experiments based on data, interoperability, privacy, and trust must be considered. For this, provenance has been widely recognized to provide a history of the steps taken in carrying out scientific experiments, assisting in the reproducibility of scientific results. In addition, one of the technologies that can improve collaboration, traceability, and confidence in scientific results, with the objective of reproducibility, is Blockchain. In this vein, this work proposes an architecture based on blockchain, provenance, and cloud infrastructure to bring confidence in the execution of collaborative scientific experiments. The architecture allows researchers to create distributed and reliable environments for collaborative scientific experimentation, supporting the collection and analysis of data from scientific workflows. The solution provides a distributed environment, which privileges interoperability, privacy, and trust in data from heterogeneous sources, to allow the reproducibility of the results obtained in collaborative scientific experimentation

Keywords: Blockchain. Cloud computing. Scientific Software Ecosystem. Collaborative Scientific Experiments. Reliability. Provenance. Reproducibility.

LISTA DE ILUSTRAÇÕES

Figura 1	– Ciclo de vida de um experimento científico	22
Figura 2	– Visão Geral da Plataforma E-SECO	23
Figura 3	– Estrutura e encadeamento de blocos blockchain	24
Figura 4	– Árvore <i>Merkle</i>	25
Figura 5	– Chaves públicas e privadas <i>blockchain</i>	29
Figura 6	– Modelo de proveniência PROV.	33
Figura 7	– Modelo de proveniência ProvONE.	35
Figura 8	– Fluxo de filtragens e etapas do mapeamento do sistemático.	40
Figura 9	– Porcentagem de artigos retornados por base.	41
Figura 10	– O total de artigos aceitos e rejeitados.	42
Figura 11	– Áreas de aplicação.	43
Figura 12	– Onde os estudos foram publicados.	44
Figura 13	– Distribuição dos estudos no decorrer dos anos.	44
Figura 14	– Método ou metodologia de pesquisa.	45
Figura 15	– Motivação uso tecnologia <i>blockchain</i> para proveniência.	45
Figura 16	– Plataformas ou arquiteturas de <i>blockchain</i>	46
Figura 17	– Mecanismos de Consensos.	47
Figura 18	– Modelos de Proveniência.	48
Figura 19	– Arquitetura BlockFlow.	56
Figura 20	– Integração E-SECO e Arquitetura BlockFlow.	56
Figura 21	– Um exemplo de uma solicitação para a camada <i>RESTful Web Service API</i> da BlockFlow.	57
Figura 22	– Digrama de solicitações e respostas para camada <i>API RESTful Web Service</i> da BlockFlow.	58
Figura 23	– Encadeamento de tarefas em um <i>workflow</i> científico.	59
Figura 24	– Exemplo de mapeamento de uma tarefa de um <i>workflow</i> para modelo ProvONE.	60
Figura 25	– Exemplo de mapeamento de uma tarefa de um <i>workflow</i> para modelo ProvONE.	61
Figura 26	– Modelo de classes da arquitetura BlockFlow.	62
Figura 27	– Digrama de solicitações e respostas para camada Model da BlockFlow.	63
Figura 28	– Interface do usuário, construída através de forma JSON.	63
Figura 29	– Fluxos de chamadas a camada <i>Client</i>	64
Figura 30	– Rede <i>blockchain</i> , <i>workflow</i> científico colaborativo.	65
Figura 31	– Fluxos de ações para criação de ambiente colaborativo da arquitetura BlockFlow.	67

Figura 32	– Interface do usuário, para a escolha entre redes locais ou redes na nuvem, na arquitetura BlockFlow.	68
Figura 33	– Tela do FrontEnd, para criar redes <i>blockchains</i> na arquitetura BlockFlow.	69
Figura 34	– Tela do FrontEnd, para especificar quais pesquisadores serão nós pares e farão parte do canal na rede <i>blockchain</i> , na arquitetura BlockFlow. . . .	70
Figura 35	– Sumarização do passo a passo a partir do FrontEnd da BlockFlow. . .	70
Figura 35	– Fragmento <i>Chaincode</i> da BlockFlow.	72
Figura 36	– Screenshot de cada instância de máquina virtual na nuvem, rondado o ambiente colaborativo.	81
Figura 37	– <i>Workflow</i> SciPhy (a) e <i>Workflow</i> ViReport (b) executados no experimento.	81
Figura 38	– Interface do usuário, para que pesquisadores possam criar redes colaborativas de experimentação científica utilizando a arquitetura BlockFlow	85
Figura 39	– Interface do usuário, com todos os componentes da rede <i>blockchain</i> onde são especificadas as configurações, de cada PEERS, CAS, Orderes.	86
Figura 40	– Interface do usuário, para que os pesquisadores possam (i) iniciar <i>peers</i> , (ii) criar canais, (iii) criar identidades, (iv) instalar <i>chaincode</i> , (v) instanciar <i>chaincode</i>	87
Figura 41	– <i>Workflow</i> Sciphy instrumentalizado com serviço da <i>web</i> da BlockFlow.	88
Figura 42	– <i>Workflow</i> Vireport instrumentalizado com serviço da <i>web</i> da BlockFlow.	88
Figura 43	– Tela de cadastros dos <i>Workflows</i> (a) Sciphy (b) ViReport.	89
Figura 44	– Tela com cadastros dos <i>Workflows</i> Sciphy, ViReport.	90
Figura 45	– Tela <i>FrontEnd</i> de <i>upload</i> de arquivos de entrada e de saída dos <i>workflows</i> .	91
Figura 46	– Tela <i>FrontEnd</i> com arquivos armazenados.	92
Figura 47	– Árvore filogenética com base nas sequências de 25 genomas completo de coronavírus, incluindo SARS-CoV-2, SARS-CoV, HCoV, morcego SARS, SARS-like CoV e MERS-CoV.	93
Figura 48	– . Árvore filogenética com base nas sequências de 61 genomas completo de coronavírus, incluindo SARS-CoV-2, SARS-CoV, HCoV, morcego SARS, SARS-like CoV e MERS-CoV.	94
Figura 49	– Interface de usuário com a proveniência coletada durante a execução do experimento.	96
Figura 50	– Interface de usuário para executar query(s).	97
Figura 51	– Interface de usuário para executar a consulta (Q1).	98
Figura 52	– Resultado em formato JSON da consulta (Q1).	99
Figura 53	– Interface de usuário para executar a consulta (Q2).	100
Figura 54	– Resultado em formato JSON da consulta (Q2).	101
Figura 55	– Interface de usuário para executar a consulta (Q3).	102
Figura 56	– Resultado em formato JSON da consulta (Q3).	103
Figura 57	– Dowloand dados formato JSON.	104

Figura 58	– Pesquisa em Cypher.	104
Figura 59	– Consulta <i>entity(s) workflow</i>	105
Figura 60	– Comparação objeto de pesquisa (dados) usados ou gerados em um experimento.	105

LISTA DE TABELAS

Tabela 1 – Descrição de elementos de um bloco blockchain	25
Tabela 2 – Mecanismo de Consenso.	26
Tabela 3 – Principais Tipos de Redes de <i>Blockchain</i>	28
Tabela 4 – String de Busca Genérica.	39
Tabela 5 – Bases de Busca Utilizadas.	39
Tabela 6 – Total de Artigos Retornados em Cada Base utilizada.	41
Tabela 7 – Comparação entre a BlockFlow e as Propostas Encontradas na Literatura.	53
Tabela 8 – Correspondência entre o mapeamento de um conjunto de tarefas para o Modelo ProvONE	61
Tabela 9 – Taxa de transferência da transação	73
Tabela 10 – Latência de transações	73
Tabela 11 – Taxa de envio	73
Tabela 12 – Configuração Máquinas Virtuais.	79
Tabela 13 – Software instalados nas máquinas virtuais.	80
Tabela 14 – Consultas.	94

LISTA DE ABREVIATURAS E SIGLAS

AMS	Alinhamento Múltiplo de Sequência
API	Application Programming Interface
AWS	Amazon Web Service
DNA	Deoxyribonucleic Acid
DSR	Design Science Research
EC2	Amazon Elastic Compute Cloud
ECOSC	Ecosistema de Software Científico
E-SECO	e-Science Ecosystem
GCP	Google Cloud Platform
HTTP	Hypertext Transfer Protocol
IaaS	Infraestrutura como serviço
JSON	JavaScript Object Notation
LPSC	Linha de Produtos de Software Científico
PaaS	Plataforma como serviço
PBFT	Byzantine fault-tolerant
PoS	Proof-of-Stake
PoW	Proof-of-Work
PRIME	PRagmatic Interoperability to MEaningful collaboration
REST	REpresentational State Transfer
RNA	Ribonucleic Acid
SaaS	Software como Serviço
SWfMSs	Sistemas de Gerenciamento de Workflows Científicos

SUMÁRIO

1	INTRODUÇÃO	15
1.1	CONTEXTUALIZAÇÃO	15
1.2	MOTIVAÇÃO	16
1.3	OBJETIVOS	18
1.4	QUESTÃO DE PESQUISA	18
1.5	ORGANIZAÇÃO	19
2	FUNDAMENTAÇÃO TEÓRICA	20
2.1	E-SCIENCE	20
2.2	E-SECO	21
2.3	BLOCKCHAIN	23
2.3.1	Transação	24
2.3.2	Mecanismo de Consenso	25
2.3.3	Tipos de Rede Blockchain	27
2.3.4	<i>Smart Contract</i>	28
2.3.5	Chaves	28
2.3.6	<i>Hyperledger Fabric</i>	30
2.4	PROVENIÊNCIA	31
2.4.1	Tipos de Proveniência	31
2.4.2	Captura de Proveniência	32
2.4.3	Modelos de Proveniência	32
2.5	CLOUD COMPUTING	35
2.6	MAPEAMENTO SISTEMÁTICO DA LITERATURA	37
2.6.1	Planejamento	37
<i>2.6.1.1</i>	<i>Questões de Pesquisa</i>	<i>37</i>
<i>2.6.1.2</i>	<i>PICOC</i>	<i>38</i>
<i>2.6.1.3</i>	<i>String de Busca</i>	<i>38</i>
<i>2.6.1.4</i>	<i>Fonte de Busca</i>	<i>39</i>
<i>2.6.1.5</i>	<i>Critério de Inclusão e Exclusão</i>	<i>39</i>
2.6.2	Condução	40
<i>2.6.2.1</i>	<i>Relatos dos Resultados</i>	<i>42</i>
2.6.3	Análise dos trabalhos	49
2.7	DISCUSSÕES	51
3	ARQUITETURA BLOCKFLOW	54
3.1	DEFINIÇÃO METODOLÓGICA	54
3.2	COMPONENTES DA ARQUITETURA BLOCKFLOW	55
3.2.1	Camada <i>API RESTful Webservice</i>	57
3.2.2	Camada <i>Wrapper</i>	58

3.2.3	Camada <i>Model</i>	61
3.2.4	Camada <i>Client</i>	64
3.2.5	<i>Camada Blockchain Network</i>	65
3.2.6	Tecnologias de Desenvolvimento	66
3.3	BLOCKFLOW EM AÇÃO	66
3.3.1	Análise de Desempenho	71
3.4	DISCUSSÕES	74
4	AVALIAÇÃO DA ARQUITETURA BLOCKFLOW	75
4.1	INTRODUÇÃO	75
4.2	CONTEXTUALIZAÇÃO	76
4.3	SARS-CoV2	77
4.4	PROVA DE CONCEITO	78
4.5	PLANEJAMENTO	79
4.5.1	Configuração do Ambiente	79
4.5.2	<i>Workflows</i> utilizados na PoC	80
4.5.3	Cenário	83
4.6	EXECUÇÃO	84
4.6.1	Coleta e Armazenamento de dados de proveniência	86
4.6.2	Análise e Consultas de dados de Proveniência	91
4.6.3	Discussões	95
5	CONSIDERAÇÕES FINAIS	107
	REFERÊNCIAS	109

1 INTRODUÇÃO

Este capítulo apresenta a motivação para este trabalho, assim como os objetivos e sua organização.

1.1 CONTEXTUALIZAÇÃO

A colaboração científica apresenta desafios e oportunidades importantes para a comunidade científica. O avanço da ciência moderna depende cada vez mais da interação entre cientistas e do uso de uma inteligência coletiva. Neste cenário de colaboração científica, com a interação entre indivíduos geograficamente distribuídos, o compartilhamento de dados, a troca de ideias e de resultados, são fundamentais para promover o conhecimento e acelerar o desenvolvimento da ciência (WAGNER, 2018).

Nesse sentido, cientistas estão sendo cada vez mais impulsionados a colaborar e compartilhar informações com outros membros da comunidade, bem como a reutilizar dados de seus pares (AMBRÓSIO et al., 2018a; BELLOUM et al., 2011; CLASSE et al., 2016; JANDRE; DIIRR; BRAGANHOLO, 2020; TENOPIR et al., 2015). Por outro lado, na última década, o paradigma da ciência orientada a dados tornou-se uma realidade amplamente difundida (HEY et al., 2009; HEY; TREFETHEN, 2020; HIMANEN et al., 2019) e fenômenos complexos passaram a ser simulados por supercomputadores através de ferramentas computacionais que exigem cada vez mais processamento e análise de grandes quantidades de dados (DE OLIVEIRA; LIU; PACITTI, 2019; HEIDSIECK et al., 2020; LIU et al., 2015). Gerenciar e integrar esses projetos científicos orientados a dados é uma tarefa complexa. Assim, esses experimentos científicos são geralmente representados como *workflows* científicos (*Scientific Workflows* - SWfs) que facilitam a modelagem, o gerenciamento, a execução de atividades e expressem facilmente todas as etapas de processamento de dados (*pipeline*) e suas dependências (MATTOSSO et al., 2010).

Devido à complexidade e a necessidade de processar grandes volumes de dados, esses *workflows* científicos comumente dependem de um conjunto de recursos especiais, tais como *hardware*, especializados e um conjunto de *softwares* para a execução de suas atividades complexas. Além disso, podem exigir um ambiente distribuído, colaborativo ou de alto desempenho (*High-Performance Computing* - HPC), como *grids* ou computação em nuvem (ZHAO et al., 2011). Com característica de elasticidade, *pool* de recursos e pagamento por uso, os ambientes em nuvens, tem sido cada vez mais adotados (DE OLIVEIRA; LIU; PACITTI, 2019; ZHAO et al., 2011). Em (DE OLIVEIRA et al., 2010) e (TERZO; MOSSUCCA, 2017) os autores descrevem as principais características dos ambientes de nuvem de acordo com uma perspectiva de *e-Science*. Uma vantagem importante fornecida pelas nuvens é que os detalhes de implementação ou configuração são abstraídos do usuário. Assim, cientistas constroem e executam seus experimentos com aplicações mais robustas,

sem a necessidade de se preocupar com detalhes de implementação, infraestrutura ou configuração.

A reprodutibilidade é uma característica importante para *workflows* científicos (SWfs) orientada a dados (CHIRIGATI et al., 2016; COHEN-BOULAKIA et al., 2017; POUCHARD, 2019; SANTANA-PEREZ; PÉREZ-HERNÁNDEZ, 2015). Um experimento só é considerado válido se puder ser reproduzido. No entanto, existe uma crise de credibilidade e reprodutibilidade na ciência (BAKER, 2016a; FANELLI, 2018; FRASER et al., 2018; MAKEL; PLUCKER; HEGARTY, 2012; BEGLEY; ELLIS, 2015; GEORGE; BUYSE, 2015; MIYAKAWA, 2020; PENG, 2015; PRINZ; SCHLANGE; ASADULLAH, 2011). Uma pesquisa realizada pela revista *Nature* com mais de 1.576, pesquisadores, mostrou que mais de 70% dos pesquisadores tentaram e falharam em reproduzir os experimentos de outros cientistas, e mais da metade não conseguiram reproduzir seus próprios experimentos (BAKER, 2016b). Da mesma forma, pesquisadores da Bayer, tentaram replicar 67 estudos e foram capazes de reproduzir apenas 24 deles (PRINZ; SCHLANGE; ASADULLAH, 2011). Em uma auditoria de rotina dos ensaios clínicos de leucemia realizados pelo Grupo B de Câncer e Leucemia, um dos grupos de estudos clínicos multicêntricos de câncer, patrocinados pelo *National Cancer Institute* (NIH, USA), relatou uma incidência de fraude de 0,25, dos ensaios (GEORGE; BUYSE, 2015).

1.2 MOTIVAÇÃO

De acordo com Chirigati et al. (2016), uma ciência de qualidade requer reprodutibilidade, não apenas para encontrar fraudes, mas também para apoiar a reutilização de experimentos científicos. Toda nova descoberta científica é construída através de um processo iterativo, com base em conhecimento já existente. Portanto, se não podemos reproduzir conhecimento já existente, estamos desperdiçando muito esforço, recursos e tempo refazendo experimentos ou parte deles que poderiam ser reutilizados. Dessa forma, um aspecto crítico associado a um *workflow* científico são seus dados de proveniência, que pode ser definida como a origem ou linhagem dos dados que auxiliam na compreensão dos resultados do experimento científico (DAVIDSON; FREIRE, 2008).

A importância da proveniência na pesquisa computacional reproduzível está bem documentada na literatura (FREIRE; CHIRIGATI, 2018; MISSIER, 2016; SILVA; FREIRE; CALLAHAN, 2007). Em experimentos colaborativos *in silico* é importante o uso de proveniência para auxiliar os pesquisadores a analisarem a qualidade, verificarem a autoria e reproduzirem os resultados alcançados. Nessa perspectiva, dados de proveniência sobre os quais descobertas científicas se baseiam, devem ser confiáveis e sua veracidade deve ser mantida. No entanto, a falta de mecanismos eficazes para proteger a integridade de dados pode levar a controvérsias ou fraudes científicas (BIK; CASADEVALL; FANG, 2016; MIYAKAWA, 2020). Assim, confiança e transparência no compartilhamento de informações

entre pesquisadores é um desafio, incluindo o reaproveitamento de conhecimento adquirido em experimentos produzidos por terceiros. Dessa forma, em um ambiente científico colaborativo, existem vários desafios, no que tange a dados de proveniência compartilhados, tais como confidencialidade, transparência e interoperabilidade.

Dados de proveniência relacionado a um experimento científico é considerado propriedade intelectual (DE OLIVEIRA et al., 2010, BHUYAN et al., 2019). Assim, a confidencialidade, ao se realizar experimentos colaborativos é um aspecto importante, somente pessoas devidamente autorizadas podem compartilhar ou visualizar resultados até que estes, sejam publicados. A transparência é garantia de que os pesquisadores terão confiança na condução do experimento colaborativo. Assim, todas as atualizações de dados devem ser rastreadas, devendo ser verificado como os dados foram criados ao longo do tempo. Além disso, a interoperabilidade dos dados é fundamental considerando que a execução de experimentos científicos é realizada por cientistas em ambientes distribuídos e heterogêneos. A falta de suporte na integração e interoperabilidade de dados dificulta o compartilhamento de informações, dificultando o compartilhamento de conhecimento.

Para ambientes científicos colaborativos, onde confiança é um requisito importante considerando tanto o processo de experimentação, seus resultados e reprodutibilidade, os sistemas baseados em *blockchain* (NAKAMOTO, 2008) podem ser uma alternativa para o suporte as atividades colaborativas e distribuídas, oferecendo confiança mútua. *Blockchain* é um *ledger* disruptivo, distribuído e imutável sobre uma rede ponto-a-ponto, onde os dados podem ser gerenciados e organizados de uma maneira, aberta, permanente e transparente sem a necessidade de um terceiro confiável (TSCHORSCH; SCHEUERMANN, 2016). Para o domínio da *e-Science*, o *blockchain* tem o potencial de aprimorar a colaboração, confiança, interoperabilidade, rastreabilidade e auditabilidade. Sob essa perspectiva, existem vários estudos na literatura que enfatizam a importância da proteção de proveniência em pesquisas científicas e apontam a potencial da aplicabilidade de *blockchain* como um facilitador para a criação de plataformas científicas. Em (COELHO et al., 2020; COELHO et al., 2021; KARASTOYANOVA; STAGE, 2018; VAN ROSSUM, 2017) os autores abordam o uso de *blockchain* com o objetivo de melhorar a colaboração, a reprodutibilidade e a confiança em dados de proveniência no contexto de *e-Science*.

Experimentos complexos envolvem interações entre pesquisadores distribuídos geograficamente. Devemos considerar aspectos como o uso de grandes quantidades de dados e a necessidade de contar com recursos e serviços de computação distribuída. Além disso, os experimentos requerem relacionamentos intensos entre recursos e aplicativos que suportam o *workflow* científico. Nesse contexto, as instituições científicas abriram suas fronteiras para colaborar com parceiros externos, surgindo um novo conceito de desenvolvimento científico. Este conceito abrange várias soluções de *software*, instituições científicas e desenvolvedores de *software* científico que podem aderir a uma plataforma compartilhada, denominada de Ecossistema de *Software* Científico (SSECO) (BOSCH,

2009; FREITAS et al., 2015).

Para apoiar a colaboração e interação entre parceiros científicos distribuídos geograficamente, a plataforma E-SECO (*E-Science Software Ecosystem*) foi especificada (FREITAS et al., 2015). A plataforma E-SECO gerencia todas as etapas do ciclo de vida da experimentação científica colaborativa e a captura de dados de proveniência, por meio do suporte de uma rede ponto a ponto. Cada nó da rede possui um repositório de dados E-SECO, armazenando dados de forma descentralizada. No entanto, embora o repositório de dados do E-SECO seja descentralizado e compartilhado entre seus usuários, ele não possui um mecanismo que ofereça confiança tanto para dados de proveniência compartilhados quanto para o processo de colaboração científica.

1.3 OBJETIVOS

Este trabalho apresenta uma arquitetura baseada em *blockchain*, denominada BlockFlow, cujo objetivo é apoiar a confiabilidade, transparência, privacidade, interoperabilidade e reprodutibilidade na pesquisa colaborativa no contexto da plataforma E-SECO. A BlockFlow tem como foco prover mecanismos que tragam maior confiabilidade aos dados e processos em *workflows* científicos colaborativos. O objetivo é permitir que cientistas trabalhem de maneira colaborativa e distribuída, compartilhando dados de proveniência de uma maneira mais confiável, com intuito de garantir a reprodutibilidade dos resultados obtidos. Além disso, este trabalho também tem como objetivo apoiar a execução de *workflows* intensivos em dados, ancorados pelo paradigma de computação em nuvem, através de infraestruturas de *cloud*.

Através de exemplos e cenários de aplicação, discutimos a viabilidade da proposta em apoiar sistemas que necessitam de interoperabilidade e reutilização dos resultados de *workflows* científicos, integrando dados de proveniência de diferentes Sistemas de Gerenciamento de *Workflows* Científico (*Scientific Workflow Management System* - SWfMS) e, por sua vez, aumentando a eficiência na pesquisa colaborativa.

1.4 QUESTÃO DE PESQUISA

Considerando o exposto acima, a seguinte questão de pesquisa é investigada neste trabalho: **Como a arquitetura BlockFlow pode auxiliar cientistas nos experimentos científicos colaborativos, oferecendo um ambiente confiável apoiando a interoperabilidade, privacidade, transparência e reprodutibilidade de *workflows* científicos?**

1.5 ORGANIZAÇÃO

Este trabalho está dividido em cinco capítulos, além desta introdução. O Capítulo 2 apresenta os principais conceitos relacionados à solução proposta e trabalhos relacionados. O Capítulo 3 apresenta a solução proposta bem como descreve seu uso para apoiar experimentos científicos colaborativos de uma maneira mais confiável, detalhando os aspectos conceituais e a implementação da solução. O Capítulo 4 apresenta a avaliação da solução proposta. O Capítulo 5 apresenta as considerações finais, discutindo as contribuições do trabalho, suas limitações e os trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são descritos os principais conceitos relacionados à proposta deste trabalho. Considerações sobre *e-Science*, *blockchain*, dados de proveniência e computação em nuvem são apresentadas a fim de embasar a abordagem proposta. Também são discutidos a plataforma *E-SECO* e os trabalhos relacionados à solução proposta nesta dissertação.

2.1 E-SCIENCE

Nas últimas décadas, aliada ao desenvolvimento de inovações tecnológicas, a ciência passou a explorar novas possibilidades de experimentação científica (OGASAWARA et al., 2008), e umas das inovações foi o uso intensivo de recursos computacionais. Neste contexto, surge o termo e-ciência ou *e-Science*, definido por (HEY et al., 2009) como uma colaboração global de áreas-chave da ciência junto com a geração de uma infraestrutura computacional capaz de suportá-la.

Neste contexto, os *workflows* científicos (SWfs) têm se tornado um padrão para representar experimentos científicos baseados em simulações computacionais. *Workflows* científicos (SWfs) podem ser entendidos como a dinâmica para representar e executar um fluxo de atividades correlatas, uma sequência lógica de invocações de programas e/ou serviços (i.e., atividades) no contexto de um experimento *in silico* (MATTOSO et al., 2010). No entanto, a modelagem de *workflows* científicos e sua representação não são tarefas triviais e abordagens *adhocs*, ou seja, sem o uso de sistemas de gerenciamento de *workflows*, podem criar barreiras que dificultam as atividades de um experimento. Desta maneira, *workflows* científicos são comumente executados por Sistemas de Gerenciamento de *Workflows* Científicos (SWfMS).

Os SWfMSs permitem que um cientista especifique um experimento científico como um conjunto de tarefas a serem processadas pelo computador. O encadeamento destas tarefas de maneira organizada deriva o modelo do *workflow*. Estas tarefas comumente realizadas em um experimento se relacionam com coleta, homogeneização, filtragem e análise de dados. Existem vários SWfMSs com características e comportamentos distintos, como o VisTrails (CALLAHAN et al., 2006), Taverna (MISSIER et al., 2010), Swift/T (WOZNIAK et al., 2013), Kepler (LUDÄSCHER et al., 2006), Pegasus (DEELMAN et al., 2007), Chiron (OGASAWARA et al., 2013) e Galaxy (GOECKS et al., 2010), entre outros.

Geralmente, um cientista define um *workflow*, a partir de um SWfMS, usando um modelo gráfico, de compreensão bastante intuitiva. A partir desse modelo, o SWfMS é capaz de executar o experimento de forma automática, com pouca ou nenhuma intervenção do cientista, utilizando, para isso, a infraestrutura computacional disponível. Assim, a *e-Science* pressupõe a construção de uma infraestrutura computacional de uso distribuído, capaz de permitir a colaboração entre cientistas, envolvendo o uso intensivo e

compartilhamento de dados, muitas vezes heterogêneos (HEY et al., 2009) e a execução de experimentos a partir de *workflows* científico (SWfs), executados em SWfMS. Nesse novo cenário, existe um esforço crescente para apoiar pesquisadores, na comunicação, na troca de ideias, e na disseminação do conhecimento (VAN ROSSUM, 2017).

Na comunidade científica, a importância da colaboração e do compartilhamento de dados entre os pesquisadores se mostra essencial para apoiar o avanço científico (TENOPIR et al., 2015) e se apoia na reprodutibilidade dos resultados e compartilhamento destes e dos processos que os geraram. Nesse sentido, com objetivo de promover o conhecimento e acelerar o desenvolvimento da ciência, atualmente é comum a criação de redes colaborativas entre grupos de pesquisadores geograficamente distribuídos.

No entanto, neste contexto, onde existem várias partes envolvidas e o compartilhamento de resultados, a confiança é crucial. A falta de confiança e transparência no compartilhamento de informações entre pesquisadores é um desafio, incluindo a reutilização do conhecimento. A capacidade de reproduzir experimentos está no cerne da ciência (BEGLEY; IOANNIDIS, 2015; CHIRIGATI et al., 2016; COHEN-BOULAKIA et al., 2017; MCNUTT, 2014), e parte da necessidade de maior transparência na pesquisa. Nesse sentido, para a troca de informações em ambientes científicos colaborativos, onde a confiança é um requisito importante, sistemas baseados em *blockchain* (KARASTOYANOVA; STAGE, 2018; VAN ROSSUM, 2017) podem ser uma alternativa.

2.2 E-SECO

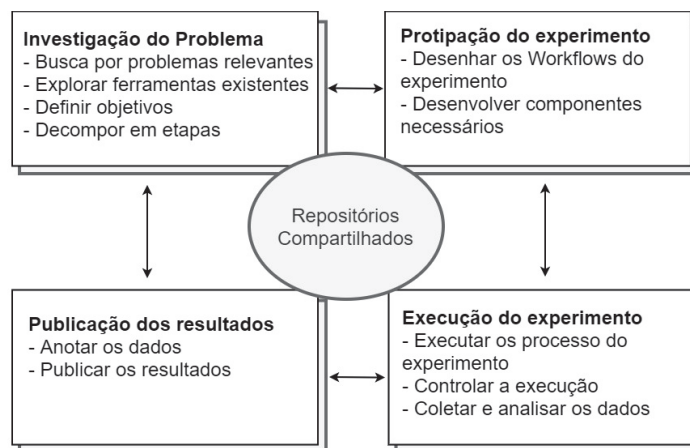
Conforme ressaltado anteriormente, experimentos científicos estão convergindo para ambientes computacionais, onde cientistas trabalham em colaboração, compartilham dados e aplicações científicas (BELLOUM et al., 2011; MISSIER et al., 2010; ZHANG, Jia; KUC; LU, 2012). Nesse cenário, onde a ciência moderna exige cada vez mais interação e colaboração entre pesquisadores, uma infraestrutura de *e-Science* precisa de além de fornecer um ambiente capaz de gerenciar grandes quantidades de dados, tratar de maneira adequada a heterogeneidade e reprodutibilidade.

Para tratar a colaboração e apoiar todo o processo de experimentação científica, a plataforma E-SECO (*e-Science Ecosystem*) (FREITAS et al., 2015) foi especificada. Esta plataforma é baseada nos conceitos de Ecossistema de *Software* (ECOS) (MANIKAS, 2016) que pode ser definido como a interação de um conjunto de atores sobre uma plataforma tecnológica comum, tendo como resultados, soluções ou serviços de *software*. Em *e-Science*, (FREITAS et al., 2015) caracterizam um ECOSs, como as relações entre fornecedores de *software* científico, institutos de pesquisa, pesquisadores, órgãos de fomento, instituições financiadoras, e as partes interessadas nos resultados de pesquisa.

A plataforma E-SECO foi projetada para apoiar todo o ciclo de vida de experimentos científicos conforme sugerido por (BELLOUM et al., 2011) e apresentado na Figura 1,

sendo composto por Investigação do Problema, Prototipação do Experimento, Execução do Experimento e Publicação dos Resultados.

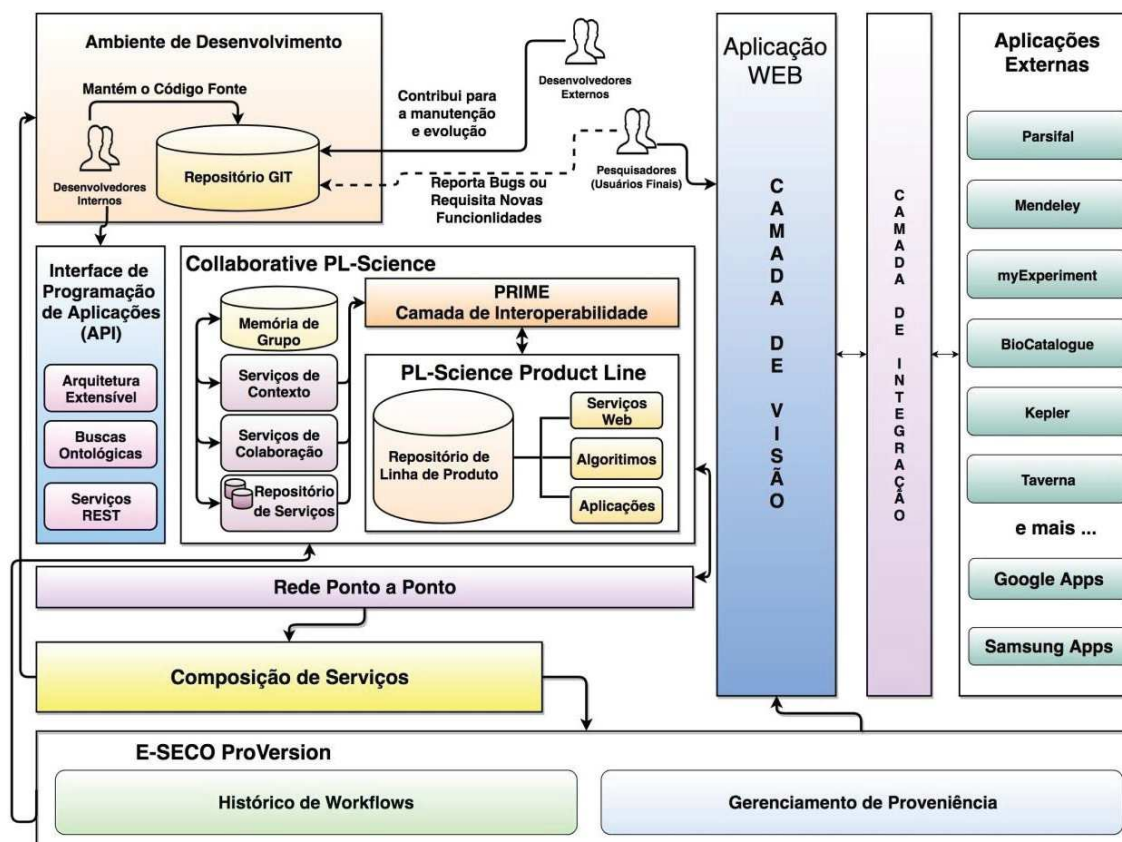
Figura 1 - Ciclo de vida de um experimento científico.



Fonte: (BELLOUM et al., 2011).

Para isso, a plataforma E-SECO é composta por diferentes componentes, conforme apresentado na Figura 2, englobando o **Núcleo**, composto pelo módulo *Collaborative PL-Science* (PEREIRA et al., 2016), responsável por todas as atividades associadas a uma Linha de Produto de Software Científico (LPSC), e pela camada de Interoperabilidade **PRIME** (*PRagmatic Interoperability to MEaningful collaboration*) (NEIVA et al., 2015), desenvolvida para apoiar a interoperabilidade nas atividades de colaboração nos diferentes níveis: sintático, semântico e pragmático. A **Rede ponto a ponto** (P2P), permite que cada instância que utiliza a plataforma seja um nó e funcione tanto como cliente quanto servidor, podendo assim, compartilhar e armazenar dados e serviços, de maneira descentralizada. O **Ambiente de Desenvolvimento** permite que desenvolvedores internos e externos possam propor melhorias e desenvolver novas funcionalidades para a plataforma. O **Módulo de Composição de Serviços** oferece recursos para a composição de serviços internos e externos na plataforma, possibilitando a reutilização, a interoperabilidade e a extensibilidade de serviços científicos (MARQUES et al., 2017). A **Camada de Integração** possui clientes para as APIs que podem ser utilizados e estendidos por desenvolvedores. É através da camada de integração que a solução proposta nesta dissertação se integra a plataforma E-SECO. O **Módulo de proveniência** denominada *E-SECO ProVersion* (SIRQUEIRA et al., 2016), suporta a captura e análise da proveniência durante a modelagem e execução de *workflows* científicos. Esse módulo foi estendido por (AMBRÓSIO, 2018b) para abranger todo o ciclo de vida de dados de proveniência, gerenciamento de contexto e apoio ao reuso dos experimentos científicos. A **Camada de Visualização**, por sua vez, fornece um ambiente, através de uma *interface web*, para o apoio a visualização de experimentos.

Figura 2 - Visão Geral da Plataforma E-SECO.



Fonte: (AMBRÓSIO, 2018b)

Embora a plataforma E-SECO apoie todo ciclo de experimentação científica, coletando e armazenando proveniência, não possui mecanismos para assegurar a confiabilidade dos dados de experimentos científicos colaborativos e distribuídos. Uma das tecnologias que pode auxiliar nesse problema é a tecnologia de *blockchain* (NAKAMOTO, 2008).

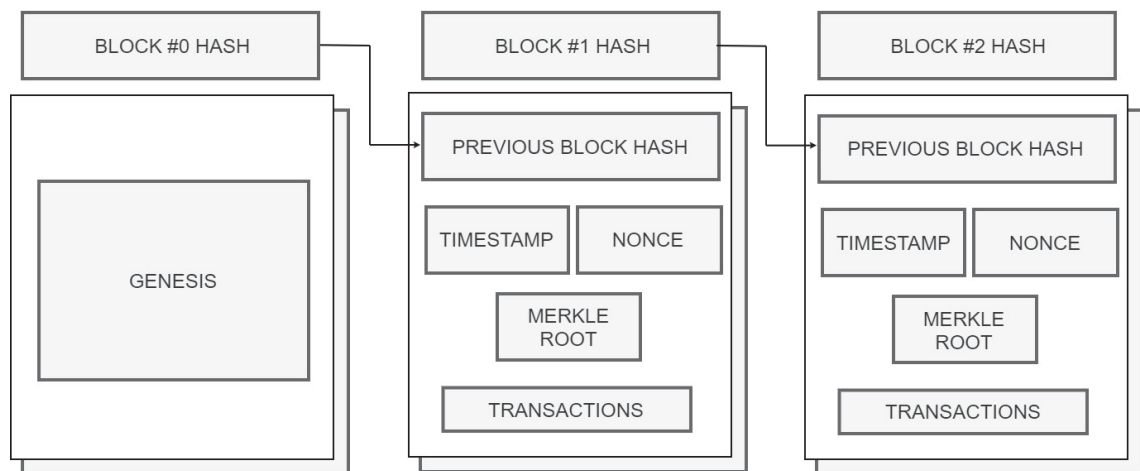
2.3 BLOCKCHAIN

Proposta por Satoshi Nakamoto, no artigo “*Bitcoin: A Peer-to-Peer Electronic Cash System*” (NAKAMOTO, 2008), a tecnologia *blockchain* tornou-se amplamente conhecida como a estrutura de dados (*ledger*) que sustenta o *Bitcoin* (TSCHORSCH; SCHEUERMANN, 2016). O *Bitcoin* é uma ferramenta para processamento de pagamentos eletrônicos distribuídos. A principal ideia por trás da tecnologia *blockchain*, conhecida como livro-razão inviolável, é um sistema criptográfico ponto-a-ponto (P2P) que resolve problemas de gasto duplo em moedas virtuais e que ainda elimina a necessidade de um terceiro confiável (NAKAMOTO, 2008).

Blockchain pode ser definido como um livro imutável, descentralizado e compartilhado, que mantém uma sequência de blocos cronológicos, criptografados, e conectados

entre si, sobre uma rede ponto-a-ponto (P2P) (FANNING; CENTERS, 2016; XU; WEBER; STAPLES, 2019). Na *blockchain*, os blocos formam uma cadeia, ou seja, uma sequência linear que possibilita a auditoria e rastreabilidade de informações. O primeiro bloco das cadeias *blockchains* são conhecidos como blocos *Genesis*. Cada bloco contém, i) um valor de *hash* exclusivo; ii) o *hash* do bloco anterior (estabelecendo um vínculo e uma ordenação relativa entre os blocos); iii) uma lista de transações (derivadas dos nós participantes da rede); iv) o carimbo de data e hora; v) a raiz da árvore *merkle*; e vi) o *nonce*. Na *blockchain*, esses blocos de dados são legíveis, ou seja, transparentes para todos os participantes, graváveis por todos e invioláveis. A estrutura de um bloco, pode ser definida conforme a Figura 3. A descrição dos elementos do bloco é detalhada na Tabela 1.

Figura 3 - Estrutura e encadeamento de blocos *blockchain*.



Fonte: Elaborada pelo autor.

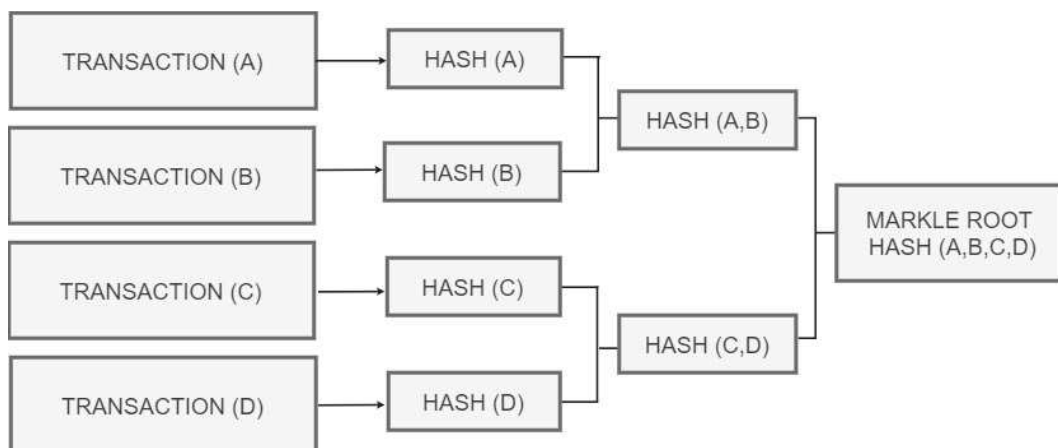
2.3.1 Transação

O *blockchain* registra entradas de dados de maneira descentralizada e permite que entidades possam interagir umas com as outras sem a presença de um terceiro confiável. Essa comunicação é feita através de transações e reflete a semântica da aplicação, podendo ser qualquer tipo de informação, seja moeda, dados científicos, ou outros. As transações são armazenadas e ordenadas em árvores *merkle*, conforme a Figura 4, nos blocos da *blockchain*.

Uma árvore *merkle*, também conhecida como uma árvore de *hash* binária, é uma estrutura de dados em que as entradas, conjuntos de transações, são alocadas nos nós folhas (nós filhos), até que a raiz seja alcançada. De uma forma geral, as árvores *merkle* são criadas repetidamente por *hash* de transações até que haja apenas um *hash* restante (esse *hash* é chamado de *Root Hash* ou *Merkle Root*). As árvores *merkles* no *blockchain*

Tabela 1 – Descrição de elementos de um bloco blockchain

Campo	Descrição
<i>Previous Block Hash</i>	É o valor do <i>hash</i> do bloco anterior. Todas as informações de um bloco são inseridas em uma função <i>hash</i> segura. Ao se obter este valor, este é atribuído ao campo <i>Previous Block Hash</i> do bloco posterior. Com essa estrutura baseada em encadeamento de <i>hashes</i> , o <i>blockchain</i> garante a integridade de informações além de sua ordem relativa.
<i>Merkle Tree Root</i>	É o valor de <i>hash</i> correspondente à raiz de uma árvore de <i>Merkle</i> , que é construída a partir de todas as transações incluídas em um bloco.
<i>Timestamp</i>	Data e hora de criação do bloco.
<i>Nonce</i>	Valor arbitrário adicionado ao bloco para dar variabilidade ao valor do <i>hash</i> do bloco.

Figura 4 - Árvore *Merkle*.

Fonte: Elaborada pelo autor.

são utilizadas para resumir e verificar eficientemente a integridade de grandes conjuntos de dados.

2.3.2 Mecanismo de Consenso

Para que cada bloco seja adicionado ao *blockchain*, é necessário um processo de validação conhecido como mecanismo de consenso. Existem vários protocolos de consenso propostos e utilizados dos quais os três mais usados são ilustrados através da Tabela 2.

Tabela 2 – Mecanismo de Consenso.

Protocolos	Descrição
<i>Proof-of-Work</i> (PoW)	<p>É um mecanismo de consenso que, para validar e publicar um bloco na blockchain, requer uma certa quantidade de trabalho computacional (PoW) (WANG et al., 2019; WAN et al., 2020). Esse protocolo de consenso é mais fortemente associado ao <i>blockchain</i> devido à sua integração com o <i>Bitcoin</i> (NAKAMOTO, 2008). Este mecanismo consiste, em encontrar um valor de <i>hash</i> para o bloco. O nó que encontra a solução, recebe incentivos econômicos e é conhecido como minerador. A resolução do PoW é conhecida como mineração. No processo de mineração, mineradores devem realizar muitos cálculos computacionais, e essa operação requer muitos recursos e conseqüentemente gera um alto consumo, principalmente, no que diz respeito à gasto com energia. A energia consumida na mineração do <i>Bitcoin</i> é comparável aos requisitos de eletricidade de um país. Assim, muitas cadeias de blocos que adotam o PoW, estão migrando gradualmente para outros mecanismos de consenso como o PoS. Outra justificativa, para migração para outros mecanismos de consenso, diz respeito a velocidade em que as transações são armazenadas na <i>blockchain</i>.</p>

<i>Proof-of-Stake</i> (PoS)	É um mecanismo de consenso em que a capacidade de verificar e publicar blocos depende da participação, ou seja, da quantidade de moeda nativa de propriedade do nó de mineração (WANG et al., 2019; WAN et al., 2020). Acredita-se que pessoas com mais moedas teriam menos probabilidade de atacar a rede. No entanto, selecionar nós com base no saldo da conta pode ser injusto, pois, mineradoras com saldos maiores podem dominar a rede. Comparar o PoW ao PoS está relacionado à economia de energia e o PoS é mais eficaz.
<i>Byzantine fault-tolerant</i> (PBFT)	É um mecanismo de consenso derivado do Problema dos Generais Bizantinos (LAMPART et al., 1982). No qual mesmo que entre n pares de validação de no máximo $\lfloor \frac{n-1}{3} \rfloor$ mintam ou se comportem arbitrariamente, todos os outros irão executar o código corretamente e assim é possível armazenar informações no <i>blockchain</i> . No PBFT, todos os nós precisam ser conhecidos da rede, o que limita o uso desse protocolo de consenso em uma <i>blockchain</i> pública (WANG et al., 2019; WAN et al., 2020).

2.3.3 Tipos de Rede Blockchain

As redes *blockchain* podem ser classificadas como sem permissão ou autorizadas, conforme apresentado na Tabela 3. Essas classificações são determinadas com base no acesso aos dados do *blockchain*, ou seja, quem pode acessar esses dados, participar ou realizar transações na rede e ainda determina a identidade de seus participantes.

Tabela 3 – Principais Tipos de Redes de *Blockchain*.

Protocolos	Descrição
<i>Permissionless</i> (Sem permissão)	Nesses tipos de instâncias, qualquer pessoa pode ingressar, fazer transações ou sair da rede. As identidades geralmente não são conhecidas. No entanto, qualquer conteúdo publicado pode ser legível por outros membros da rede. Assim, na literatura, existem técnicas criptográficas para <i>blockchain</i> sem permissão, a fim de ocultar informações que exigem privacidade. <i>Bitcoin</i> (NAKAMOTO, 2008) e <i>Ethereum</i> (BUTERIN, 2013) são instâncias de <i>blockchains</i> sem permissão.
<i>Permissioned</i> (Com permissão)	Nas redes com permissão <i>Hyperledger</i> (ANDROULAKI et al., 2018) e R3 Corda (MOHANTY, 2019), a rede é controlada por um grupo de nós conhecidos. Uma autoridade central geralmente decide e atribui o direito a pares individuais de operações de gravação ou leitura de <i>blockchain</i> .

2.3.4 *Smart Contract*

Nick Szabo introduziu este conceito em 1994 e definiu um contrato inteligente, como um “ protocolo de transação informatizado que executa os termos de um contrato ” (SZABO, 1994).

No contexto de *blockchain*, os contratos inteligentes atuam como um aplicativo distribuído e confiável que obtém sua confiança através da rede *blockchain* e do consenso subjacente entre os pares. Como os *smart contracts* residem na cadeia, estes têm um endereço exclusivo, através do qual o usuário final pode endereçar uma transação para ele. Conforme os dados que acionam a condição predefinida, o contrato inteligente é executado automaticamente e de forma independente e de maneira prescrita por todos os pares da rede.

2.3.5 Chaves

Em redes *blockchains*, cada usuário possui um par de chave, pública e privada. Os endereços dessas chaves são baseados em um par de chaves criptográficas de 256 bits

(REID; HARRIGAN, 2013), baseado no algoritmo de criptografia assimétrica, onde as mensagens são criptografadas usando a chave pública e só podem ser descriptografadas usando a chave privada Figura 5.

Na *blockchain*, Figura 5 as chaves privadas são utilizadas para assinar digitalmente as transações aos quais serão transmitidas por toda a rede e as chaves públicas para endereçá-las na rede. O uso de criptografia assimétrica, traz autenticação, integridade e não-repúdio à rede (REID; HARRIGAN, 2013).

Figura 5 - Chaves públicas e privadas *blockchain*.



Fonte: Elaborada pelo autor.

A comunidade científica considera as principais vantagens e características da tecnologia *blockchain* como sendo:

- **Transparência** é obtida no processo de cópia de transações, onde os dados são compartilhados entre os vários nós da rede. Assim, todos os nós da rede podem verificar como a *blockchain* foi criada ao longo do tempo.
- **Descentralização** o *blockchain* é construído sobre uma rede ponto a ponto, para que não haja um ponto único, de falha. No *blockchain* não existe um único sistema de computador que possa ser desligado, censurado ou bloqueado para interromper um serviço. Aplicações são executadas de maneira distribuída, através de confiança entre as partes, sem a necessidade de um entidade intermediária confiável.
- **Confiança** a confiabilidade é alcançada garantindo a integridade dos dados. No *blockchain*, nenhum dado pode ser alterado ou apagado.
- **Imutabilidade** a imutabilidade é alcançada quando o valor de *hash* para cada bloco é gerado e adicionado ao bloco e ao bloco anterior. Sempre que os dados em um bloco são alterados, o valor do *hash* também é alterado. Portanto, qualquer tentativa de fazer alterações em um bloco existente tornará as informações dos blocos subsequentes incorretas, e fará com que todo o *blockchain* seja inválido.

2.3.6 *Hyperledger Fabric*

Considerando todos os benefícios que a tecnologia *blockchain* propõe e para oferecer um ambiente colaborativo, distribuído e confiável de experimentação científica, optamos nesta dissertação por utilizar a tecnologia *blockchain Hyperledger Fabric* (ANDROULAKI et al., 2018).

O *Hyperledger* é um projeto de código aberto mantido pela *Linux Foundation*, que foi concebido com objetivo de apoiar o desenvolvimento colaborativo de um conjunto de ferramentas, *frameworks* e bibliotecas *blockchains*, com foco em melhoria de desempenho e a confiabilidade desses sistemas. Atualmente o *Hyperledger* conta com diferentes *frameworks* tais como o *Hyperledger Fabric*¹, *Swatooth*², *Indy*³ e ferramentas como o *Hyperleger Caliper*⁴ e bibliotecas como a *Hyperledger Ursa*⁵.

O *Hyperledger Fabric* (ANDROULAKI et al., 2018), está sendo desenvolvido ativamente sob Projeto *Hyperleger* da IBM e diferente de outras plataformas, como *Bitcoin* (NAKAMOTO, 2008) e *Ethereum* (BUTERIN, 2013), não possui nenhuma criptomoeda. O acesso à rede é restrito há pessoas autorizadas, caracterizando-a como *permissioned blockchain* e seu mecanismo de consenso é o PBFT (WANG et al., 2019; WAN et al., 2020). A rede do *Hyperledger Fabric* consiste em um conjunto de *peers* (nós pares), geograficamente distribuídos executados em *docker contêineres*⁶. Cada nó mantém o estado do razão e o *log* de transações através do *Apache CouchDB*⁷ ou *LevelDB*⁸. Suas transações são controladas e geradas através de contratos inteligentes que são conhecidos como *chaincodes*. *Chaincode* é um *software* escrito para ler e atualizar o estado do razão. Este *software* no *Hyperledger Fabric* pode ser escrito em diferentes linguagens de programação, como, por exemplo, *Go*, *Java*, *Node.js*.

Para manter a privacidade, a confidencialidade e isolar as atividades apenas entre as partes autorizadas no *Hyperledger Fabric*, é necessário criar um mecanismo de isolamento conhecido como canal. Para realizar transações, os nós participantes precisam se registrar e ter identidades. Os registros de identidade são fornecidos por uma Autoridade de Certificação (CA), que também emite certificados a serem usados para assinar transações. Juntamente com a CA, há outro componente importante para identificação, que é o MSP (*Membership Service Provider*), responsável pelo mapeamento de certificados entre os nós.

Com objetivo de trazer mais confiabilidade para dados científicos, considerando a reprodutibilidade, a solução proposta nesta dissertação apresenta uma arquitetura

¹ <https://www.hyperledger.org/use/fabric>

² <https://www.hyperledger.org/use/sawtooth>

³ <https://www.hyperledger.org/use/hyperledger-indy>

⁴ <https://hyperledger.github.io/caliper/>

⁵ <https://www.hyperledger.org/use/ursa>

⁶ <https://www.docker.com/>

⁷ <https://couchdb.apache.org/>

⁸ <https://dbdb.io/db/leveldb>

baseada em *blockchain* para experimentos científicos colaborativos. Um sistema baseado em *blockchain* pode levar a um ambiente confiável de experimentação científica, permitindo auditoria transparente de todos os dados coletados, processados e acessados por diferentes *workflows*, distribuídos geograficamente, fornecendo assim transparência, imutabilidade e confiabilidade.

2.4 PROVENIÊNCIA

Conforme ressaltado, no contexto de *e-Science*, a reprodutibilidade de experimentos e seus resultados são muito importante. No entanto, para reproduzir e validar experimentos científicos, é necessário ter informações sobre as transformações dos dados desde sua origem até os resultados gerados. Esse tipo de informação é conhecido como dados de proveniência (Freire et al., 2008).

A proveniência ou linhagem dos dados são metadados, que descrevem a origem de um dado ou todos os processos e transformações que os originam (HERSCHEL; DIESTELKÄMPER; LAHMAR, 2017). Mais formalmente, proveniência são os metadados que descrevem entidades, dados, processos, atividades e pessoas envolvidas no processo de criação de um produto (MOREAU et al., 2008). O termo proveniência foi originalmente utilizado no contexto de obras de arte, descrevendo suas propriedades e a localização dos objetos (MOREAU et al., 2013).

Considerando o cenário atual da experimentação científica, onde experimentos são guiados e executadas através de *workflows* científicos colaborativos, a proveniência auxilia os cientistas a terem uma melhor compreensão do experimento, interpretar e compreender resultados e diagnosticar problemas, ao longo de todo o processo científico. Além disso, pode ser utilizada para avaliações sobre a qualidade, confiabilidade e reprodutibilidade, dando credibilidade ao experimento (DAVIDSON; FREIRE, 2008).

2.4.1 Tipos de Proveniência

Existem diferentes tipos de proveniência. Lim et al. (2010) classificam a proveniência, especialmente considerando *workflows* científicos, em dois tipos, proveniência prospectiva e retrospectiva. E em um estudo posterior, de acordo com (KOOP; FREIRE, 2014), foi especificado um terceiro tipo, chamada de proveniência evolutiva. Estes tipos de proveniência são detalhados a seguir.

- **Prospectiva:** captura a estrutura e o contexto estático de um *workflow*, ou seja, expressa as etapas a serem seguidas para gerar um conjunto de dados. É uma especificação das tarefas computacionais que serão executadas no experimento.
- **Retrospectiva:** está associada a informações sobre a execução de um *workflow* ou seja, informações sobre as atividades executadas, i.e., etapas adotadas para derivar

um conjunto de dados. Mais especificamente, é um *log* detalhado da execução de cada tarefa no *workflow*.

- **Evolutiva:** reflete as alterações feitas entre duas versões executadas do *workflow*, ou seja, o histórico da evolução, mantendo todas as alterações aplicadas ao longo de seu ciclo de vida.

2.4.2 Captura de Proveniência

De acordo com Freire et al. (2008) a captura de proveniência pode operar em três níveis, (i) baseado em *workflows*, (ii) baseado em atividades, (iii) e em sistema operacional (SO).

Quando o mecanismo de captura é baseada no *workflow*, um SGWCs é responsável por monitorar e capturar todas as informações de proveniência. No entanto, esse mecanismo possui desvantagem, pois é fortemente acoplado aos SGWCs. Na abordagem baseada em atividades, cada atividade é responsável por capturar sua própria proveniência. Embora tenha a vantagem de ser independente do SWfMS, requer instrumentação das atividades (tarefas) do *workflow*. Mecanismos de coleta de proveniência que funcionam no nível do sistema operacional usam as funcionalidades do sistema operacional para capturar informações de proveniência. A vantagem dessa abordagem é a independência dos SWfMS. No entanto, estes mecanismos não são acoplados em todos os processos do *workflow* e necessitam de processamento posterior com intuito de extrair as relações entre as chamadas do sistema e as tarefas que foram processadas.

Optamos por utilizar, nesta dissertação, a abordagem baseada em atividades, para que a captura de proveniência seja independente do SWfMS, garantindo a independência do formato da proveniência capturada e permitindo a interoperabilidade entre *workflows* científicos distribuídos, capturados a partir de diferentes SWfMS.

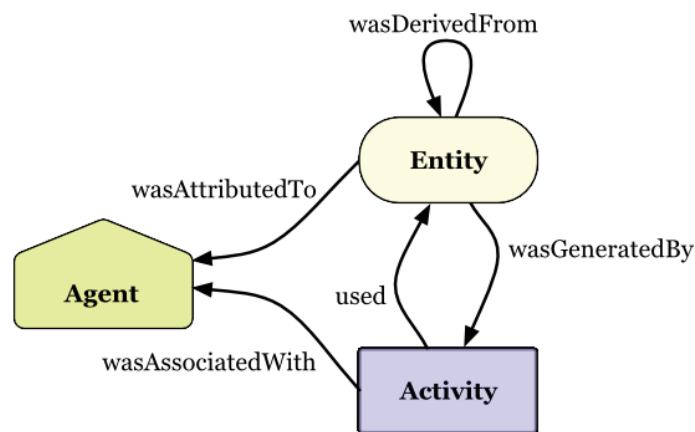
2.4.3 Modelos de Proveniência

Embora haja facilidades oferecidas pelo uso do SWfMS no gerenciamento de um experimento *in silico*, no geral, esses SWfMS capturam dados de proveniência em modelos proprietários. No entanto, quando consideramos experimentos colaborativos, envolvendo múltiplos tipos de SWfMS, a heterogeneidade dos dados de proveniência dificulta o compartilhamento, a cooperação e a reprodutibilidade dos resultados. Dessa forma, vários esforços da comunidade culminaram com o desenvolvimento de modelos genéricos para representar a proveniência e promover a interoperabilidade, dentre estes o OPM (Open Provenance Model) (MOREAU et al., 2008) e o PROV (GROTH; MOREAU, 2013; MISSIER et al., 2013).

O PROV é um modelo de proveniência padrão e recomendado pelo W3C. O PROV surgiu fortemente influenciado pelo modelo OPM, e foi projetado para ser um modelo agnóstico para representar a proveniência. De uma maneira geral, o modelo de proveniência PROV é mais abrangente, mais genérico e seu modelo de dados é capaz de representar transformações e propriedades, de diferentes áreas. Utiliza um grafo para representar as informações de proveniência, a partir dos elementos *Entity* (Entidades), *Activity* (Atividades) e *Agent* (Agentes), e juntamente com os relacionamentos *used*, *wasGeneratedBy*, *wasInformedBy*, *wasAssociatedWith*, *wasDerivedFrom*, *wasAttributedTo* e *actedOnBehalfOf*, formam seu núcleo.

A Figura 6 apresenta as principais construções do modelo.

Figura 6 - Modelo de proveniência PROV.



Fonte: (MISSIER et al; GROTH; MOREAU, 2013).

Entidade (*Entity*) é um tipo físico, digital ou conceitual, entidades podem ser reais (documentos, arquivos, sistemas) ou imaginárias. Atividade (*Activity*) é algo que ocorre durante um período e atua sobre as entidades. Dentre esses podemos incluir uso ou geração de entidades. Agentes (*Agent*) são responsáveis por iniciar uma atividade, para existência de uma entidade ou para a atividade de outro agente.

Os principais relacionamentos são: *used*, relaciona uma entidade a uma atividade que a usou; *wasGeneratedBy*, indica que a entidade foi gerada por uma atividade; *wasInformedBy*, relaciona atividades implicando que uma atividade informada foi gerada pela atividade que a informou, porém, essa atividade é desconhecida ou não é de interesse; *wasAssociatedWith*, relaciona atividades implicando que uma atividade informada foi gerada pela atividade que a informou; *wasDerivedFrom*, relaciona entidades, no sentido de que uma entidade foi originada da outra; *wasAttributedTo*, relaciona um agente a uma entidade a qual ele foi atribuído; *wasAssociatedWith*, relaciona um agente a uma atividade da qual ele tem responsabilidade; *actedOnBehalfOf*, relaciona agentes indicando que um

agente tem autoridade ou responsabilidade por outro - significa que um agente subordinado atuou em nome de agente responsável em uma atividade.

Em experimentos científicos, é necessária a coleta tanto da proveniência retrospectiva, quanto da prospectiva. Porém, embora o modelo PROV seja capaz de representar elementos de proveniência retrospectiva e seus relacionamentos, este modelo contém apenas uma entidade para representar aspectos de proveniência prospectiva. Assim, o PROV foi estendido para melhor representar a captura de proveniência em experimentos científicos, gerando o ProvONE (CUEVAS-VICENTTÍN et al., 2015).

O ProvONE é um modelo de proveniência que estende a recomendação PROV do W3C (CUEVAS-VICENTTÍN et al., 2015), e foi criado especificamente para o contexto de *workflows* científicos. O ProvONE permite a interoperabilidade de proveniência sendo capaz de integrar em um formato comum, informações heterogêneas de múltiplos *workflows* produzidas por diferentes SGWC. Além disso, representa tanto a proveniência prospectiva quanto a proveniência retrospectiva e também a proveniência evolutiva.

O ProvONE é composto por diferentes classes, Figura 7. A classe *Workflow* é uma especialização da classe *Program*, ou seja, representa um tipo especial de *Program*, um experimento computacional, por exemplo. A classe *Program* representa uma tarefa computacional, *Task* (atividade), que consome e produz dados por meio de suas portas (*Ports*). Instâncias da classe *Program* podem ser atômicas ou compostas (ter subprogramas (*hasSubProgram*)), podem ter versões (*wasDerivedFrom*), que descreve a evolução de instâncias das classes *Program* e *Workflow*, e pode ser gerenciada por (*controlledBy*) uma instância da classe *Controller*. Cada *Program* podem também ter um *Plan* (associação com usuário e execução).

A classe *Port* pode ter parâmetros (*hasDefaultParam*) representados por Entidades (*Entity*) - que tem como subclasses (documentos (*Document*), dados (*Data*) e visualizações (*Visualization*)) que podem ser consumidas (*Used*) ou produzidas por (*wasGeneratedBy*) em uma execução (*Execution*). A classe, Dados (*Data*) é definida para ser genérica e representa itens de dados de vários tipos, por exemplo, arquivos XML, JSON, CSV. As visualizações classe, (*Visualization*) são uma classe diferenciada destinada a representar vários itens de visualização, por exemplo, arquivos JPG, PNG, SVG, MP4, geralmente gerados a partir de *workflows*. A classe documento (*Document*) é uma representação genérica de um artigo ou relatório publicado ou não publicado que foi criado como resultado de uma determinada execução de um programa ou *workflow*. Coleções de entidades são representadas por meio da classe *Collection*. Uma coleção pode, por sua vez, representar um conjunto, lista ou outra variante de um grupo de itens. As instâncias da classe (*Port*) dos vários programas (*Program*) são conectadas por canais (*Channel*) pelo relacionamento *connectTo*. A classe *Execution* pode ser vinculada a (*wasAssociatedWith*) a um Usuário (*User*) e uma Porta (*Port*) de entrada (*Usage - hadInPort*) ou de saída (*Generation -*

computacionais, sejam elas dentro de um *cluster* de computadores de alto desempenho, em *grid* ou computação em nuvem.

Para a comunidade científica, o uso do ambiente de computação em nuvem pode ter vantagens e ser atraente de diferentes maneiras. Na literatura, existem vários autores que discutem as vantagens e benefícios do uso da computação em nuvem para executar experimentos científicos (DE OLIVEIRA et al., 2010; KHAN et al., 2019; TERZO; MOSSUCCA, 2017; ZHAO et al., 2011). Com características de alta disponibilidade, mecanismos de tolerância a falhas e principalmente elasticidade (capacidade de provisionamento, de fornecer recursos de *hardware* e *software* sob demanda), os ambientes em nuvem tem sido cada vez mais adotados. Nesse paradigma, cientistas não precisam se preocupar em manter infraestrutura ou configurações, e se precisarem de mais recursos, como, por exemplo (processamento e armazenamento), precisam somente solicitar ao provedor da nuvem. Muitos desses provedores oferecem *interfaces* personalizadas para lidar com os seus recursos como *Amazon Web Services* (AWS), *Microsoft Azure*, *Google Cloud Platform* (GCP) e entre outros. Assim cientistas podem ser concentrar-se apenas na especificação e na execução de seus fluxos de trabalho.

Em (DE OLIVEIRA et al., 2010) e (TERZO; MOSSUCCA, 2017), os autores descrevem as principais características dos ambientes em nuvem de uma perspectiva da *e-Science*. Diante destas necessidades e domínios de aplicação, os ambientes em nuvens podem ser divididos entre três camadas principais: Infraestrutura como serviço (IaaS), Plataforma como serviço (PaaS) e *Software* como serviço (SaaS) (DE OLIVEIRA et al., 2010).

O IaaS é um modelo de computação em nuvem que oferece aos usuários, acesso a um *pool* de recursos computacionais, como servidores, RAM, CPUs, disco rígido, imagem do sistema operacional, armazenamento e redes de maneira escalável. O *Amazon Elastic Compute Cloud* (Amazon EC2) é um exemplo de IaaS. O SaaS é um modelo de computação em nuvem que oferece aos usuários, aplicações (*software*) baseado em *cloud* de um fornecedor por meio de *interfaces* e programas da *web*. O PaaS é um modelo de serviço que oferece aos usuários uma plataforma para criar e executar aplicativos por meio de uma interface de programação fornecida e suportada por provedores de serviços. Ou seja, o PaaS é uma camada incorporada ao IaaS, consistindo de sistemas operacionais e aplicativos intermediários que ajudam desenvolvedores a construir suas aplicações para nuvem, por exemplo, *Google App Engine* e *Microsoft Azure*. Para implementação da proposta apresentada nessa dissertação, optamos por usar o modelo IaaS. Para isso, provisionamos instâncias de máquina virtuais da *Amazon Web Services* EC2⁹.

⁹ <https://aws.amazon.com/pt/>

2.6 MAPEAMENTO SISTEMÁTICO DA LITERATURA

Com o objetivo de investigar e identificar na literatura abordagens que conectam os tópicos da tecnologia *blockchain* como benefícios e mecanismos para o armazenamento e a gerência de dados de proveniência, um Mapeamento Sistemático da Literatura relacionado ao tema foi proposto.

Um Mapeamento Sistemático da Literatura (*Systematic Literature Mapping* - SLM) (KITCHENHAM; CHARTERS, 2007) é um estudo secundário no qual o estado da arte em uma área de pesquisa pode ser determinado, reunindo e resumindo as descobertas existentes sobre a área. Por meio de um SLM é possível classificar pesquisas sobre um tema em um contexto mais amplo (KITCHENHAM; CHARTERS, 2007), no qual é possível identificar estudos primários de modo a responder a questões através de análises e sínteses mais superficiais (WOHLIN et al., 2012). Dessa forma, é possível reunir evidências para identificar lacunas e oportunidades de pesquisa em uma área, alvo, onde podemos verificar quais são as limitações apresentadas em um campo de pesquisa e encontrar soluções para superar essas limitações.

O processo de mapeamento sistemático proposto consiste em três fases: planejamento, condução e relato dos resultados (KITCHENHAM, 2004). Esta metodologia permite a auditoria do estudo e melhora sua confiabilidade. Nesta dissertação, o mapeamento foi organizado com base nas atividades propostas por Kitchenham (2004).

2.6.1 Planejamento

2.6.1.1 Questões de Pesquisa

O objetivo principal desse estudo, é investigar abordagens na literatura que conectam os tópicos da tecnologia *blockchain* com a gerência de dados de proveniência. Partindo deste objetivo, o mapeamento sistemático visa responder algumas questões de pesquisa.

- *MQ1) Como a tecnologia blockchain tem sido utilizada como mecanismo, método e ferramenta para a proveniência?* O objetivo desta MQ é identificar o estado da arte para identificar em quais domínios a tecnologia *blockchain* tem sido utilizada como mecanismo, método ou ferramenta para a proveniência?
- *MQ2) Veículos em que os artigos foram publicados?* O objetivo dessa MQ foi identificar os principais veículos de publicação dos estudos relevantes sobre o tema.
- *MQ3) Qual é a distribuição dos estudos no decorrer dos anos?* O objetivo dessa MQ foi identificar os estudos relevantes sobre o tema e verificar a distribuição desses trabalhos ao longo dos anos.

- *MQ4) Quais foram os métodos de pesquisa?* O objetivo desta MQ é investigar e compreender quais métodos os pesquisadores aplicaram para cada estudo.
- *MQ5) Quais são as vantagens e benefícios obtidos nas abordagens encontradas com a utilização da tecnologia blockchain para a proveniência?* O objetivo desta MQ foi identificar a motivação para as soluções propostas, para a adoção da tecnologia *blockchain*, como mecanismo para gerenciamento e armazenamento de proveniência.
- *MQ6) Quais são os métodos, padrões ou tecnologias mais utilizadas (ou propostas) pelos autores para embasar suas propostas?* Pretendemos avaliar as soluções sob alguns aspectos, (1) plataformas ou arquiteturas de *blockchain* utilizadas (2) mecanismo de consenso.
- *MQ7) Nas abordagens encontradas, quais são os modelos utilizados para representar os dados de proveniência?* O objetivo é avaliar as quais são os modelos mais utilizadas pelas abordagens, i.e., PROV, OPM ou outros.

2.6.1.2 PICOC

Nesta mapeamento sistemático, para a especificação do método de busca utilizou-se a abordagem PICOC (*Population, Intervention, Comparison, Outcome, Context*) (PETTICREW; ROBERTS, 2006). Os elementos que compõem esta diretriz são explorados a seguir: População refere-se ao conjunto de elementos que estamos investigando e do interesse do estudo. A intervenção refere-se ao elemento que aborda o estudo. A comparação procura comparar a intervenção especificada anteriormente. Resultado refere-se aos resultados obtidos incluindo um ponto de vista prático deles. O contexto delimita o contexto em que a intervenção é entregue.

A seguir são apresentados nossos elementos definidos através da diretriz PICOC:

- População (P): Blockchain
- Intervenção (I): Provenance
- Comparação (C): Não se aplica
- Resultado (O): approach, architecture, framework, Infrastructure, Method, Model, Solution, Technique, Tool, Platform, Process, Software.
- Contexto (C): não se aplica

2.6.1.3 String de Busca

A construção da *string* de consulta seguiu a diretriz PICOC apresentada na seção 2.6.1.2. A *string* de pesquisa foi gerada concatenando os elementos por meio de conectores

lógicos E / OU. Conforme apresentado na Tabela 4, alguns sinônimos e termos semelhantes foram adicionados, visando assim uma maior eficácia no retorno de pesquisas primárias.

Tabela 4 – String de Busca Genérica.

String de Busca
<i>(“blockchain”) AND (“provenance” OR “data provenance”) AND (“approach” OR “architecture” OR “framework” OR “infrastructure” OR “method” OR “model” OR “solution” OR “technique” OR “platform” OR “tool” OR “process” OR “software”).</i>

A validação desta *string* foi feita através da recuperação de alguns artigos previamente conhecidos, aos quais deveriam ser retornados na busca no momento de execução nas bases. Os artigos selecionados para controle foram: (i) *ProvChain: A Blockchain-Based Data Provenance Architecture in Cloud Environment with Enhanced Privacy and Availability*; (II) *SmartProvenance: A Distributed, Blockchain Based Data Provenance System*; (III) *Blockchain Based Provenance Sharing of Scientific Workflows*.

2.6.1.4 Fonte de Busca

O processo de seleção do banco de dados, obedeceu alguns critérios, como: Disponibilidade para executar a string elaborada; Repositórios de fácil acesso; Repositórios com conteúdo de Ciência da Computação e Engenharia.

Tabela 5 – Bases de Busca Utilizadas.

Base de dados	URL
ACM Digital Library	dl.acm.org
EI Compendex	www.engineeringvillage.com
IEEEExplore	ieeexplore.ieee.org
Scopus	www.scopus.com
Springer	www.springer.com
Web of Science	apps.webofknowledge.com

2.6.1.5 Critério de Inclusão e Exclusão

Os critérios de inclusão e exclusão visam selecionar estudos de pesquisa que se enquadrem nas questões de pesquisa propostas.

- Critérios de Inclusão: (IC1): O estudo propõe uma solução que utiliza o *blockchain* como mecanismo para o armazenamento e a gerência de dados de proveniência; (IC2):

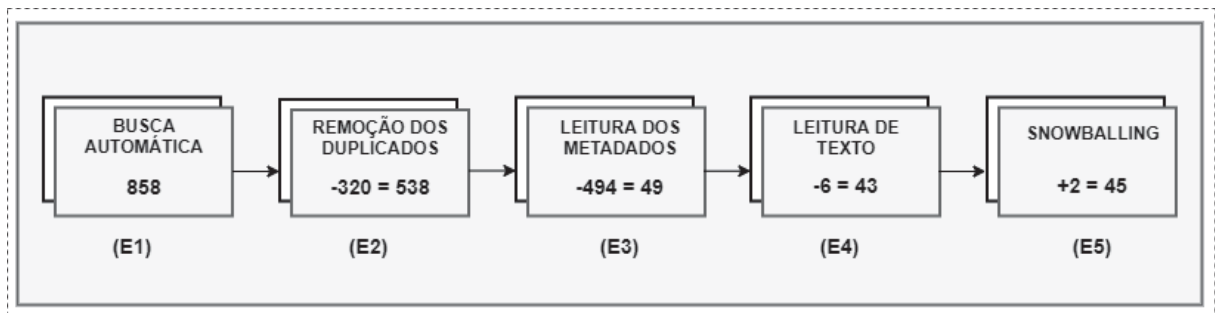
O estudo foi redigido em inglês; (IC3): O estudo foi publicado de 2008 até 2020; (IC4): Disponível como artigos completos em bases de dados digitais;

- Critérios de Exclusão: (EC1): Combina com a palavra-chave na *string* de pesquisa, mas o contexto é diferente dos propósitos da pesquisa; (EC2): O resumo não abordou nenhum aspecto da pesquisa questões; (EC3): Duplicado, ou seja, o trabalho já foi recuperado em outra base de conhecimento; (EC4): O artigo não contém resumo; (EC5): Não é um estudo primário; (EC6): Não disponível para as credenciais da universidade; (EC7): O estudo foi publicado como um artigo curto; (EC81): O estudo não está escrito em inglês; (EC9): O estudo não foi publicado em uma conferência ou periódico relacionado à Informática Ciência; (EC10): O estudo não foi publicado em um veículo de revisão por pares; (EC11): O estudo foi publicado antes de 2008; (EC12): O estudo não propõe uma solução que utiliza o *blockchain* como mecanismo para o armazenamento e a gerência de dados de proveniência.

2.6.2 Condução

Para auxiliar na condução do mapeamento a ferramenta Parsif.al¹⁰ foi utilizada. Esta ferramenta é de código aberto e auxilia na gestão e na execução do processo de condução da pesquisa. A Figura 8 sintetiza como o processo foi conduzido, ao qual envolve 5 cinco etapas.

Figura 8 - Fluxo de filtragens e etapas do mapeamento do sistemático.



Fonte: Elaborada pelo autor.

Na primeira etapa (E1) depois da construção da string de busca, foi realizada uma pesquisa automática nas bibliotecas digitais definidas na seção anterior. Os resultados foram exportados como entradas BibTeX e foram mesclados na ferramenta Parsifal. Um total de 858 estudos primários foram devolvidos na E1. Alguns critérios de exclusão já foram aplicados devido à disponibilização, pela ferramenta, de filtros que permitem isso,

¹⁰ <https://parsif.al/>

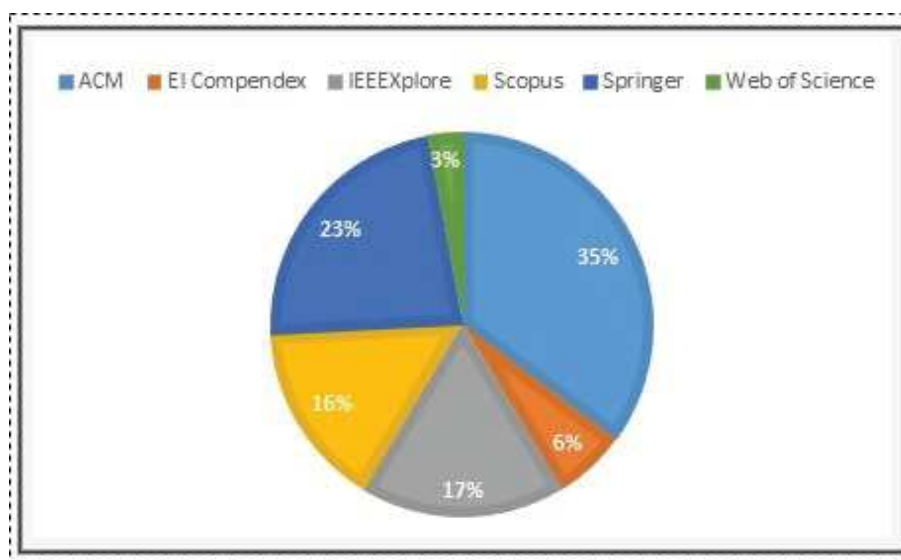
tal como o filtro de ano da publicação. A Tabela 6 apresenta o total de artigos retornados de acordo com cada biblioteca.

Tabela 6 – Total de Artigos Retornados em Cada Base utilizada.

Base de dados	URL	Total Retornado
ACM Digital Library	dl.acm.org	194
EI Compendex	www.engineeringvillage.com	167
IEEEExplore	ieeexplore.ieee.org	94
Scopus	www.scopus.com	173
Springer	www.springer.com	140
Web of Science	apps.webofknowledge.com	90

Na segunda etapa (E2) com auxílio da ferramenta Parsif.al, foi possível detectar e remover artigos duplicados. A remoção destes trabalhos foi feita de forma automática. No final dessa etapa, 320 artigos foram excluídos, resultando 538 estudos. Notamos que *ACM Digital Library* e a biblioteca *Springer* foram as bibliotecas que retornaram o maior número de artigos, após a etapa (E2). Juntos, eles foram responsáveis por 58% do total de artigos retornados. A Figura 9 representa o cenário da etapa (E2).

Figura 9 - Porcentagem de artigos retornados por base.



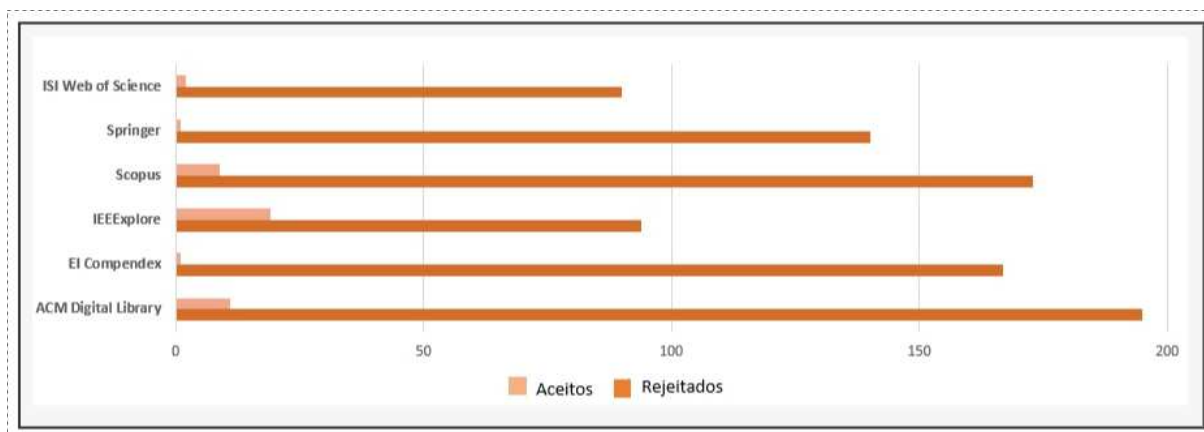
Fonte: Elaborada pelo autor.

Na terceira etapa (E3), os títulos e os resumos dos trabalhos foram lidos. O objetivo dessa etapa foi verificar se os trabalhos estavam no contexto da utilização de *blockchain* para proveniência. Como resultado dessa filtragem, um total de 494 artigos foram eliminados, restando assim 49 artigos. Na quarta etapa (E4), uma leitura do texto

foi realizada e a aplicação dos critérios de inclusão e exclusão foram realizados. Ao final da etapa (E4), 6 artigos foram eliminados, resultado 43 artigos.

Após a quarta etapa (E4), observamos que dos 35% de artigos avaliados, que correspondem a 184 documentos da biblioteca ACM, 11 artigos foram aceitos. Dos 123 artigos da biblioteca *Springer* (23%), 1 artigo foi aceito. O *IEEE Explorer* teve 93 artigos, o que corresponde a 17%. Desses, 19 foram aceitos. Dos 84 artigos da biblioteca *Scopus* que correspondem a 16%, 9 artigos foram aceitos. O total de artigos retornados no *EI Compendex* foi 33, correspondendo a 6%, do total. Desses, 1 artigo foi aceito. Por fim, a biblioteca *Web of Science* representou 3%, o que corresponde a 16 artigos, e 2 artigos foram aceitos. O total de artigos aceitos e rejeitados, distribuídos pelas bibliotecas, pode ser visualizado na Figura 10.

Figura 10 - O total de artigos aceitos e rejeitados.



Fonte: Elaborada pelo autor.

Na quinta etapa (E5), os métodos de *Backward Snowballing* foi utilizado. Como resultado dessa etapa, 2 artigos foram incorporados aos anteriores, totalizando 45 artigos no final.

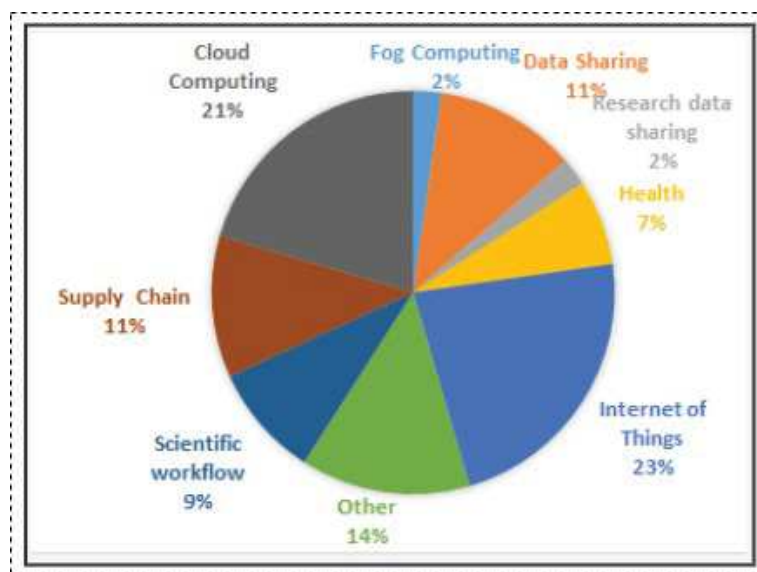
2.6.2.1 Relatos dos Resultados

Os 45 artigos foram analisados e classificados para responder às questões de mapeamento.

MQ1) Como a tecnologia blockchain tem sido utilizada como mecanismo, método e ferramenta para a proveniência?

O objetivo desta MQ foi identificar o estado atual da arte para identificar em quais áreas de aplicação a tecnologia *blockchain* tem sido utilizada como mecanismo, método ou ferramenta para a proveniência. A Figura 11 apresenta os resultados. Nove

Figura 11 - Áreas de aplicação.



Fonte: Elaborada pelo autor.

áreas de aplicação foram identificadas, de acordo os estudos primários selecionados. A saber, *Internet of Things*, *Cloud Computing*, *Scientific Workflow*, *Research Data Sharing*, *Data Sharing*, *Health*, *Fog Computing*, *Supply Chain* e *Others*. Através dos resultados, foi identificado que existem um grande número de áreas e oportunidades que utilizam *blockchain* como mecanismo, método e ferramenta para a proveniência. *Internet of Things* e *Cloud Computing* representaram a maioria das pesquisas totalizando 44% dos artigos. *Internet of Things* representou 23% do total e *Cloud Computing* 21%. *Supply Chain* e o de *Data sharing* representa 11% cada. *Scientific Workflow* representou 9% do total, o de *Health*, 7%, *Research Data Sharing* 2%, *Fog computing*, 2% e por último outros (*Others*), que representou 14%.

MQ2) Veículos em que os artigos foram publicados?

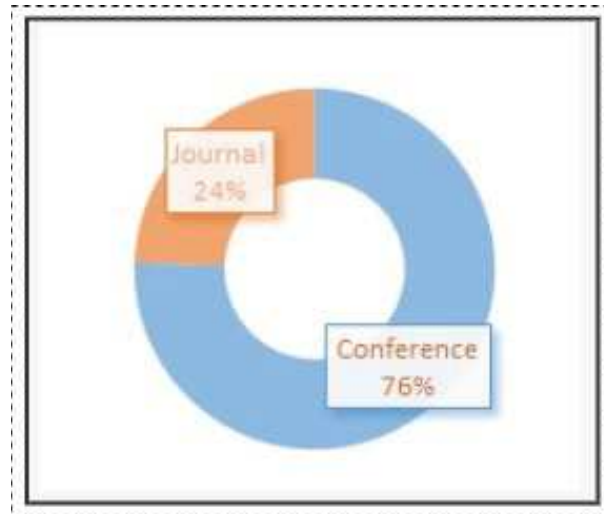
O objetivo dessa MQ foi descobrir onde os estudos foram publicados. Através dos resultados foi possível identificar que a maioria dos estudos publicados foram em conferências científicas, totalizando 76% dos artigos verificados. A Figura 12 apresenta os resultados.

MQ3) Qual é a distribuição dos estudos no decorrer dos anos?

A partir da Figura 13, podemos verificar que a maioria dos trabalhos foram publicados a partir de 2018, com uma curva ascendente. 2019 e 2020 foram os anos mais promissores, o que indica um aumento de trabalhos na área, mostrando a importância das pesquisas.

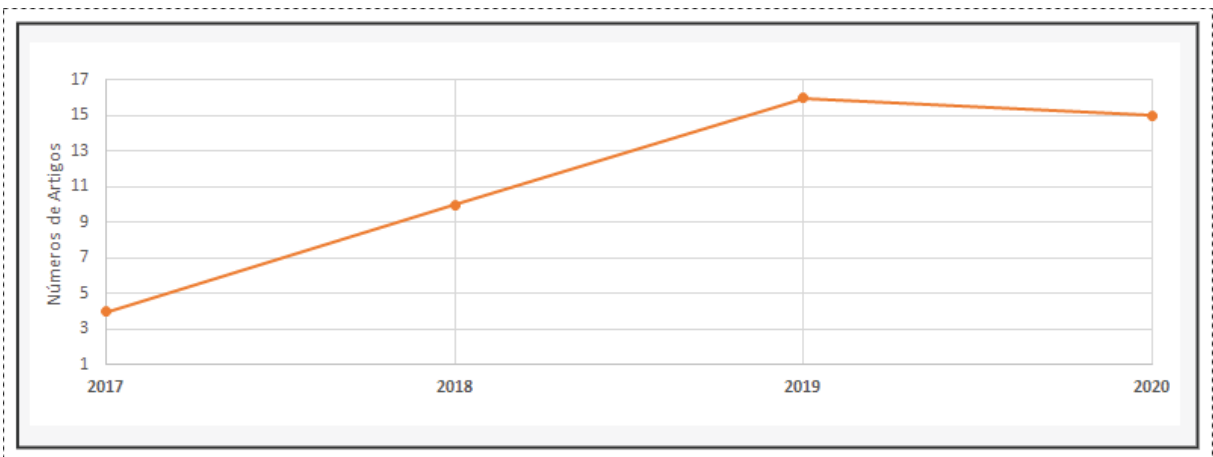
MQ4) Quais foram os métodos de pesquisa?

Figura 12 - Onde os estudos foram publicados.



Fonte: Elaborada pelo autor.

Figura 13 - Distribuição dos estudos no decorrer dos anos.



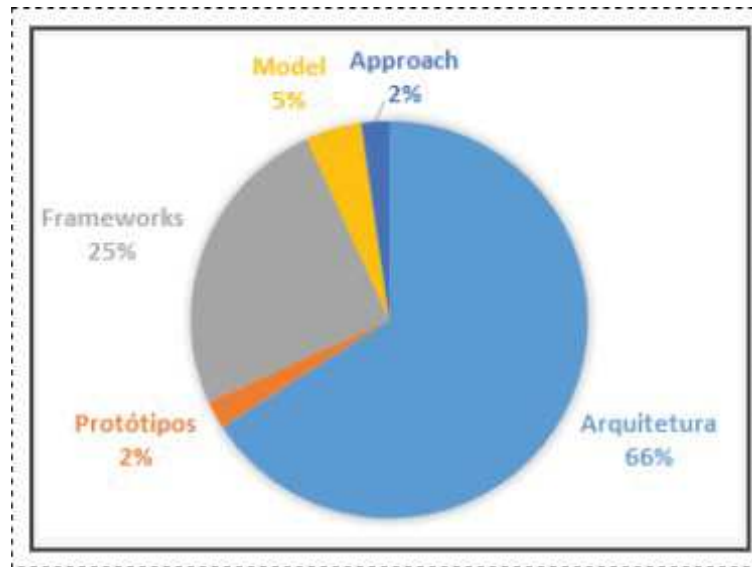
Fonte: Elaborada pelo autor.

O objetivo deste MQ foi descobrir quais metodologias de pesquisa foram utilizados em cada estudo. Foi possível identificar que a maioria dos estudos detalham arquiteturas, representando 66% dos artigos analisados. *Frameworks* representam 25% dos artigos analisados. *Model*(modelos) representam 5%. E por fim Protótipos e *Approach* (abordagens) representam 2% cada dos artigos analisados. A Figura 14 apresenta os resultados.

MQ5) Quais são as vantagens e benefícios obtidos nas abordagens encontradas com a utilização da tecnologia blockchain para a proveniência?

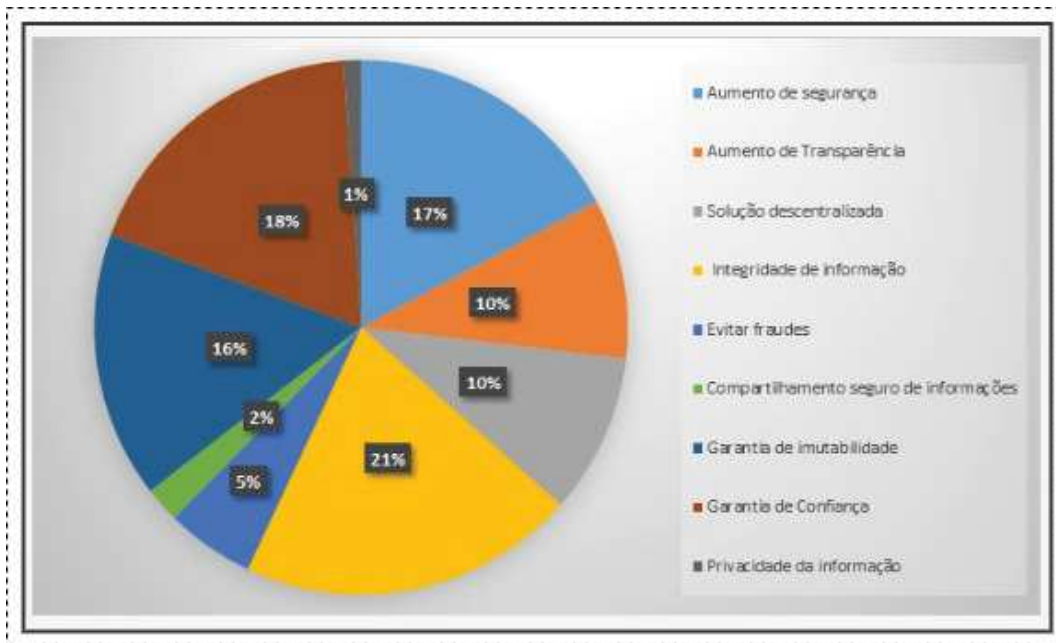
O objetivo desta MQ foi identificar as motivações dos estudos em aplicar os recursos

Figura 14 - Método ou metodologia de pesquisa.



Fonte: Elaborada pelo autor.

Figura 15 - Motivação uso tecnologia *blockchain* para proveniência.



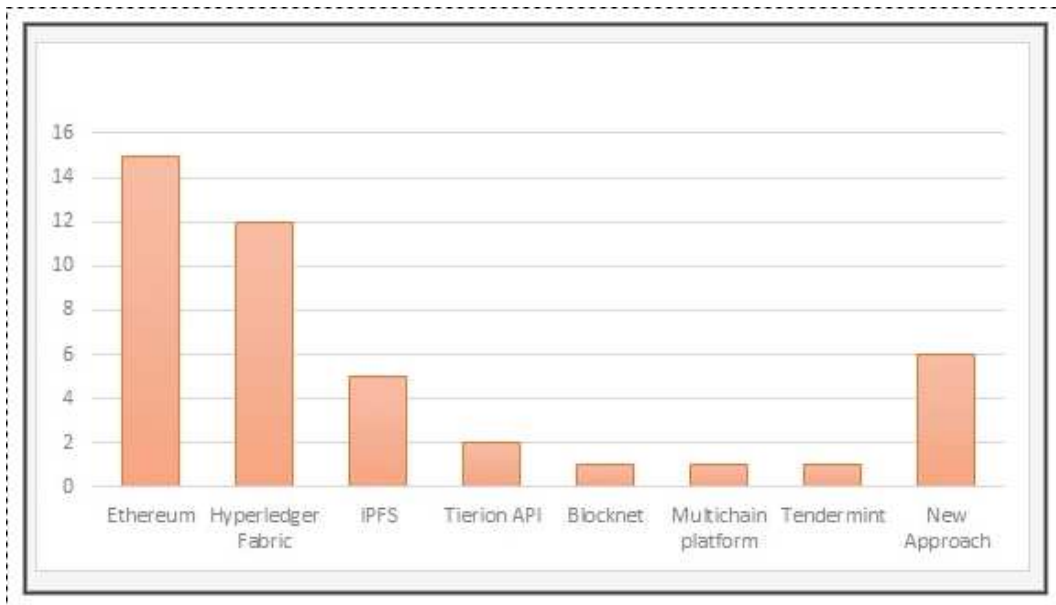
Fonte: Elaborada pelo autor.

da tecnologia *blockchain*. Às três maiores motivações, conforme a Figura 15 foram: i) preocupação com “**Integridade de Informação**” que representou um total 21%, a integridade é garantida em um *blockchain*, visto que os dados não podem ser apagados ou alterados, garantindo assim a credibilidade dos dados ou proveniência, ii) “**Garantia**

de Confiança” que representou um total de 18%, em um *blockchain* como os membros compartilham uma visualização única dos dados, é possível ver todos os detalhes de uma transação, o que oferece maior confiança e iii) **“Aumento de segurança”** que representou um total de 17%, a natureza distribuída de um *blockchain* permite que cada nó que participa da rede, tenha e verifique os dados do razão, aumentando assim a segurança da informação.

MQ6) Quais são os métodos, padrões ou tecnologias mais utilizados (ou propostos) pelos autores para embasar suas abordagens? Pretendemos avaliar as soluções sob alguns aspectos, (1) plataformas ou arquiteturas de blockchain utilizadas e (2) mecanismo de consenso. Quais plataformas ou arquiteturas de blockchain foram utilizadas pelos autores em suas propostas?

Figura 16 - Plataformas ou arquiteturas de *blockchain*.



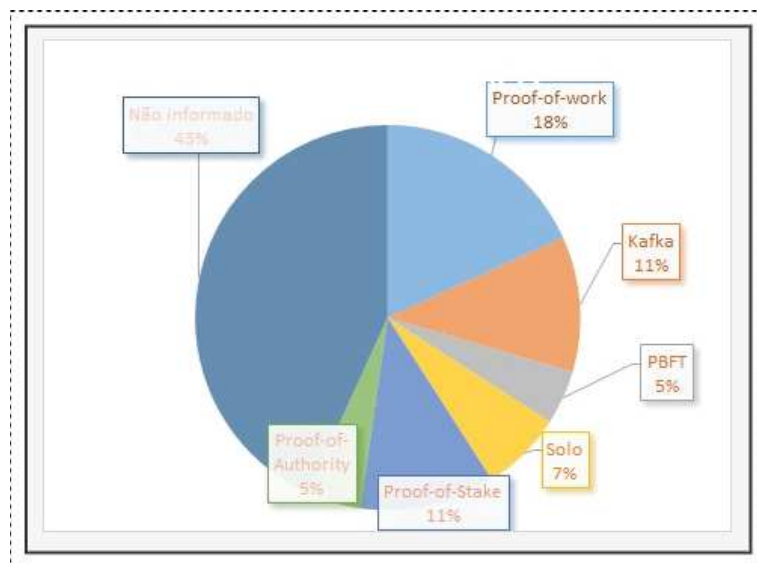
Fonte: Elaborada pelo autor.

O objetivo deste MQ foi descobrir quais eram as plataformas utilizadas e identificar os seus principais benefícios e aplicabilidade considerando dados de proveniência. A Figura 16. apresenta os resultados.

A plataforma de *blockchain* mais usada foi a *Ethereum* (BUTERIN, 2013). A escolha do *Ethereum* é justificada por ser a plataforma mais conhecida, e por ser *open source*. No entanto, nota-se um crescimento de uso da plataforma *Hyperledger Fabric* (ANDROULAKI et al., 2018), sendo utilizada pela abordagem apresentada nesta dissertação. A escolha da *Hyperledger Fabric* é justificada na seção que detalhamos seu uso. Também foram identificados o uso de outras plataformas, mas em menor número de propostas.

Quais foram os mecanismos de consenso utilizados pelos autores em suas propostas?

Figura 17 - Mecanismos de Consensos.



Fonte: Elaborada pelo autor.

A maioria dos estudos, (43%) não apresenta o consenso aplicado, não sendo possível afirmar se eles tiveram a intenção de não relatar ou por falta de conhecimento, utilizaram a implementação padrão da plataforma escolhida.

Proof-of-Work, foi utilizado em 18% dos artigos, e é um mecanismo de consenso que, para validar e publicar um bloco na *blockchain*, requer uma certa quantidade de trabalho computacional (PoW) (WANG et al., 2019; WAN et al., 2020). Esse protocolo de consenso é mais fortemente associado ao *blockchain* devido à sua integração com o *Bitcoin* (NAKAMOTO, 2008).

Proof-of-Stake, representa uma classe de algoritmos de consenso em que os validadores votam no próximo bloco, e o peso da votação depende do tamanho de sua aposta (WANG et al., 2019; WAN et al., 2020). 11% dos artigos utilizam essa abordagem.

O consenso *Kafka*, foi utilizado em 11% dos artigos, e é o mecanismo de consenso usado no *Hyperledger Fabric*. Ele é baseado em votação com permissão, o que fornece tolerância a falhas e um bom para o desempenho. No entanto, este consenso, não é um tolerante a falhas bizantinas, o que evitaria que o sistema chegasse a um acordo no caso de nós maliciosos ou defeituosos (WANG et al., 2019; WAN et al., 2020). A implementação do consenso *Kafka* utiliza o *Apache Kafka*¹¹, uma plataforma de *streaming* distribuída.

Solo é um algoritmo de consenso simples para o *Hyperledger Fabric*. É utilizado em 7% dos trabalhos. Ele é chamado de solo porque executa uma única instância do serviço do solicitante. É útil para o desenvolvimento, mas não é recomendado para o ambiente de

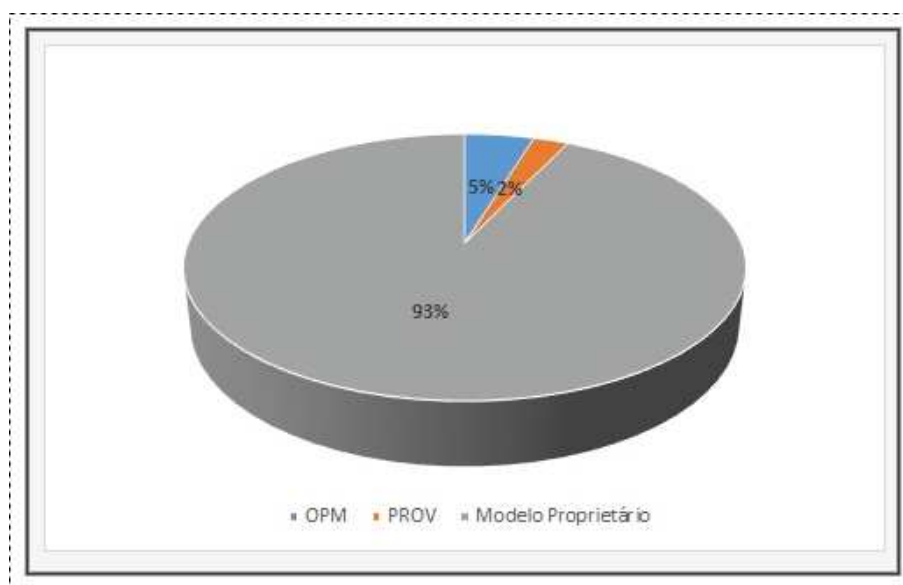
¹¹ <https://kafka.apache.org/>

produção, visto que será a única instância do serviço de pedido na rede, a instância será uma única ponto de falha.

O *Practical Byzantine Fault Tolerance* (PBFT) é um algoritmo de consenso genérico, que permite que os sistemas distribuídos continuem funcionando mesmo que um nó na rede *blockchain* esteja com defeito ou atue como um nó malicioso. Esses problemas são comuns em sistemas distribuídos. Com esta implementação, os nodos honestos chegam a um consenso e a rede não é afetada por um sistema malicioso ou nó defeituoso (WANG et al., 2019; WAN et al., 2020). É utilizado em 5% dos trabalhos.

MQ7) Nas abordagens encontradas quais são os modelos utilizados para representar os dados de proveniência?

Figura 18 - Modelos de Proveniência.



Fonte: Elaborada pelo autor.

A maioria dos estudos não apresenta um modelo de proveniência específico. Conforme apresentado na Figura 18, 93% dos trabalhos utilizam algum modelo proprietário, que não é padrão. 5% dos artigos utilizam OPM e 2% PROV. Este resultado mostra claramente que o PROV, por ser um modelo novo, ainda não foi utilizado largamente nos trabalhos, apesar de seus benefícios. Esse resultado corrobora com a importância de se ter uma abordagem que permita a interoperabilidade e integração dos dados de proveniência, visto que 87% dos utilizam modelos proprietários que sem algum mecanismo específico, não são interoperáveis.

2.6.3 Análise dos trabalhos

Com base nos resultados do mapeamento sistemático, os trabalhos resultantes foram analisados considerando a abordagem proposta nesta dissertação e as questões de pesquisa. Além disso, foi realizada uma busca *ad-hoc* por trabalhos que tratam de questões como reprodutibilidade científica, interoperabilidade ou que utilizam o modelo PROVONE, de forma a complementar aos resultados do mapeamento sistemático. Discutimos a seguir os principais resultados.

Chen et al. (2018) propuseram, uma abordagem baseada em *blockchain* denominada ProChain, para facilitar o compartilhamento de dados de proveniência de *workflows* científicos entre cientistas geograficamente distribuídos. Com a ProChain, cientistas podem compartilhar com outros cientistas a execução de *workflows* de uma forma confiável, podendo reaproveitar resultados já obtidos, economizar tempo, custos, além melhorar a cooperação entre os cientistas. No entanto, na ProChain os autores não consideram a coleta de proveniência em tempo real, o que pode dificultar a transparência, a integridade, e a confiança dos dados de proveniência coletados entre cientista geograficamente distribuído. Além disso, não utilizam uma infraestrutura de armazenamento distribuído e de alto desempenho, como é o caso da Blockflow. A integridade dos dados também é comprometida, uma vez que existe um mecanismo que avalie se a proveniência compartilhada foi alterada ou manipulada.

Fernando et al. (2019) propuseram um sistema baseado em *blockchain* chamado SciBlock para fornecer um armazenamento confiável e à prova de violação para dados de proveniência de *workflows* científico em um ambiente colaborativo. No SciBlock, cada registro de proveniência contém um subconjunto de informações de proveniência relacionadas a uma tarefa de *workflow*, incluindo o ID da tarefa, a entrada para a tarefa, a saída gerada pela tarefa, o tempo de execução e o usuário que executou a tarefa. Além disso, os autores usam uma abordagem *off-chain* que usa uma combinação de consulta em um banco de dados relacional não criptografado e verificação *blockchain* com intuito de acelerar o processo de consulta da proveniência armazenada. No entanto, na SciBlock, os autores não consideram a captura e o armazenamento de proveniência, prospectiva e retrospectiva. Esses tipos de proveniência são identificados como um requisito essencial para registro de todo o processo computacional em um *workflow*, para atingir reprodutibilidade. Além disso, os autores também não consideram um modelo para captura de proveniência, o que é importante para interoperabilidade em um *workflow* colaborativo.

Liang et al. (2017) propuseram uma arquitetura baseada em *blockchain* chamada ProvChain, para garantir a descentralização, integridade e a confiabilidade em dados de proveniência, em aplicativos de armazenamento na nuvem. A principal ideia é fornecer um ambiente seguro, a prova de falsificação, privacidade e confiabilidade, no qual um usuário pode ter controle da manipulação dos seus dados armazenados na nuvem. A ProvChain

rastreia e coleta em tempo real, todas as atividades (eventos correspondentes às ações de usuário, tais como, criação, modificação, cópia, compartilhamento e exclusão) que ocorrem sobre dados armazenados com intuito de gerar proveniência. A proveniência produzida após a detecção de uma operação de usuário é enviada para a *blockchain* incluindo atributos tais como, data/hora, operação realizada pelo usuário na manipulação desses arquivos (criação, modificação, cópia, compartilhamento ou exclusão) entre outros. Na ProvChain todo o registro de proveniência é associado a um ID de usuário com *hash* para preservar a privacidade, ou seja, para que nós da rede *blockchain* não possam correlacionar registros de dados associados a um usuário específico. No entanto, ainda sim, esses mesmos dados podem ser acessados e visualizados por usuários não autorizados que pertençam à rede. Esse acesso pode ser um problema, considerando a privacidade e propriedade intelectual de um *workflows* científico. Na BlockFlow, os dados são gerenciados entre diferentes parceiros de pesquisa de uma forma segura e privada até que queiram compartilhar os seus resultados de maneira aberta em publicações científicas.

Tosh et al. (2017) propuseram uma estrutura baseada em *blockchain* para dados de proveniência em uma plataforma de nuvem chamada BlockCloud. No BlockCloud, todas as operações do usuário em arquivos armazenados na plataforma de nuvem são rastreadas para gerar dados de proveniência. Os autores descrevem os desafios e oportunidades ao incorporar PoS (*Proof-of-Stake*) em relação ao PoW (*Proof-of-Work*) como mecanismo de consenso. Em contraste, o *blockchain* com permissão aproveita protocolos mais rápidos para alcançar o consenso. Desta maneira, *blockchain* com permissão torna-se uma opção mais realista para *workflows* científicos colaborativos com compartilhamento de dados de proveniência, como é o caso da Blockflow.

Ramachandran et al. (2018) propuseram uma arquitetura denominada Smartprovenance/DataProv, baseada em *blockchain* para o gerenciamento de dados de proveniência, seguro e imutável, baseadas em controle de acesso. A SmartProvenance/DataProv é um sistema de proveniência que rastreia alterações em documentos em plataformas da nuvem e registra a proveniência no *blockchain* ao longo das atualizações nesses documentos com base em um mecanismo de votação. Além disso, aplica penalidade aos usuários, para garantir que nenhuma alteração maliciosa seja realizada. Os autores utilizam o *blockchain* como uma plataforma para facilitar a coleta, verificação e gerenciamento de dados de proveniência, em ambiente distribuído, garantindo alta disponibilidade e tolerância a falhas. No Smartprovenance os autores utilizam o modelo OPM. No entanto, não possui um mecanismo de consulta aos dados de proveniência, o que é necessário para cenários de *workflows* científicos colaborativos, uma vez que pesquisadores frequentemente consultam simultaneamente o repositório de proveniência, seja para monitorá-lo ou para planejar ações futuras.

Kim, Henry et al. (2018) propôs a ontologia de rastreabilidade TOVE, relacionada a um *blockchain* para proveniência da cadeia de suprimentos. No trabalho, os dados de

proveniência são armazenados e representados no *blockchain* por contrato inteligente e por meio da *TOVE Traceability Ontology*. Na BlockFlow, a proveniência é representada por meio do modelo de proveniência ProvONE, que fornece interoperabilidade para proveniência a partir de dados científicos heterogêneos.

Costa et al. (2014) propuseram uma arquitetura denominada ProvSearch que combina técnicas de gerenciamento de *workflows* distribuído com gerenciamento de dados de proveniência. Os dados de proveniência são tratados em um modelo chamado PROV-Wf, uma extensão do modelo PROV para o domínio dos *workflows* científicos (COSTA et al., 2014). Na ProvSearch os nós de banco de dados formam uma rede descentralizada de servidores de bancos de proveniência. No entanto, ao contrário dos *blockchains*, ainda existe uma certa medida de centralidade nas arquiteturas distribuídas tradicionais, levando a uma baixa confiabilidade dos dados de proveniência, como é o caso desta abordagem. Esses sistemas também apresentam problemas de segurança no armazenamento de informações, visto que qualquer usuário autorizado pode corromper ou alterar os dados de proveniência. Assim, é necessário para a reprodutibilidade e confiabilidade dos experimentos científicos, que nenhum usuário possa alterar os dados armazenados. Na BlockFlow, os dados de proveniência são armazenados imutavelmente no ambiente *blockchain*.

Mendes et al. (2019) propuseram uma arquitetura baseada em uma abordagem Polystore para representar dados de proveniência heterogêneos gerados por diferentes WfMSs, em um cenário de ciência colaborativa. Oliveira et al. (2016) apresentou uma proposta para integrar dados de proveniência de *workflows* distribuídos e heterogêneos. PBase (CUEVAS-VICENTTÍN et al., 2014) propuseram um repositório de proveniência de *workflows* científicos que implementa a ontologia ProvONE, permitindo armazenamento, análise e replicação de experimentos científicos.

SciCumulus (DE OLIVEIRA et al., 2010b) propuseram um *middleware* para orquestrar *workflows* científicos por meio do SGWfC em ambientes distribuídos e paralelos. Esta abordagem oferece um serviço de captura de proveniência e tempo real, repositório de proveniência, onde o acesso a estas informações é feito através de consultas a este banco de dados.

No entanto, essas abordagens têm como principal desvantagem um sistema de armazenamento centralizado para dados de proveniência. Se o servidor central for comprometido, os dados de proveniência podem ser comprometidos e perdidos. Portanto, não há um único ponto de falha na arquitetura *blockchain*, visto que os dados de proveniência são descentralizados, compartilhados entre os pesquisadores distribuídos geograficamente.

2.7 DISCUSSÕES

Conforme apresentado, existem diferentes soluções na literatura que utilizam *blockchain* como mecanismo para armazenamento confiável de proveniência, além de

diversas abordagens para a gerência de proveniência no contexto de experimentos científicos. No entanto, essas abordagens apresentam algumas limitações, considerando a abordagem proposta nesta dissertação e as questões de pesquisa.

Com objetivo de comparar a arquitetura BlockFlow em relação aos trabalhos relacionados. Definimos 7 aspectos a saber: **Aumento de Transparência:** na solução proposta é possível obter uma visão geral dos dados de proveniência de forma transparente, podendo verificar, como os dados de proveniência foram criados ao longo do tempo?; **Garantia de confiança e integridade:** na solução proposta é possível garantir a confiança e a integridade nos dados de proveniência, ou seja, é possível garantir e verificar se um dado ou a proveniência foi manipulado ou não; **Solução descentralizada (*blockchain*) e segura:** a solução proposta garante que os dados sejam compartilhados de forma que não haja nenhum ponto falha, garantindo assim segurança na informação; **Modelo de proveniência padrão e interoperável:** a solução propõe ou tem um modelo proveniência padrão, como PROV, OPM ou ProvONE; **Privacidade de informações:** a solução proposta garante que as informações ou proveniência, sejam acessadas somente por pessoas autorizadas. **Armazenamento distribuído e escalável:** a solução proposta garante que dados sejam armazenados de maneira distribuída e escalável, ou seja, em um *blockchain* que permite armazenamento ou consenso rápido e escalável.

De acordo com a Tabela 7, constatamos que o BlockFlow possui características que outros projetos não possuem na sua totalidade. Muitas abordagens não utilizam um modelo padrão e interoperável de proveniência, outras não apoiam transparência, confiança ou a integridade de dados nos dados e na proveniência. Assim, diante dessas lacunas, foi constatada viabilidade de desenvolvimento da proposta desse trabalho.

Tabela 7 – Comparação entre a BlockFlow e as Propostas Encontradas na Literatura.

	<i>Chen et al., 2018</i>	<i>Fernando et al., 2019</i>	<i>Liang et al., 2017</i>	<i>Tosh et al., 2017</i>	<i>Ramachandran et al., 2018</i>	<i>Kim, Henry et al., 2018</i>	<i>Costa et al., 2014</i>	<i>Mendes et al., 2019</i>	<i>Oliveira et al., 2016</i>	<i>Cuevas-Vicentín et al., 2014</i>	<i>BlockFlow</i>
Garantia de Confiança e Integridade de dados e proveniência	x	y	y	y	y	y	x	x	x	x	y
Solução descentralizada	x	y	y	y	y	y	x	x	x	x	y
Privacidade de informações	y	y	y	y	x	y	x	x	x	x	y
Modelo de proveniência padrão e interoperável	x	x	x	x	y	y	y	y	y	y	y
Armazenamento distribuído e escalável	x	x	x	x	y	x	x	x	x	x	y

3 ARQUITETURA BLOCKFLOW

No capítulo anterior foram apresentados os conceitos relativos ao entendimento da solução proposta nesta dissertação, incluindo o detalhamento da plataforma E-SECO, além dos principais trabalhos relacionados, a partir da condução de um mapeamento sistemático. A partir dos resultados do mapeamento, identificamos algumas lacunas nos trabalhos relatados na literatura no que tange a captura de proveniência em *workflows* colaborativos, distribuídos e heterogêneos, considerando o suporte a segurança, compartilhamento de informações, confiabilidade, heterogeneidade e reprodutibilidade. Desta forma, este capítulo apresenta e detalha a BlockFlow, uma arquitetura cujo objetivo é prover mecanismos para suporte a confiabilidade, transparência, privacidade, interoperabilidade e reprodutibilidade na experimentação científica colaborativa e distribuída. Para tanto, a BlockFlow é utilizada no contexto do ecossistema científico E-SECO (FREITAS et al., 2015), para auxiliar principalmente na confiabilidade dos dados científicos utilizados e disponibilizados pelo E-SECO.

3.1 DEFINIÇÃO METODOLÓGICA

Esta pesquisa foi realizada com base na metodologia *Design Science Research* (DSR) (HEVNER et al., 2004) (HEVNER, et al., 2008). Esta metodologia assume que as soluções (artefatos) são projetadas para problemas práticos. Em DSR, o projeto do artefato corresponde a uma atividade iterativa e incremental (HEVNER et al., 2008). A avaliação do artefato ocorre a cada ciclo de DSR e fornece *feedback* para construção e aprimoramento do produto.

A arquitetura BlockFlow como solução proposta, corresponde ao artefato. Cada ciclo gera conhecimento científico, a partir das avaliações realizadas. Esse conhecimento ajuda a construir novas versões da arquitetura, para compor a solução no contexto da plataforma E-SECO. Para a execução da DSR, seguimos algumas etapas, como definição do problema, revisão da literatura e discussão das soluções existentes, desenvolvimento do artefato, avaliação e discussão dos resultados.

Portanto, primeiro identificamos a relevância do problema como “apoiar a execução de experimentos científicos colaborativos, ancorados por rede distribuída, na nuvem, com foco na interoperabilidade, privacidade e confiança em dados de proveniência compartilhados, de forma dar suporte a reprodutibilidade dos resultados obtidos na experimentação científica”. Resumimos nossas contribuições de pesquisa como a proposta de uma arquitetura baseada em *blockchain*, denominada BlockFlow, para suporte a segurança, compartilhamento de informações, confiabilidade e heterogeneidade e na pesquisa colaborativa no contexto da plataforma E-SECO, compartilhando dados de proveniência de uma maneira mais confiável, com intuito de reprodutibilidade dos resultados obtidos. Para

verificar a viabilidade da proposta, realizamos uma avaliação com dados de experimentos relacionados a doença COVID-19 no capítulo 4.

A arquitetura do BlockFlow foi desenvolvida considerando os seguintes requisitos:

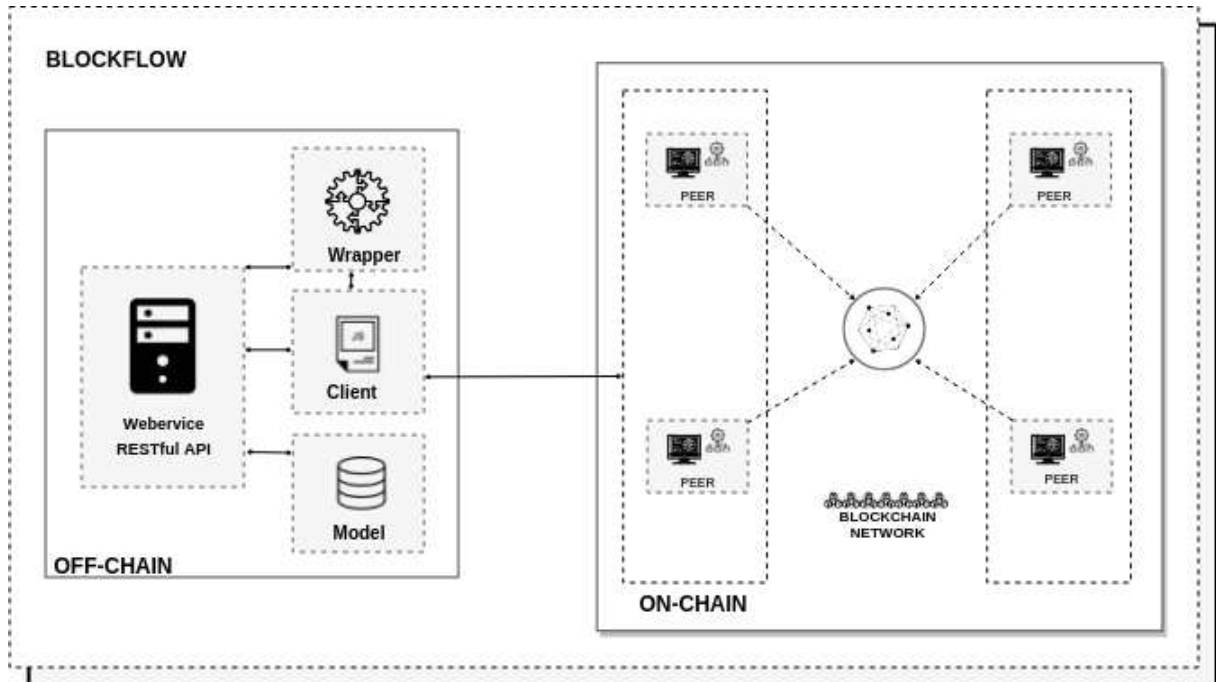
- **Reprodutibilidade:** a confiança nos dados de proveniência obtidos a partir dos *workflows* é crucial para apoiar a reprodutibilidade dos resultados científicos. Desta forma, os dados de proveniência na BlockFlow são coletados e armazenados de maneira confiável e imutável. Isso significa que os dados não podem ser manipulados sem deixar um rastro. Esse mecanismo impede manipulações arbitrárias de dados, seja consciente ou inadvertidamente (por exemplo, por pesquisadores tendenciosos).
- **Privacidade:** dados científicos são considerados propriedade intelectual, portanto, na BlockFlow, são compartilhados apenas entre partes ou pessoas autorizadas. A BlockFlow permite que os dados sejam gerenciados entre diferentes parceiros de pesquisa de uma forma segura e privada até que queiram compartilhar os seus resultados de maneira aberta em publicações científicas.
- **Transparência:** dados compartilhados entre os pesquisadores em um experimento devem ser transparentes. Todos os nós da rede (cientistas conectados a um ponto da rede), que compõem um experimento, podem verificar como os dados na cadeia (*blockchain*) foram criados ao longo do tempo. Desta forma, todas as atualizações de dados podem ser rastreadas entre nós. Além disso, os dados da pesquisa podem ser analisados e revisados pelos pares de uma maneira comprovada.
- **Interoperabilidade:** os Sistemas de Gerenciamento de *Workflows* Científicos (SWfMS) armazenam geralmente dados de proveniência em diferentes formatos. Isto não é um problema quando cientistas analisam estes dados de uma maneira isolada ou em um mesmo SWfMS. Em ambientes colaborativos onde os cientistas necessitam analisar e interpretar resultados de maneira colaborativa, oriundos de diferentes SWfMS, os dados de proveniência devem ser integrados. Na BlockFlow, para interoperabilidade de dados de proveniência, permitindo que pesquisadores analisem e comparem informações advindas de aplicações científicas heterogêneas, utiliza-se o modelo ProvONE (CUEVAS-VICENTTÍN, 2015), que será detalhado nas próximas seções.

3.2 COMPONENTES DA ARQUITETURA BLOCKFLOW

A BlockFlow foi especificada com base no modelo arquitetural em camadas e serviços. A Figura 19 apresenta um visão alto nível da arquitetura.

A camada **API RESTful Web Service** foi implementada para que a solução proposta pudesse ser conectada com outros aplicativos e plataformas que têm como objetivo

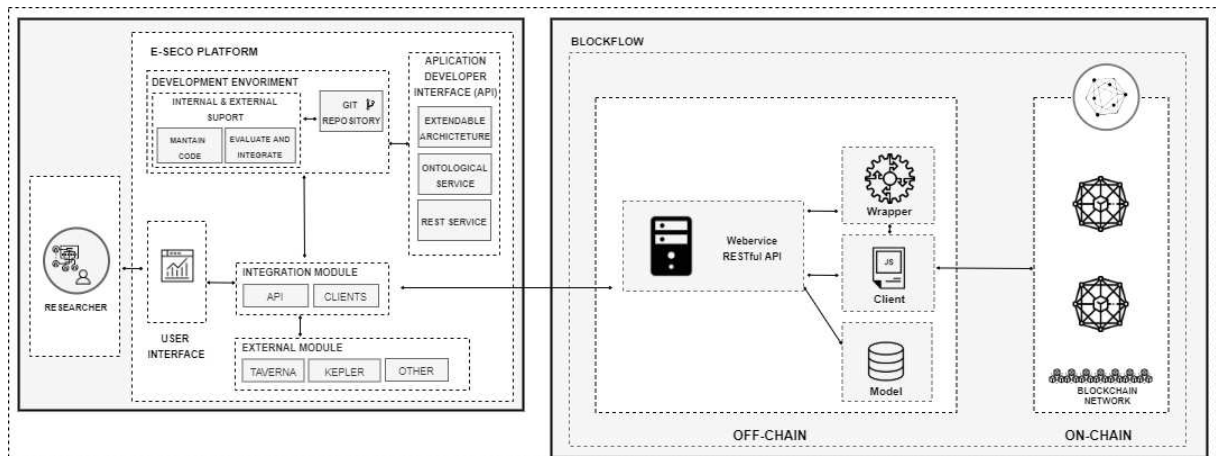
Figura 19 - Arquitetura BlockFlow.



Fonte: Elaborada pelo autor.

permitir que seus usuários criem redes *blockchain* para colaborar e garantir a confiança e reprodutibilidade de experimentos científicos. É através da camada *RESTful* que a arquitetura BlockFlow se integra à camada de Integração da plataforma E-SECO. A Figura 20 apresenta uma visão alto nível da comunicação entre a E-SECO e a BlockFlow.

Figura 20 - Integração E-SECO e Arquitetura BlockFlow.



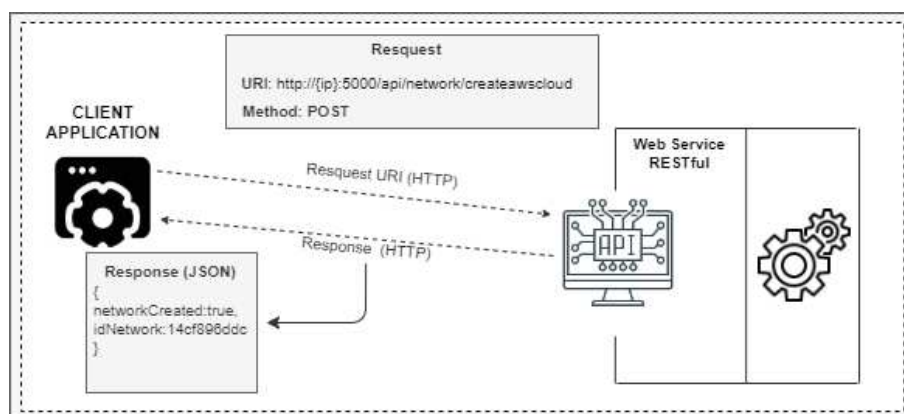
Fonte: Elaborada pelo autor.

A camada **Wrapper** foi desenvolvida para fazer a tradução dos dados de proveniência para o modelo ProvONE. A camada **Model** armazena informações referentes ao fluxo de processos do BlockFlow, como, por exemplo, dados de especificação dos ambientes de redes para colaboração, tais como, quais instituições e cientistas (geograficamente distribuídos) fazem parte de um experimento, dados de cadastro de usuário, entre outras informações relevantes para a execução da BlockFlow. A camada **Client** foi desenvolvida para que a API pudesse se conectar às redes *blockchains*. Por sua vez, a camada de **Blockchain Network**, permite que cientistas possam ter ambientes confiáveis de experimentação científica. Detalhamos a seguir cada uma dessas camadas.

3.2.1 Camada *API RESTful WebService*

A camada *API RESTful WebService* é uma API (*Application Programming Interface*) desenvolvida para permitir que a BlockFlow possa ser integrada a quaisquer outras plataformas ou aplicativos, com base na comunicação via *web services REST* (*Representational State Transfer*). Esta camada permite a interoperabilidade, ou seja, comunicação entre sistemas. Uma das grandes vantagens das APIs baseadas em serviços REST (*Representational State Transfer*) é que estas podem ser integradas com qualquer outra ferramenta que trabalhe com o protocolo de comunicação HTTP *Hypertext Transfer Protocol* (Protocolo de Transferência de Hipertexto - RFC 2616). As operações de requisição (solicitação de recursos ao servidor) são por meio de Endpoints, ou URIS e as respostas são por meio de JSON.

Figura 21 - Um exemplo de uma solicitação para a camada *RESTful Web Service API* da BlockFlow.

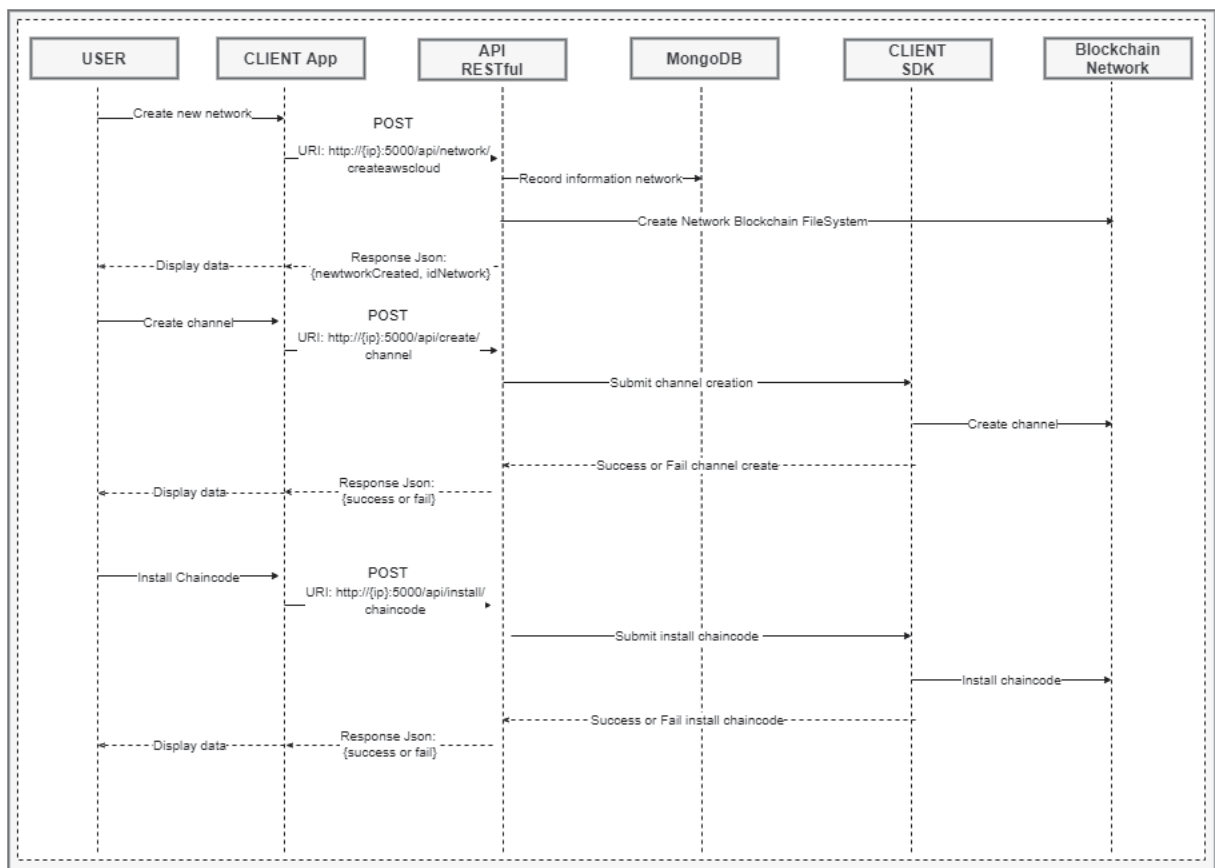


Fonte: Elaborada pelo autor.

Para ilustrar de maneira didática sua operação, a Figura 21 ilustra um exemplo no qual o fluxo de solicitação-resposta entre um cliente e a camada *API RESTful Web Service* da BlockFlow é representado. Com base na solicitação a um serviço, *web*, através da URI:

“http://ip:5000/api/network/createawscloud”, uma rede *blockchain*, para que pesquisadores possam colaborar em seus experimentos na nuvem, é criada. Como resposta, em formato JSON, um valor booleano, “networkCreated: true” é retornado caso a rede seja criada com sucesso juntamente com um identificador único para rede criada, “idNetwork:14cf89ddo”. O diagrama apresentado na Figura 22, mostra um fluxo de solicitações e respostas para camada *API RESTful Web Service* da *BlockFlow*, ao qual um cliente solicita a API, por exemplo, para criar uma rede, criar canais e instalar *chaincode*.

Figura 22 - Diagrama de solicitações e respostas para camada *API RESTful Web Service* da *BlockFlow*.



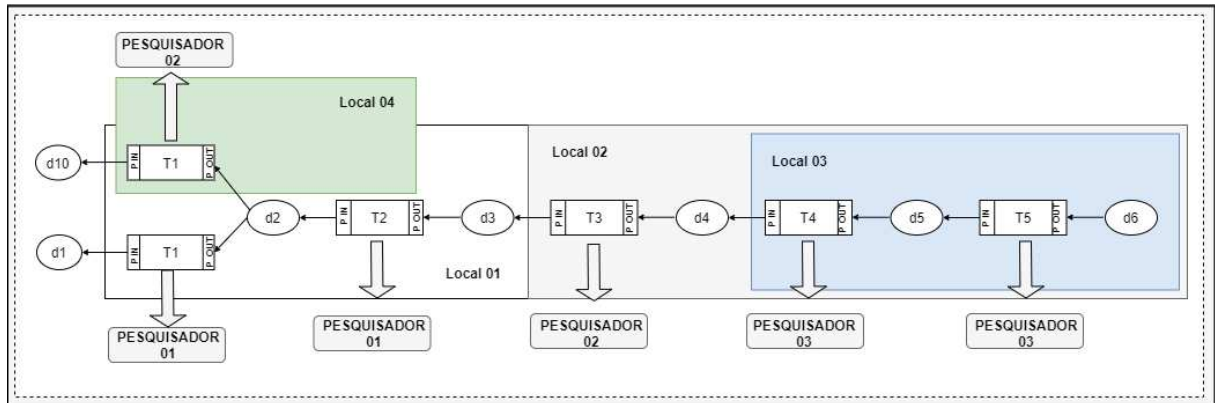
Fonte: Elaborada pelo autor.

3.2.2 Camada *Wrapper*

Cientistas, em experimentos colaborativos, podem realizar parte do experimento em ambientes heterogêneos, ou seja, utilizando diferentes SWMS. Esses SWMS geralmente expressam proveniência em diferentes formatos de dados. Assim, a camada *Wrapper* traduz e integra os dados heterogêneos de proveniência, vindos de diferentes SWfMS, para

o formato do modelo ProvONE, que é utilizado como modelo padrão e integrador na BlockFlow.

Figura 23 - Encadeamento de tarefas em um *workflow* científico.



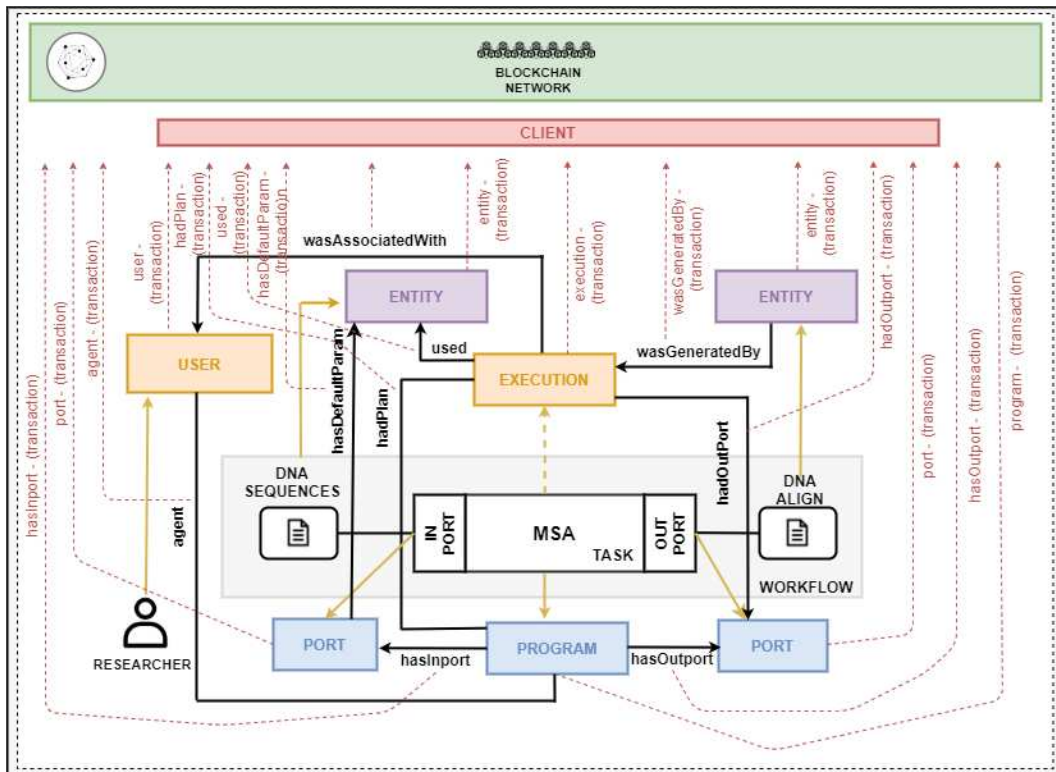
Fonte: Elaborada pelo autor.

Conforme apresentado na Figura 23, um *workflow* científico, pode ser visto como um grafo direcionado, cujos nós, são suas tarefas ($t - t_n$). Cada tarefa (t) do *workflow*, representa uma etapa computacional e possui um conjunto de portas de entrada (ip) e de saída (op). Essas tarefas consomem dados (di) como parâmetros, em suas portas de entrada, e produzem dados (do) vinculados às portas de saída. As arestas denotam como esses valores fluem de uma tarefa para outra e representam dependências de dados entre essas tarefas.

Como já dito, na Blockflow, uma tarefa, pode ser executada, entre pesquisadores geograficamente distribuídos. Considerando como base o *workflow* apresentado na Figura 22, as tarefas T1 e T2 podem ser executadas pelo pesquisador O1 no local L1, enquanto a tarefa T3 pelo pesquisador O2 no local L2, ou mesmo a tarefa T1 pode ser executada pelo pesquisador 01 e 02 com diferentes tipos de dados. Na BlockFlow, o mapeamento de proveniência para o modelo ProvONE (Figura 24), se dá observando a invocação de tarefas e principalmente o ciclo de vida dos conjuntos de dados consumidos ou produzidos, durante a execução do *workflow*.

A captura de informações de proveniência é feita através de um serviço *web*. Cada tarefa é instrumentalizada com esse serviço, para capturar as informações de entrada e de saída da tarefa. A Figura 24 mostra uma representação do mapeamento de proveniência de uma tarefa, pertencente a um *workflow*, para o modelo ProvONE através da camada *wrapper*. A tarefa do *workflow* apresentado na Figura 24, consiste em uma atividade de Alinhamento Múltiplo de Sequência (AMS), que recebe como entrada um arquivo multi-fasta contendo sequências de *DNA*, e gera como saída o alinhamento dessa sequência.

Figura 24 - Exemplo de mapeamento de uma tarefa de um *workflow* para modelo ProvONE.

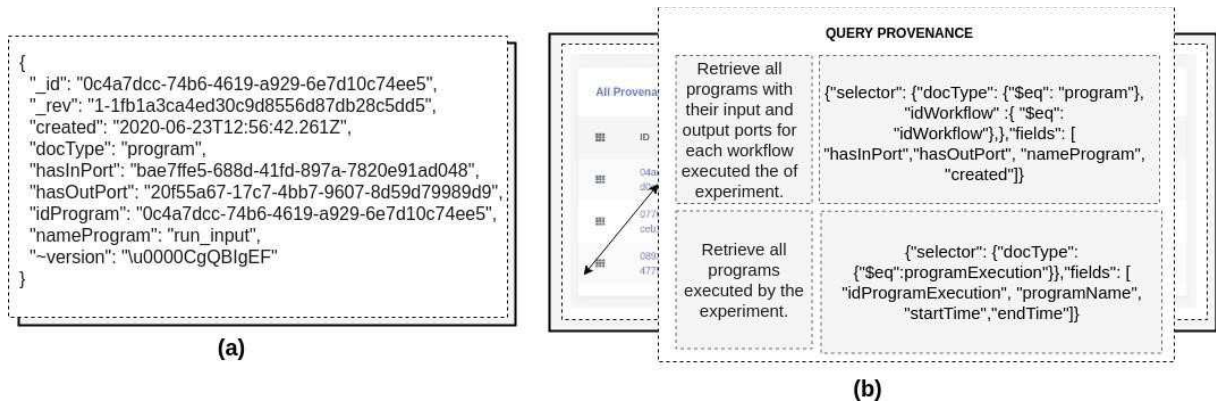


Fonte: Elaborada pelo autor.

Considerando o modelo ProvONE, suas classes e relacionamentos, conforme anteriormente apresentado na Figura 24, na BlockFlow, a tarefa (AMS) é mapeada para a classe *Program* do modelo ProvONE, e suas portas de entrada e saída para a classe *Port*, onde o relacionamento *hasInport* e *hasOutport*, as relacionam, respectivamente com um *Program* (as setas tracejadas em amarelo expressam a correspondência, entre os elementos do *Workflow* e as classes do modelo ProvONE). A tarefa quando executada é mapeada para a classe *Execution*, e o arquivo de entrada “DNA SEQUENCES” é mapeado como uma *Entity*, que expressa o relacionamento *hasDefaultParam* para uma *Port* de entrada e que é *used* por uma *Execution*. Para reduzir o volume de dados de proveniência, armazenamos o *hash*, i.e., o caminho de entrada e dados de saída, em vez dos dados como um todo, no *blockchain*. O arquivo *DNA ALIGN*, mapeado como uma *Entity*, gerado como saída é *wasGenerationBy* pela *Execution* da tarefa MSA. A execução, relacionada a classe *Execution*, da tarefa (MSA) *wasAssociatedWith* a um pesquisador, mapeado para a classe *User*, que por sua vez *hadPlan* com uma instância da classe *Program*.

A camada **Wrapper**, após o mapeamento, envia cada uma dessas informações de proveniência coletadas, (classes e relacionamentos) do modelo ProvONE para camada **Client**, que então as envia como transações para a *Blockchain Network*. A **Blockchain**

Figura 25 - Exemplo de mapeamento de uma tarefa de um *workflow* para modelo ProvONE.



Fonte: Elaborada pelo autor.

Network então, após uma série de transformações, registra cada transação de proveniência no sistema de arquivos *blockchain* e armazena no banco de dados de estado, *CouchDB*¹. A Figura 25 (a) é a representação desse mapeamento no formato JSON e exemplos de consultas que podem ser realizadas Figura 25 (b). A Tabela 8 apresenta a correspondência entre o mapeamento de um conjunto de tarefas para o Modelo ProvONE.

Tabela 8 – Correspondência entre o mapeamento de um conjunto de tarefas para o Modelo ProvONE

Mapping Workflow	ProvONE:Class	ProvONE:Association
Workflow	Workflow	<i>wasDerivedFrom</i>
task	program	<i>hadPlan</i>
task.execution	execution	<i>hadPlan, hadOutPort</i>
task.port.input	port	<i>hasInport</i>
task.port.output	port	<i>hasOutport</i>
data.input	entity	<i>used, hasDefaultParam</i>
data.output	entity	<i>wasGeneratedBy, used</i>

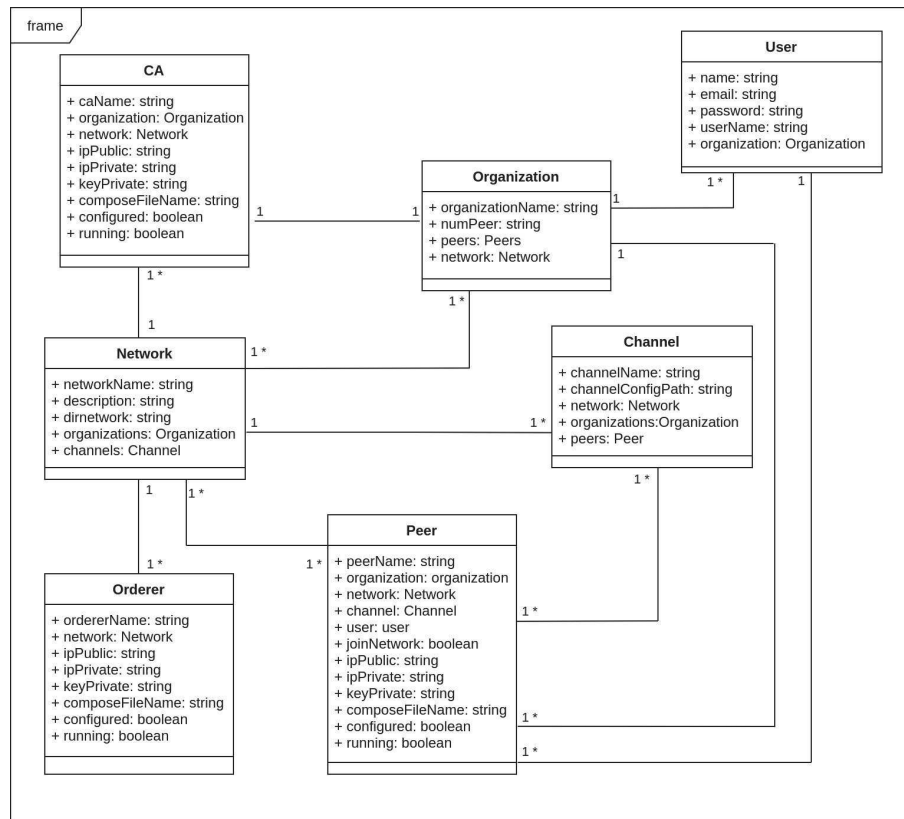
3.2.3 Camada Model

Na BlockFlow esta camada define o modelo que persiste o conjunto de informações relacionadas ao fluxo de configuração e chamadas da API BlockFlow. Essas informações englobam dados da criação de redes para um experimento, quais são os cientistas (*peers*) que irão colaborar em um experimento, quem colabora no experimento, dados de cadastro

¹ <https://couchdb.apache.org/>

de usuário, dados de cadastros e configurações de redes *blockchain*, canais. A Figura 26 representa este modelo de dados.

Figura 26 - Modelo de classes da arquitetura BlockFlow.

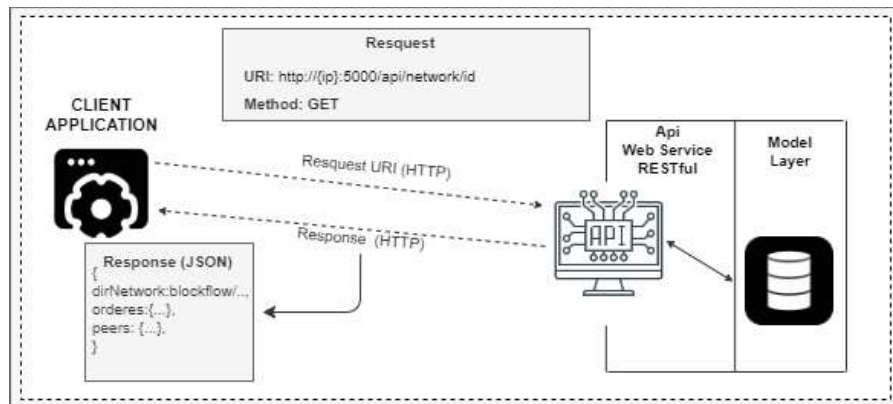


Fonte: Elaborada pelo autor.

Com o objetivo de detalhar o funcionamento desta camada, apresentamos a seguir o detalhamento dos modelos apresentados na Figura 26. A classe *Network*, armazena um conjunto de informações referentes a criação de uma rede (*blockchain*) para um dado experimento, tais como, nome da rede, descrição, diretório. Esse conjunto de informações se relaciona com um conjunto de instâncias, como organizações e *peers*, que armazenam informações, tais como, nome dos *peers* que fazem parte do experimento, organizações a qual ele pertence, entre outros. Posteriormente estas informações armazenadas podem ser acessadas, para consultas e configurações da rede por um pesquisador.

A Figura 27 apresenta um diagrama ao qual a API *Restful*, através de uma requisição, solicita um recurso a camada de dados. A solicitação ao serviço da *web* é através da URI: “http://ip:5000/api/network/id”, a qual é solicitado informações de uma rede (experimento), pelo seu identificador único “id”. Os dados são retornados em formato JSON, o que permite que o pesquisador possa visualizar as informações persistidas sobre uma rede criada. Esses dados englobam diretório, canais, orderes, CAs, organizações e *peers* persistidos que fazem parte de um experimento. Os dados retornados em formato

Figura 27 - Digrama de solicitações e respostas para camada de dados da BlockFlow.



Fonte: Elaborada pelo autor.

JSON, podem compor uma visualização em uma página *web*, conforme a Figura 28. Essa Figura ilustra informações de uma rede criada, com informações de seus *peers* (nome do *peer*, ip público, ip privado, dns público, entre outros) *orderes* e CAS.

Figura 28 -Interface do usuário, construída através de forma JSON.

Network Experiment Created										
Network Components										
Peer	Organization	Public IP	Private IP	Public DNS	Key private	Docker File	Configured	Update	Check the Configuration	
peer0.UFJF.com	UFJF	3.21.240.14	172.31.26.8	ec2-3-21-240-14.us-east-2.compute.amazonaws.com	peer0.UFJF.com/blockchain-vm1.pem	docker-compose-peer0-UFJF.yaml	false	Update	Check	
peer0.UFRJ.com	UFRJ	13.59.47.146	172.31.18.179	ec2-13-59-47-146.us-east-2.compute.amazonaws.com	peer0.UFRJ.com/blockchain-vm2.pem	docker-compose-peer0-UFRJ.yaml	false	Update	Check	
peer0.UFF.com	UFF	18.222.137.146	172.31.29.195	ec2-18-222-137-146.us-east-2.compute.amazonaws.com	peer0.UFF.com/blockchain-vm3.pem	docker-compose-peer0-UFF.yaml	false	Update	Check	
peer0.UFMG.com	UFMG	18.188.77.99	172.31.21.18	ec2-18-188-77-99.us-east-2.compute.amazonaws.com	peer0.UFMG.com/blockchain-vm4.pem	docker-compose-peer0-UFMG.yaml	false	Update	Check	

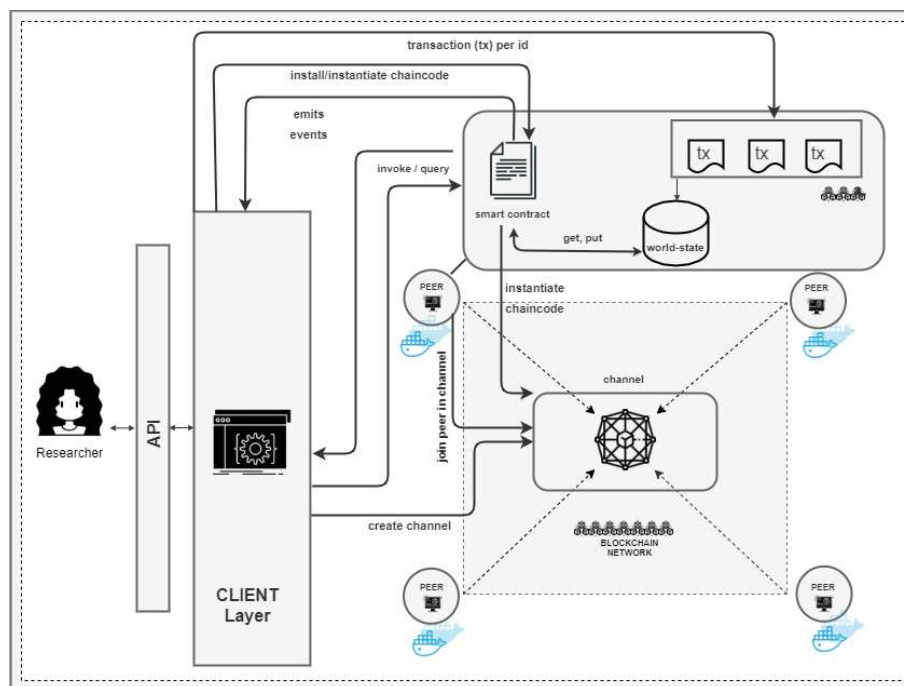
CREATE CONFIGURATION OF NETWORK

Fonte: Elaborada pelo autor.

3.2.4 Camada *Client*

A camada *Client* permite que os aplicativos conectem-se a rede *blockchain* e aos nós para que então possam interagir com o *ledger* (razão). De uma forma geral, essa camada permite que os aplicativos clientes se conectem a rede e invoquem códigos de chamada ao razão, como, por exemplo, chamadas de consultas, chamadas para invocar transações, entre outros. Essa camada é composta, por um conjunto de métodos, detalhados a seguir.

Figura 29 - Fluxos de chamadas a camada *Client*.



Fonte: Elaborada pelo autor.

A Figura 29 ilustra fluxos de chamadas a camada *Client*, ao qual um pesquisador pode fazer solicitações, tais como:

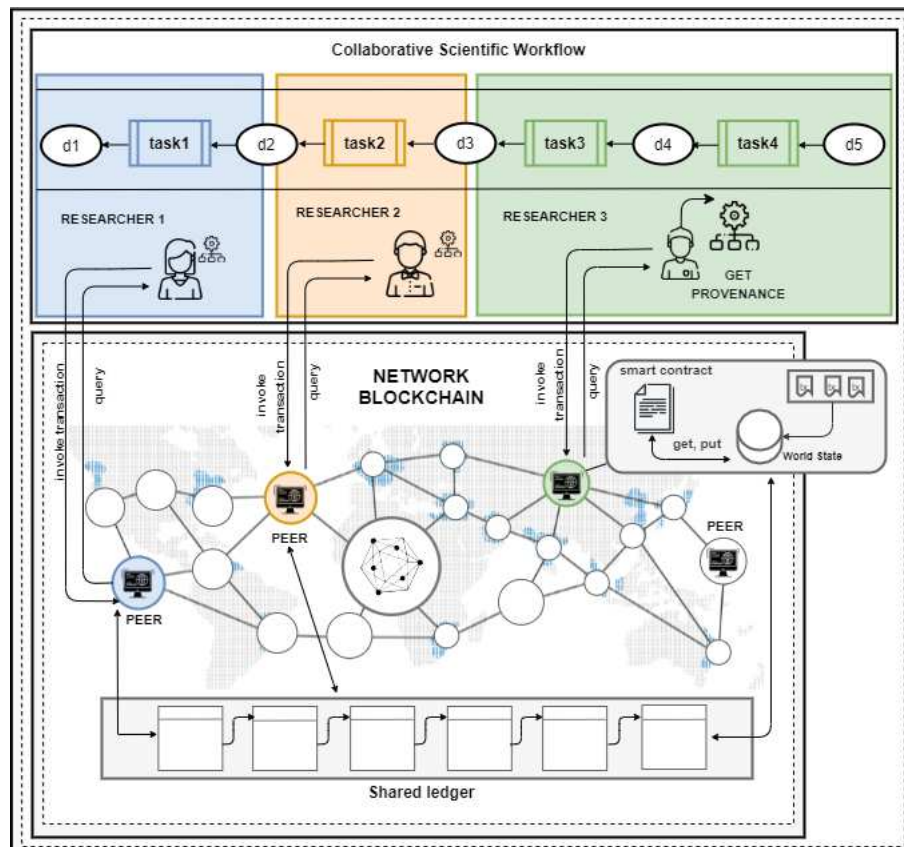
- Enviar registro de proveniência: a partir da invocação do método *invokeTransaction()*, a camada *Client* se conecta ao *peer* para atualizar o *ledger*. Do mesmo modo, quando um pesquisador necessita recuperar informações de proveniência o método *queryTransaction()* é invocado. A camada *Client*, após receber a solicitação de chamada ao método, se conecta ao *peer*, para recuperar as informações de proveniência no *ledger*.
- Instalar ou instanciar *chaincodes*: para que um *peer* possa enviar ou ler transações, é necessário que um *chaincode* esteja instalado em seu *peer*, que é instanciado no canal ao qual ele faz parte. Para isso, o método chamado *installChaincode()* é invocado, e a camada *Client* se conecta ao *peer* e instala o *chaincode*. Do mesmo

modo, ao receber a solicitação para instanciar um *chaincode* em um canal, o método *instantiateChaincode()* é invocado, e a camada *Client* se conecta a rede e ao canal para instanciar o *chaincode*.

3.2.5 Camada Blockchain Network

A camada de rede *Blockchain* representa um *workflow* colaborativo, cujos nós, são pesquisadores geograficamente distribuídos, que podem pertencer a diferentes instituições de pesquisa, conectados através de instâncias de máquinas locais ou de máquinas virtuais na nuvem. A Figura 30 ilustra uma representação de um *workflow* colaborativo, onde cada pesquisador pertencente a um experimento e está conectado a um nó de uma rede *blockchain*.

Figura 30 - Rede *blockchain*, *workflow* científico colaborativo.



Fonte: Elaborada pelo autor.

As diferentes tarefas presentes no *workflow* colaborativo, podem ser executadas entre os pesquisadores geograficamente distribuídos. Estes pesquisadores coletam proveniência, que então são enviadas como transações para rede *blockchain*. Nas redes *blockchains*, todos os dados de proveniência coletados entre os pesquisadores que colaboram em um experimento, serão armazenados na forma de blocos, e distribuídos entre os outros pares.

Cada nó participante da rede têm sua própria cópia do *ledger* permitindo assim, auditoria, consulta transparente de todos os dados coletados, processados e acessados por diferentes *workflows* executados nos diferentes nós geograficamente distribuídos.

Um sistema de proveniência baseado em *blockchain* para experimentos científicos colaborativos pode levar a um ambiente confiável de experimentação científica, uma vez que a proveniência coletada não pode ser manipulada sem deixar um rastro. Essa camada garante transparência, imutabilidade e confiabilidade para dados de proveniência científica.

3.2.6 Tecnologias de Desenvolvimento

Para implementar a arquitetura, a BlockFlow foi dividida em dois módulos, on-chain e off-chain.

O Módulo off-chain é composto pelo serviço da *web* da **API RESTful**, pelas camadas **Client**, **Wrapper** e **Model**. O serviço *web* da **API RESTful** e a Camada **Wrapper** foram implementados usando a tecnologia *Node.js*². A camada **Client** foi implementada usando o *Hyperledger Fabric* SDK para *Node.js*³ para interagir com a **Blockchain Network**. A camada **Model** foi implementada usando o banco de dados *MongoDB*⁴.

O módulo on-chain foi implementado usando a plataforma *Hyperledger Fabric*, todos os módulos da arquitetura do *Hyperledger Fabric*⁵ funcionam com base na tecnologia de *containers Dockers*⁶ e foram especificados em arquivos *yaml*⁷ e são inicializados usando a ferramenta *docker-compose*⁸.

3.3 BLOCKFLOW EM AÇÃO

Para verificar a viabilidade técnica de uso da arquitetura BlockFlow na captura, armazenamento e compartilhamento seguro de informações de proveniência, apresentamos a seguir um passo a passo de uso da arquitetura, considerando um experimento científico simulado. No capítulo 4, uma avaliação envolvendo o uso da BlockFlow será apresentada, com o objetivo de verificar a viabilidade da arquitetura e responder à questão de pesquisa dessa dissertação.

Projetos de pesquisa científica atuais são essencialmente de natureza colaborativa. Nesses projetos, membros da equipe geralmente residem em locais geograficamente distribuídos, permitindo assim o uso de uma inteligência coletiva. Nesta dissertação, propomos

² <https://nodejs.org/en/>

³ <https://hyperledger.github.io/fabric-sdk-node/release-1.4/module-fabric-network.html>

⁴ <https://www.mongodb.com/>

⁵ <https://www.hyperledger.org/>

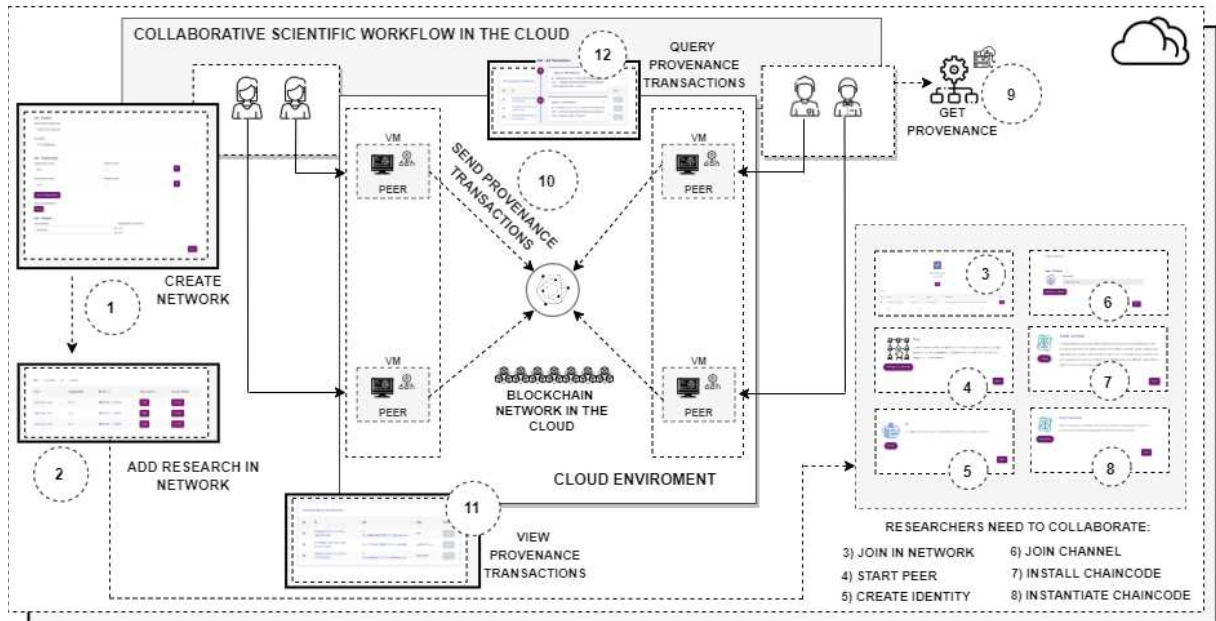
⁶ <https://www.docker.com/>

⁷ <https://yaml.org/>

⁸ <https://docs.docker.com/compose/>

uma abordagem através da qual cientistas geograficamente distribuídos podem colaborar, integrar dados de proveniência e resolver os problemas considerando a análise de dados heterogêneos. Na BlockFlow para colaborar, armazenar e compartilhar informações é necessário seguir um fluxo de ações, conforme detalhado na Figura 31.

Figura 31 - Fluxos de ações para criação de ambiente colaborativo da arquitetura BlockFlow.



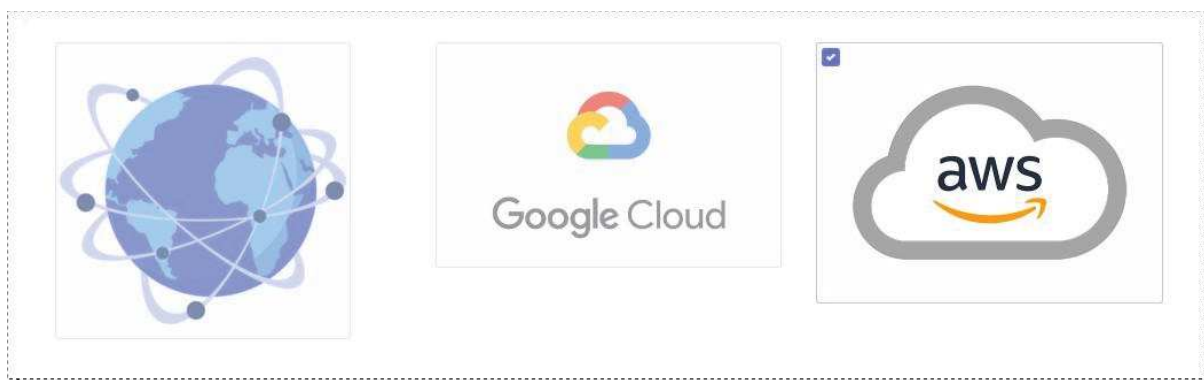
Fonte: Elaborada pelo autor.

Para colaborar em um experimento, cientistas devem definir como primeiro passo, as redes *blockchain*. As redes, na BlockFlow, podem ser locais ou instâncias de máquinas virtuais na nuvem, dependendo da especificidade e das necessidades do experimento. Experimentos científicos intensivos em dados, geralmente, necessitam de muitos recursos, necessitando, portanto, de ambientes colaborativos ou de alto desempenho, como os ambientes de computação em nuvem.

Neste sentido, na BlockFlow, cientistas podem provisionar instâncias de máquinas virtuais como no *Amazon Elastic Compute Cloud* (Amazon EC2). Esta é uma maneira para que os cientistas possam garantir uma infraestrutura escalável e robusta, além de uma variedade de recursos, como *hardware* e *software*, sob elasticidade (capacidade de adicionar ou remover dinamicamente recursos computacionais para atender à demanda do experimento), sem a necessidade de adquirirem infraestruturas computacionais. A Figura 32 apresenta a tela da *interface* da BlockFlow, para a escolha entre redes locais e/ou redes na nuvem.

A criação de uma rede *Blockchain* pode ser um processo demorado, que envolve o

Figura 32 - Interface do usuário, para a escolha entre redes locais ou redes na nuvem, na arquitetura BlockFlow.



Fonte: Elaborada pelo autor.

entendimento de conceitos, implementação de tecnologias e seu uso. No entanto, a criação de uma rede *blockchain* na BlockFlow é transparente para o usuário (geralmente cientistas). Para isso, na BlockFlow, um implementador *Hyperledger Fabric* baseado em GUI permite que os pesquisadores implementem suas redes *blockchain* para então colaborarem. Cada nó na rede é uma instância que representa um cientista na plataforma E-SECO. Toda a configuração de rede *Blockchain* é automática e sem intervenção do usuário. Portanto, cientistas não precisam se preocupar com as configurações e tarefas meticulosas entre os diferentes componentes da tecnologia *blockchain* usada. A Figura 33 apresenta a tela do *frontend* da BlockFlow, para a criação de redes *blockchains* e mais adiante é apresentado o fluxo para criação da rede.

Nesse ambiente, a definição de organizações e seus pares podem ser entendida como um consórcio. Os consórcios são especificados como um canal, permitindo o compartilhamento e a transação (troca de informações) apenas entre as partes interessadas. Os canais na rede *Hyperledger Fabric* podem ser entendidos como uma rede de sobreposição privada, onde um conjunto específico de nós concordam em colaborar e compartilhar informações. Nesse ambiente, os pesquisadores podem definir o canal como “Mychannel”, onde “Org1” e “Org2” fazem parte deste e a partir desta especificação, podem criar a rede.

Após criarem a rede, os pesquisadores podem designar quais pesquisadores específicos serão os nós, pares, e farão parte do canal na rede *blockchain*. Assim os pesquisadores podem especificar um pesquisador como “peer0.org1.com” da organização “Org1”, um pesquisador como “peer0.org2.com” em “Org2” e por último um pesquisador como “peer1.org2.com” da “Org2”. A partir dessa configuração, cada pesquisador estará vinculado a um nó na rede *blockchain*. A Figura 34 apresenta a tela do *frontend* da BlockFlow, onde pesquisadores podem designar quais pesquisadores específicos serão os nós, pares e farão

Figura 33 - Tela do FrontEnd, para criar redes *blockchains* na arquitetura BlockFlow.

The screenshot shows a web interface for creating a blockchain network. It is titled "Network Experiment" and is organized into three main sections:

- Network:** Contains a "Name Network Experiment" text input field and a "Description" text area.
- Organizations:** Contains two text input fields: "Organizations Name" and "Number of Peer". There is a purple "Add new organization" button below these fields. At the bottom of this section is a "Save all organizations" label with a purple "Save" button.
- Channel:** Contains a "Channel Name" text input field and an "Organizations in channel" text input field.

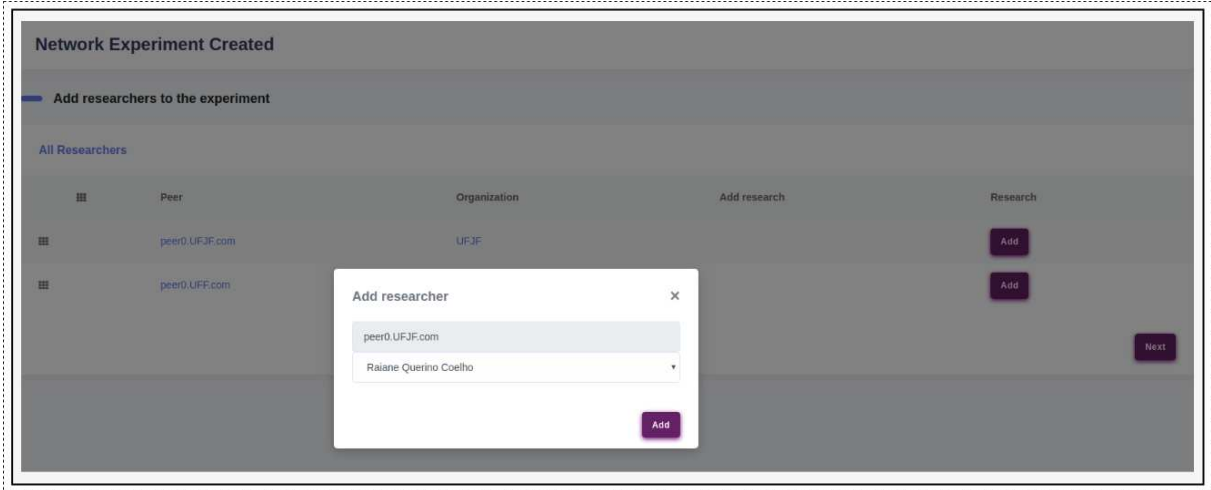
A purple "Next" button is located at the bottom right of the interface.

Fonte: Elaborada pelo autor.

parte do canal na rede *blockchain*.

Para colaborar e realizar transações na rede, os pesquisadores precisam: (i) Participar da rede: quando definido como nó, os pesquisadores precisam ingressar na rede especificada. (ii) Iniciar pares: os pares são os principais componentes da rede porque eles hospedam o *ledger* (razão) e o *chaincode*. Cada pesquisador distribuído geograficamente deve iniciar suas instâncias de nós, que neste caso são componentes de contêiner *docker*. (iii) Criar suas identidades: os pesquisadores que desejam realizar transações em um canal da rede precisam que seu nó se registre e tenha uma identidade na rede. (iv) Participar do canal: os pesquisadores que desejam realizar transações em um canal precisam que seus pares estejam associados a esse canal. (v) Instalar *chaincode*: o *chaincode* gerencia e inicializa o estado do razão, enviando transações. Os pesquisadores que desejam realizar transações e ler dados do razão precisam instalar um *chaincode* em seu nó. (iv) Instanciar *chaincode*: após a instalação dos *chaincodes* no nó, o *chaincode* deve ser instanciado no

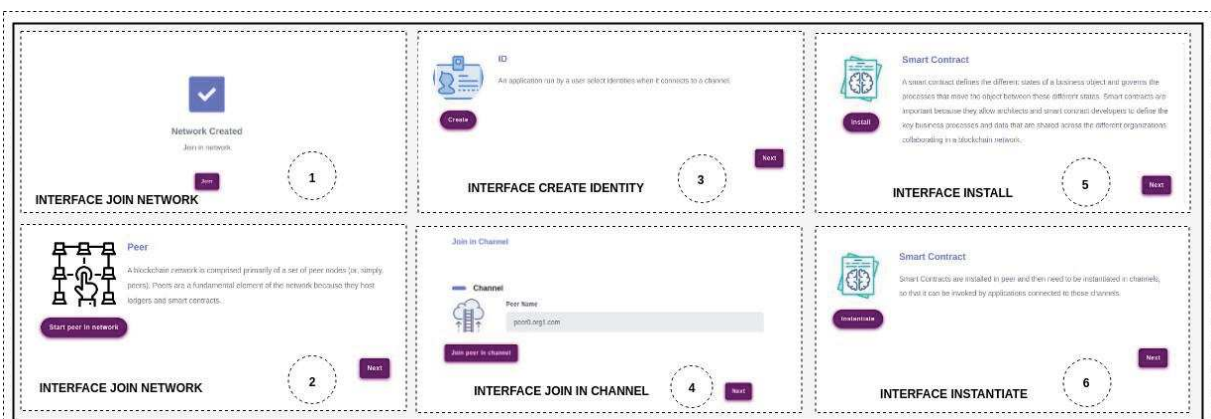
Figura 34 - Tela do FrontEnd, para especificar quais pesquisadores serão nós pares e farão parte do canal na rede *blockchain*, na arquitetura BlockFlow.



Fonte: Elaborada pelo autor.

canal ao qual o par está associado. Dessa maneira, o par pode interagir com o razão através do *chaincode*. A Figura 35 sumariza todo fluxo entre as *interfaces* de usuário presente na BlockFlow, conforme a descrição.

Figura 35 - Sumarização do passo a passo a partir do FrontEnd da BlockFlow.



Fonte: Elaborada pelo autor.

Uma vez configurada a rede, os pesquisadores podem colaborar em ambiente confiável, incluindo o gerenciamento dos dados de proveniência do experimento a ser conduzido. Em experimentos colaborativos e distribuídos, pesquisadores podem executar e

analisar experimentos em diferentes SWfMS, tais como, Kepler, Taverna e Vistrails, entre outros.

Com objetivo de capturar dados de proveniência e interoperar esses dados entre os pesquisadores que colaboram no experimento, a BlockFlow fornece um serviço *web*. Essa captura é feita em tempo real e é independente do SWfMS. Para capturar os dados, é necessário instrumentalizar com o serviço *web* fornecido pela BlockFlow, cada atividade (tarefa) do *workflow*. Dados de entrada e saída de cada atividade são coletados, assim como outras informações relevantes ao contexto do ambiente em que o experimento é realizado, conforme exemplo detalhado na descrição da camada *Wrapper*.

No cabeçalho de cada serviço *web*, o *token* do usuário recebido durante a autenticação deve ser enviado ao sistema. Essa é uma maneira de confirmar a autenticidade e recuperar a identidade do usuário como um nó pertencente à rede *blockchain*, além de garantir que os dados sejam posteriormente assinados e negociados na rede. Se a identidade do usuário não puder ser verificada, as informações serão rejeitadas e não consideradas, e os dados de proveniência não serão registrados.

Após coletar dados de proveniência entre os pesquisadores geograficamente distribuídos, esses dados são convertidos para o modelo de dados ProvONE. Na sequência, esses dados são armazenados no *blockchain*. Os dados são enviados para rede através de transações por meio de contratos inteligentes (*chaincode*). Um fragmento do contrato inteligente implementado na arquitetura BlockFlow pode ser visualizado na Figura 35.

Após essa etapa, os dados de proveniência são enviados como transações para rede *blockchain*. Essas transações passam por um fluxo, que pode ser compreendido em: i) Cada transação é armazenada em uma lista que formará um bloco. Cada bloco será validado e então será armazenado no razão, formando uma cadeia. ii) Essa cadeia será compartilhada entre os vários pares da rede, sendo possível visualizar todo fluxo de informações. iii) Esse fluxo de dados garante que não haja um único ponto de falha e que os dados de proveniência sejam transparentes, ou seja, visíveis a todos os pesquisadores no experimento. Além de disso garante imutabilidade e confiança aos dados de proveniência.

3.3.1 Análise de Desempenho

Nossa abordagem visa garantir um ambiente colaborativo, distribuído e confiável para experimentação científica. Para atingir esse objetivo, integramos e armazenamos dados de proveniência durante a execução de um fluxo de trabalho científico, ancorado através de uma rede *blockchain* na nuvem.

Em muitos cenários de fluxo de trabalho científico de uso intensivo de dados, como no cenário proposto nesta dissertação, que será detalhado no próximo capítulo, podemos ter centenas, até dezenas, de registros de proveniência durante a execução de um fluxo de trabalho. Além disso, em fluxo de trabalho científico colaborativo, os pesquisadores

Figura 35 - Fragmento Chaincode da BlockFlow.

```

E: > blockflow > chaincode > blockflow-app > blockflow-chaincode.go
1  package main
2
3  import (
4      "bytes"
5      "encoding/json"
6      "fmt"
7      "strings"
8      "github.com/hyperledger/fabric/core/chaincode/shim"
9      sc "github.com/hyperledger/fabric/protos/peer"
10 )
11
12 // Define the Smart Contract structure
13 type SmartContractBlockflow struct{
14 }
15
16 type program struct{
17     ObjectType string `json:"docType"`
18     IdProgram string `json:"idProgram"`
19     NameProgram string `json:"nameProgram"`
20     Created string `json:"created"`
21     HasOutPort string `json:"hasOutPort"`
22     HasInPort string `json:"hasInPort"`
23 }
24
25 func (s *SmartContractBlockflow) recordProgram(APIStub shim.ChaincodeStubInterface, args []string) sc.Response {
26
27     if len(args) != 5 {
28         return shim.Error("Incorrect number of arguments. Expecting 4 recordProgram")
29     }
30
31     docType := "program"
32     idProgram := args[0]
33     nameProgram := strings.ToLower(args[1])
34     dataCreated := args[2]
35     hasOutPort := args[3]
36     hasInPort := args[4]
37
38     program := &program{docType, idProgram, nameProgram, dataCreated, hasOutPort, hasInPort}
39
40     programAsBytes, _ := json.Marshal(program)
41     err := APIStub.PutState(args[0], programAsBytes)
42     if err != nil {
43         return shim.Error(fmt.Sprintf("Failed to record program catch: %s", args[0]))
44     }
45
46     return shim.Success(nil)
47 }

```

Fonte: Elaborada pelo autor.

geralmente armazenam ou consultam simultaneamente o repositório de proveniência, seja para monitorá-lo ou para planejar ações futuras. Assim, neste contexto, são necessários mecanismos eficientes tanto para o armazenamento como para a consulta dos dados de proveniência. Assim, com objetivos de verificar se a solução proposta atende à essas necessidades, avaliamos a arquitetura BlockFlow em termos de desempenho.

A avaliação foi conduzida em uma instância de VM no *Amazon Elastic Compute Cloud* (EC2) com Intel (R) Xeon (R) CPU E5-2690, 2,60 GHz, CPU de 24 núcleos, 24 GB de RAM rodando Ubuntu 16.04., o *benchmark Hyperledger Caliper* e a Versão do Hyperledger Fabric 1.4.1⁹. O Hyperledger Caliper¹⁰ é fornecido pelo projeto *Hyperledger* e é uma ferramenta de *benchmarking* usada para medir o desempenho de *blockchains*.

Algumas métricas suportadas pelo Hyperledger Caliper são:

- Taxa de transferência da transação (*Transaction Throughput*): essa taxa indica o número de transações enviadas, válidas e confirmadas na rede *blockchain* por

⁹ <https://hyperledger-fabric.readthedocs.io/en/release-1.4/whatis.html>

¹⁰ <https://hyperledger.github.io/caliper/>

segundo.

- Latência de transação: indica o tempo que uma transação leva para estar disponível em toda a rede, esta métrica é calculada por transação. O *Caliper* mede a latência através de três métricas:
 1. latência mínima de uma transação.
 2. latência máxima de uma transação.
 3. latência média de todas as transações.
- Taxa de envio (*Send Rate*): é a taxa de envio real do *Hyperledger Caliper*, com base no TPS de destino.

Assim, avaliamos nossa proposta de acordo com as métricas especificadas, variando a carga de trabalho das transações (10 a 10.000), entre as solicitações, (gravação / invocar e consultar) a partir de dados de proveniência, no livro-razão realizado por um conjunto de pares simultaneamente. Os resultados são apresentados através das Tabelas 9, 10, 11.

Tabela 9 – Taxa de transferência da transação

Type Transaction	10	100	1000	10000
Invoke	4.1(s)	4.7(s)	5.0(s)	5.0(s)
Query	6.2(s)	5.1(s)	5.0(s)	5.0(s)

Tabela 10 – Latência de transações

Type	Carga de Trabalho	Latência mínima	Latência máxima	Latência média
Invoke	10	0.82(s)	2.42(s)	1.62(s)
Invoke	100	0.79(s)	3.45(s)	1.40(s)
Invoke	1000	5.1(s)	5.0(s)	5.0(s)
Invoke	10000	0.35(s)	3.63(s)	1.30(s)
Query	10	0.01(s)	0.02(s)	0.01(s)
Query	100	0.01(s)	0.02(s)	0.01(s)
Query	1000	0.01(s)	0.01(s)	0.01(s)
Query	10000	0.01(s)	0.11(s)	0.01(s)

Tabela 11 – Taxa de envio

Type	10	100	1000	10000
Invoke	6.3(s)	5.1(s)	5.0(s)	5.0(s)
Query	6.3(s)	5.1(s)	5.0(s)	5.0(s)

Após a análise, obtivemos evidências de que o sistema pode operar em baixa latência mesmo se tratando de grandes conjuntos de dados de proveniência. Este resultado fornece evidências iniciais de que podemos oferecer escalabilidade e eficiência em ambientes distribuídos de experimentação científica.

3.4 DISCUSSÕES

A arquitetura BlockFlow é uma arquitetura baseada em *blockchain*, cujo objetivo é permitir que cientistas trabalhem de maneira colaborativa e distribuída, organizando e compartilhando dados de proveniência de uma maneira mais confiável, com intuito de reprodutibilidade científica. Este capítulo apresentou o desenvolvimento da abordagem proposta onde foi detalhada cada uma de suas camadas, seu funcionamento e tecnologias utilizadas para o seu desenvolvimento. Através de um exemplo, ilustrando o passo a passo de um experimento científico simulado, foi possível verificar a sua viabilidade e algumas de suas funcionalidades, considerando o suporte a interoperabilidade, confiabilidade, transparência, segurança e reprodutibilidade.

Para responder à questão de pesquisa proposta nesta dissertação, **“Como a arquitetura BlockFlow pode auxiliar cientistas nos experimentos científicos colaborativos, oferecendo um ambiente confiável apoiando a interoperabilidade, privacidade, transparência e reprodutibilidade de *workflows* científicos?”**, no próximo capítulo apresentamos a condução de uma Prova de Conceito, considerando um experimento científico distribuído.

4 AVALIAÇÃO DA ARQUITETURA BLOCKFLOW

4.1 INTRODUÇÃO

Este capítulo apresenta uma avaliação inicial da arquitetura BlockFlow, a partir de uma Prova de Conceito (PoC). Uma PoC tem o intuito de verificar a aplicabilidade de um conceito teórico em um cenário prático (BELL, 1993). No contexto de uma arquitetura, a prova de conceito engloba a implementação desta arquitetura e a verificação dos seus efeitos, na prática. As provas de conceito normalmente podem ser divididas sob a perspectiva do que exploram: estrutura e comportamento. As que exploram estrutura podem ser usadas para comparar tecnologias e/ou ferramentas, enquanto as provas de conceito que exploram comportamento investigam um cenário de utilização. Nesta dissertação, foi adotada a perspectiva que explora o comportamento em um cenário de utilização.

O objetivo da Prova de Conceito foi definido de acordo com a abordagem Goal/Question/Metric (GQM) (VAN SOLINGEN et al., 2002). Os objetivos, segundo a abordagem GQM, devem ser formulados conforme o template a seguir:

“Analisar o **<objeto de estudo>** com a finalidade de **<objetivo>** com respeito à **<foco da qualidade>** do ponto de vista de **<perspectiva>** no contexto de **<contexto>**”.

Com base no template GQM, o objetivo da Prova de Conceito foi: Analisar a **arquitetura BlockFlow** com a finalidade de **avaliar sua viabilidade** com respeito ao suporte a **interoperabilidade, confiabilidade, transparência, privacidade e reprodutibilidade de *workflows* científicos** do ponto de vista de **equipes de cientistas geograficamente distribuídos** no contexto de **Experimentos Colaborativos**.

Com este escopo definido, derivamos a seguinte questão de pesquisa que nos guiará na condução da PoC:

(QP) **Como a arquitetura BlockFlow pode auxiliar cientistas nos experimentos científicos colaborativos, oferecendo um ambiente confiável apoiando a interoperabilidade, privacidade, transparência e reprodutibilidade de *workflows* científicos?**

A partir da questão de pesquisa derivamos questões secundárias que devem ser verificadas:

- (QS1) A BlockFlow pode fornecer uma visão geral dos dados de proveniência **de forma transparente**, onde pesquisadores geograficamente distribuídos, podem verificar como os dados de proveniência foram criados na cadeia (*blockchain*) ao longo do tempo?
- (QS2) A BlockFlow pode ser usada como um ambiente científico colaborativo e

confiável apoiando a **interoperabilidade** de dados de proveniência advindos de SWfMSs heterogêneos?

- (QS3) A BlockFlow pode ser usada como um **ambiente confiável** de troca de proveniência em *workflows* intensivos em dados?
- (QS4) A BlockFlow pode ser usada como um ambiente que fornece **privacidade** aos dados de proveniência, considerando a propriedade intelectual, onde os dados são compartilhados apenas entre partes ou pessoas autorizadas?
- (QS5) A BlockFlow pode ser usada como um ambiente de experimentação científica colaborativa, considerando a **reprodutibilidade** ?

4.2 CONTEXTUALIZAÇÃO

Ao longo dos anos, a ciência da computação deixou de ser uma ferramenta de apoio para se tornar um alicerce no processo de criação de conhecimento em diversas ciências, como, por exemplo, na Bioinformática. Os avanços tecnológicos proporcionaram um grande passo frente a ciência da biologia molecular e proporcionou a chamada era das ômicas (genômica, transcriptômica, proteômica, metabolômica) (VAILATI-RIBONI; PALOMBO; LOOR, 2017). Novas tecnologias de sequenciamento tornaram possível sequenciar milhares de genomas de diversos organismos, como, por exemplo, o genoma humano (LANDER et al., 2001), colocando assim o sequenciamento em larga escala ao alcance de muitos cientistas.

Um sequenciamento é a leitura do genoma ou transcriptoma de um organismo. Todos os organismos vivos são compostos por DNA, ou RNA. Tanto um como outro são formados por um conjunto de letras, bases nitrogenadas, que funcionam como um código (palavras), conhecidas como sequências de nucleotídeo (Adenina (A), Citosina (C), Timina (T) e Guanina (G) no caso do RNA temos a Uracila (U) no lugar da Timina) (ROBERTS et al., 2002). O processo de sequenciamento consiste na tarefa de descobrir, para um determinado organismo, qual é a sequência dessas bases nitrogenadas que forma cada fragmento de DNA ou RNA que está sendo investigado (ROBERTS et al., 2002).

De forma geral, projetos genoma e transcriptoma possuem suporte computacional, nos quais são projetados *workflows* que transformam fragmentos de entrada (sequências *reads* de fragmentos de RNA ou DNA), com objetivo de extrair informações biológicas, ou seja, a ordem das sequências de nucleotídeos ao longo da molécula de DNA ou RNA de um organismo (ANSORGE, 2009).

Identificar todas as bases que compõem o genoma é importante para aprender mais sobre o organismo e pode ajudar a cientistas a entender como os genes e as células funcionam. Através dos resultados obtidos a partir do sequenciamento de genomas ou transcriptoma,

podemos obter informações, por exemplo, sobre a linha evolutiva (Filogenética), de diversos organismos. A Filogenética pode ajudar a entender a relação evolutiva entre grupos de organismos, e pode resultar na melhor compreensão dos eventos evolutivos envolvidos em determinados genes (WILEY; LIEBERMAN, 2011).

Além disso, a Filogenética pode contribuir para novos métodos de diagnóstico, formulação de novos medicamentos, prevenção, vacinas, e tratamentos mais eficazes contra doenças. Uma vez que soubermos na linha evolutiva, ou seja, o quão perto ou distante uma sequência em estudo se encontra de outra bem conhecida, podemos, por exemplo, estender o tratamento de um terminada doença à outra causada por um determinado ancestral comum.

Como exemplo temos o sequenciamento genômico do novo coronavírus SARS-CoV-2, que a partir dessas técnicas foi possível identificar que este compartilha da homologia de sequência significativa com outros dois coronavírus, SARS (Coronavírus da Síndrome Respiratória Aguda Grave - SARS-CoV) e MERS (Síndrome Respiratória do Oriente Médio - (MERS-CoV)) (YAQINUDDIN, 2020). No entanto, reconstruir as relações entre os organismos vivos está longe de ser uma tarefa trivial além de envolver o processamento de grandes volumes de dados. Para facilitar a análise de dados genômicos e gerar uma árvore filogenética a partir de sequências de DNA, RNA e aminoácidos, vários *workflows* científicos foram projetados e estão disponíveis, tais como, SciPhy (OCAÑA et al., 2011) e SciEvol (OCAÑA et al., 2012) entre outros.

Como grande parte dos experimentos de bioinformática que possuem suporte de *workflows* científicos, principalmente na área de filogenética, dependem de grandes volumes de dados, é importante o suporte computacional adequado. A principal dificuldade para apoiar esses experimentos está no processamento de dados, demandando assim o uso de novas técnicas de computação como ambientes colaborativos, distribuídos ou de alto desempenho (HPC), como grades ou nuvens para sua execução (ZHAO et al., 2011). Além disso, no contexto de experimentação científica, a expectativa de que o experimento seja reproduzível é considerada fundamental. Apesar dessa importância, o suporte a reprodutibilidade em experimentos filogenéticos, segundo (MAGEE; MAY; MOORE, 2014), ainda é incipiente.

4.3 SARS-CoV2

Atualmente, estamos diante de uma epidemia global de um novo coronavírus, relatado pela primeira vez no início de dezembro de 2019 na província de Hubei (China). A COVID-19 (doença de coronavírus 2019) ou SARS-CoV-2, infectou milhares de pessoas e se espalhou rapidamente pelo mundo, causando grandes impactos sociais, econômicos e na área da saúde.

Os coronavírus são um grupo de vírus de RNA de fita simples de sentido positivo

que pertence ao gênero Betacoronavirus da família Coronaviridae (KIM, Dongwan et al., 2020; KUMAR, 2020; ZHOU, Hong et al., 2020;), um gênero que inclui muitos vírus que infectam seres humanos, aves, morcegos, animais domésticos e selvagens. Nas últimas duas décadas, diferentes cepas de coronavírus foram documentadas, incluindo as cepas dos coronavírus altamente patogênicos, como dos coronavírus da síndrome respiratória aguda grave (SARS-CoV) em 2002, coronavírus da síndrome respiratória do Oriente Médio (MERS-CoV) em 2012, e mais recentemente, um novo coronavírus chamado COVID-19 ou SARS-CoV-2 (KIM, Dongwan et al., 2020; ZHOU, Peng et al., 2020).

A rápida disseminação do SARS-CoV-2 demonstrou a necessidade de entender a sua homologia e como o vírus se espalha. Embora o diagnóstico imediato e o isolamento do paciente sejam pontos, chaves para o controle inicial desse novo surto, a utilização de modelos evolutivos e análises filogenéticas podem auxiliar a entender a sua origem, evolução e a desenvolver possíveis vacinas e medicamentos.

Com intuito de se obter informações sobre a transmissão, origens e de se estimar a variabilidade genética do novo coronavírus, foram relatados na literatura vários estudos de filogenética, com objetivo de quantificar e visualizar a relação das sequências virais do novo coronavírus (FORSTER et al., 2020; JAIMES et al., 2020; ZHANG, Tao et al., 2020; ZHOU Peng et al., 2020). É possível reconstruir a história evolutiva de um vírus a partir da identificação de mudanças nas sequências genéticas amostradas de diferentes pacientes, considerando que o vírus é transmitido através de uma população e acumula mutações em seu código genético. Desde os primeiros dias da pandemia, houve uma importante mobilização da comunidade científica para entender sua epidemiologia e ajudar a fornecer uma resposta. Equipes de pesquisa de várias partes do mundo sequenciaram maciçamente e publicaram sequências genômicas virais para o estudo sobre a origem do novo coronavírus. Essas várias sequências genômicas do vírus (SARS-CoV2) foram divulgadas publicamente, e estão presentes em muitos bancos de dados públicos incluindo NCBI¹, GISAID² e ViPR³.

4.4 PROVA DE CONCEITO

Dada a importância das pesquisas relacionadas ao SARS-CoV-2, a garantia de reprodutibilidade de seus achados e a confiabilidade dos dados, além da necessidade de colaboração entre cientistas do mundo todo, consideramos que o uso da BlockFlow e sua capacidade de suporte a interoperabilidade, privacidade, transparência, confiabilidade e reprodutibilidade dos experimentos pode auxiliar neste cenário.

Para verificar a viabilidade de uso da Blockflow nesse cenário, a partir da in-

¹ ncbi.nlm.nih.gov

² gisaid.org

³ viprbrc.org

investigação das questões de pesquisa referentes a solução proposta nesta dissertação, foi conduzida uma Prova de Conceito através de um experimento colaborativo (*workflow*) de análise filogenética com base em sequências completas de genoma de diferentes coronavírus, incluindo cepas dos coronavírus (SARS, MERS e SARS-CoV-2), que foram obtidas no GISAID e NCBI GenBank.

O tipo de Prova de Conceito utilizada foi a Pesquisa Histórica. A Pesquisa Histórica (YIN et al., 2014) mede a capacidade e a probabilidade de conclusão de um projeto, incluindo todos os fatores relevantes. Segundo (YIN et al., 2014), a Pesquisa Histórica é o método de avaliação recomendado para responder a questões explicativas quando praticamente não há controle sobre os eventos. A vantagem do método de pesquisa histórica é que ele não depende de observações diretas dos eventos. Em vez disso, as fontes de dados do estudo contam com documentos e artefatos como fontes primárias de evidência, como é o caso dos dados referentes ao experimento colaborativo que estamos avaliando. Usando a Pesquisa Histórica, criamos conhecimento sobre o uso da tecnologia (SHULL et al., 2004). O pesquisador pode avaliar se essa tecnologia atende aos objetivos inicialmente definidos, justificando a manutenção da pesquisa. Os conhecimentos obtidos com o uso da Pesquisa Histórica fornecem uma base para a proposição de refinamentos e a geração de novas hipóteses a serem investigadas em pesquisas futuras.

4.5 PLANEJAMENTO

4.5.1 Configuração do Ambiente

Os *workflows* utilizados nesta avaliação foram *workflows* científicos que no geral demandam grande capacidade de processamento e envolvem grande volume de dados, compondo um cenário ideal para avaliar a arquitetura Blockflow. Para os experimentos executados, foi utilizado o ambiente na nuvem da *Amazon Elastic Compute Cloud* (EC2)⁴. Foram instanciadas 4 tipos diferentes de máquinas virtuais, com diferentes características, a saber, capacidade de CPU e capacidade de memória RAM, além de estarem fisicamente distribuídas em diferentes regiões. A Tabela 12 resume os diferentes tipos de máquinas virtuais que foram utilizadas nos experimentos, com suas respectivas configurações de hardware.

Tabela 12 – Configuração Máquinas Virtuais.

VM	Descrição
Máquina virtual 1	EC2 ID: m4.large - 8 GB RAM, 2 núcleos.
Máquina virtual 2	EC2 ID: m4.xlarge - 16 GB RAM, 4 núcleos.

⁴ <https://aws.amazon.com/>

Máquina virtual 3	EC2 ID: m5.large - 8 GB de RAM, 2 núcleos.
Máquina virtual 3	EC2 ID: m5.xlarge - 16 GB RAM, 4 núcleos.

Para execução dos *workflows*, as seguintes versões de programas foram utilizadas: MAFFT versão 7.471⁵, Readseq versão 2.1.19⁶, ModelGenerator versão v0.85⁷, e RAxML e 8.2.12⁸ e para o ViReport: ViralMSA⁹ versão 1.0.6 utilizando o minimap2 versão v2.17, FastTree versão 2.1.11¹⁰, FastRoot¹¹ e LSD2¹². A Tabela 13 resume os *softwares* utilizados em cada instância de máquina virtual na nuvem, considerando a necessidade de configurar e inicializar os serviços necessários do ambiente colaborativo com base na tecnologia *blockchain* Hyperledger Fabric.

Tabela 13 – Software instalados nas máquinas virtuais.

Softwares
Sistema operacional Ubuntu Linux 18.04.1 LTS.
Docker Engine versão (18.06.1-ce)
Docker-Compose versão (1.13.0)
Node (v8.11.4)
Hyperledger Fabric (v1.4.1)
Go Lang — 1.12.0

A Figura 36 apresenta detalhes de cada instancia inicializada, rondado o ambiente colaborativo na nuvem.

4.5.2 *Workflows* utilizados na PoC

Embora experimentos de Análises Filogenética possam ser implementados de várias maneiras, nesta dissertação, para execução desta PoC, consideramos duas implementações de *workflows*, a saber, SciPhy (OCAÑA et al., 2011) e ViReport (SONG; MOSHIRI, 2020).

O Sciphy (OCAÑA et al., 2011) é um *workflow* científico de análise filogenética que foi projetado para gerar árvores filogenéticas com máxima verossimilhança. Este *workflow* é composto por cinco atividades:

⁵ <https://mafft.cbrc.jp/alignment/software/source.html>

⁶ <https://readseq-bioinformatics-data-conversion.soft112.com/>

⁷ <http://mcinerneylab.com/software/modelgenerator/>

⁸ <https://cme.h-its.org/exelixis/web/software/raxml/>

⁹ <https://github.com/niemasd/ViralMSA>

¹⁰ <http://www.microbesonline.org/fasttree/>

¹¹ <https://github.com/uym2/MinVar-Rooting>

¹² <https://github.com/tothuhien/lsd2>

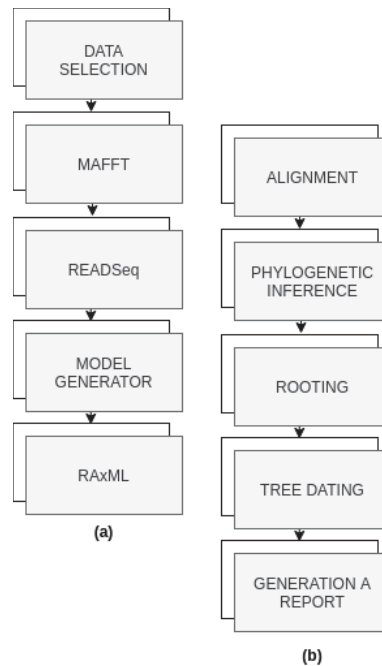
Figura 36 - Screenshot de cada instância de máquina virtual na nuvem, rondado o ambiente colaborativo.

The screenshot displays several terminal windows from different virtual machines, each showing the output of the command `docker ps -a`. The output lists various containers, their images, commands, creation times, and statuses. Key components visible include:

- peer nodes:** Containers like `peer0.UFRJ.com` and `peer0.UFMG.com` using `hyperledger/fabric-peer` images.
- orderer:** A container `orderer.example.com` using `hyperledger/fabric-orderer`.
- couchdb:** Containers `couchdb0` through `couchdb3` using `hyperledger/fabric-couchdb`.
- CA (Certificate Authority):** Containers like `ca.UFRJ.com` and `ca.UFMG.com` using `hyperledger/fabric-ca`.

Fonte: Elaborada pelo autor.

Figura 37 - *Workflow* SciPhy (a) e *Workflow* ViReport (b) executados no experimento.



Fonte: Elaborada pelo autor.

1. Alinhamento Múltiplo de Sequências (MSA): esta atividade constrói alinhamentos individuais utilizando um dos cinco programas disponíveis para alinhamento genético, o ClustalW, o Kalign, o MAFFT, o Muscle, ou o ProbCons. Cada programa de MSA, recebe como entrada um arquivo multi-fasta (que pode representar um aminoácido ou não) (a partir de um conjunto de arquivos multi-fasta), produzindo como saída um alinhamento (MSA).
2. Conversão de Alinhamento: esta atividade converte para o formato PHYLIP, o alinhamento (MSA) produzido pela atividade 1. Este formato é alcançado através do programa ReadSeq.
3. Eleição de Modelo Evolutivo: esta atividade encontra o melhor modelo evolutivo utilizando o programa ModelGenerator.
4. Construção de Árvores Filogenéticas: esta atividade gera árvores filogenéticas utilizando o programa RAxML com parâmetro de bootstrap configurável. A Figura 37 (a) apresenta uma visão alto nível das atividades do SciPhy.

O segundo *workflow* utilizado foi o ViReport (SONG; MOSHIRI, 2020), que é um *workflow* para realizar análises filogenéticas em sequências virais e gerar relatórios epidemiológicos moleculares abrangentes. Este *workflow* é composto por cinco atividades:

1. Alinhamento Múltiplo de Sequências (MSA) (*Alignment*): esta atividade refere-se ao alinhamento de sequências biológicas relacionadas de comprimento semelhante, cuja saída pode ser usada para inferir relações evolutivas mais específicas. Esta atividade pode ser realizada através de programas disponíveis para alinhamento genético: MAFFT, MUSCLE e ViralMSA.
2. Inferência filogenética (*Phylogenetic Inference*): esta atividade é a reconstrução de árvores evolucionárias agrupando elementos individuais com base na ancestralidade compartilhada. A estimativa de árvores filogenéticas pode ser realizada através de vários métodos, como Inferência Bayesiana e Máxima Verossimilhança. Esta atividade pode ser realizada através de programas disponíveis tais como FastTree 2, e RAxML.
3. Enraizamento de árvores filogenéticas (*Rooting*): Árvores filogenéticas inferidas usando modelos comumente usados de evolução de sequência não têm raiz, mas a raiz é importante tanto para a interpretação quanto para as aplicações posteriores. Esta atividade enraiza uma árvore, através do programa FastRoot.
4. Datação Filogenética (*Tree Dating*): Esta atividade constrói uma árvore de tempo com informações de data. Esta atividade é realizada através do programa LSD2.

5. Visualização da árvore: esta atividade permite a visualização da árvore inferida. Os programas iTOL e IcyTree podem ser utilizados. A Figura 37 (b) apresenta uma visão alto nível das atividades do ViReport.

4.5.3 Cenário

Nesta prova de conceito, o cenário de condução do experimento é o de pesquisadores e suas equipes geograficamente distribuídas, pertencentes a diferentes instituições de pesquisa. Para conduzir o experimento, os pesquisadores necessitavam de um ambiente colaborativo que oferecesse confiança aos dados processados e permitisse a consulta a proveniência destes dados, considerando que a reprodutibilidade é essencial nesse contexto.

Além disso, esses pesquisadores necessitavam de um ambiente que oferecesse suporte à interoperabilidade de dados oriundos de diferentes contextos, considerando que os pesquisadores executam partes do experimento utilizando diferentes SWfMSs. E por fim, necessitavam de um ambiente que oferecesse uma infraestrutura escalável, robusta e de alto desempenho para atender às necessidades de execução do experimento, que é intensivo em dados. Para execução dos *workflows* de análise filogenética, foi considerada ainda a necessidade de colaboração entre os cientistas, onde duas ou mais equipes geograficamente distribuídas pudessem trabalhar colaborativamente.

Nesta PoC, consideramos que a colaboração em um experimento científico é a execução metódica de *workflows* científicos com muitos conjuntos de dados executados colaborativamente, em diferentes momentos por um ou mais usuários, ou executados várias vezes por um ou mais cientistas, que combinam vários conjuntos de dados ou diferentes *workflows*. Além disso, é considerado que experimentos científicos passam por três fases: composição, execução e análise.

Na Composição, cientistas estruturam e configuram todo o experimento, estabelecendo a sequência lógica de atividades, o tipo de dados de entrada a serem fornecidos e o tipo de dados de saída. Na execução, os cientistas materializam o experimento, definem os dados de entrada necessários para executar o experimento, disparam sua execução (geralmente realizada por SWfMSs) e obtêm os resultados a serem analisados. Na análise, os cientistas estudam os dados coletados de fases anteriores (MATTOSO et al., 2010), com o objetivo de provar ou refutar suas hipóteses. Assim, consideramos que cada uma dessas fases do experimento pode envolver diferentes formas de colaboração.

Um experimento típico de filogenia pode analisar centenas ou milhares de arquivos multifasta, cada um contendo centenas ou milhares de sequências biológicas. Dependendo da quantidade de dados de entrada e da complexidade do método de atividade de alinhamento genético MSA, uma execução comum de um workflow deste tipo pode levar horas. Assim, apesar do SciPhy ser computacionalmente simples, considerando que apenas seis programas são orquestrados, na prática, ele é demasiadamente custoso.

Considerando a Blockflow, as atividades de alinhamento genético (MSA) podem ser executadas de maneira distribuída, onde cada pesquisador pode alinhar uma porção de dados que logo após é concatenada de forma a gerar um superalinhamento. A principal vantagem dessa abordagem é que o tempo de execução do *workflow* pode ser reduzido ou uma porção maior de sequências podem ser analisadas gerando assim uma árvore filogenética construída colaborativamente. Como existe uma grande história de recombinação de coronavírus, e pelo fato de trocarem uma boa parte do seu material genético, com somente uma árvore filogenética, pode não ser possível representar sua história evolutiva.

Assim, na execução da PoC, consideramos que cada equipe geograficamente distribuída pudesse adotar abordagens ligeiramente diferentes (genomas diferentes, por exemplo, para então gerar diferentes árvores), que podem ser potencialmente gerenciadas por diferentes *workflows* executados em diferentes SWfMSs. Assim, a equipe A, familiarizada com o SWfMS Taverna e a equipe B, que trabalha melhor com o SWfMS Kepler, puderam continuar utilizando os SWfMSs comumente utilizados nas suas pesquisas. No entanto, esse cenário levou a necessidade de interoperabilidade de dados, onde os pesquisadores necessitaram analisar os dados, notadamente os dados de proveniência, advindos dos SWfMS diferentes, de maneira integrada.

Embora os *workflows* SciPhy e ViReport difiram, ambos têm o mesmo objetivo, e seus resultados podem ser comparáveis. Além disso, a capacidade de realizar análises nos dados de proveniência combinados ajuda as equipes colaborativas a obter uma compreensão mais detalhada dos *workflows* relacionados. Portanto, nesta PoC, dada a importância de reprodutibilidade de resultados científicos e da interoperabilidade de dados de proveniência, os pesquisadores puderam consultar dados de proveniência integrados, gerados pelos *workflows* SciPhy e ViReport e consolidados pela Blockflow utilizando o ProvONE.

4.6 EXECUÇÃO

Com o objetivo de atender os requisitos de confiabilidade, privacidade, transparência, interoperabilidade e reprodutibilidade dos dados, a arquitetura *Blockchain* foi instanciada para ser utilizada nesta PoC. Assim, para instanciar o ambiente científico colaborativo, distribuído e confiável, uma rede *blockchain* para o ambiente de nuvem foi criada.

A Figura 38 apresenta a interface que permitiu que pesquisadores implementassem e configurassem suas redes *blockchain* através da BlockFlow. Conforme a Figura 38, foi especificado:

1. (Figura 38-A) - Nome do experimento como "ColaboracaoNaNuvemCovid19".
2. (Figura 38-B) - Uma descrição para o experimento como "Laboratório colaborativo para o novo coronavírus (SARS-CoV-2). Árvores filogenéticas."

Figura 38 - Interface do usuário, para que pesquisadores possam criar redes colaborativas de experimentação científica utilizando a arquitetura BlockFlow.

The screenshot shows the BlockFlow user interface for creating a collaborative network. It is divided into three main sections: Network, Organizations, and Channel.

- Network Section:**
 - Name Network Experiment:** ColaboracaoNaNuvemCovid19 (marked with A).
 - Description:** Laboratório colaborativo para o novo coronavirus (SARS-CoV-2). Árvores filogenéticas. (marked with B).
- Organizations Section:**
 - Four rows of organization configuration, each with a delete button and a count (marked with C):
 - UFJF, Number of Peer: 1, count: 1
 - UFF, Number of Peer: 1, count: 2
 - UFMG, Number of Peer: 1, count: 3
 - UFRJ, Number of Peer: 1, count: 4
- Channel Section:**
 - Channel Name:** sarscovChannel (marked with D).
 - Organizations in channel:** A list of checkboxes for UFJF, UFF, UFMG, and UFRJ, all of which are checked.

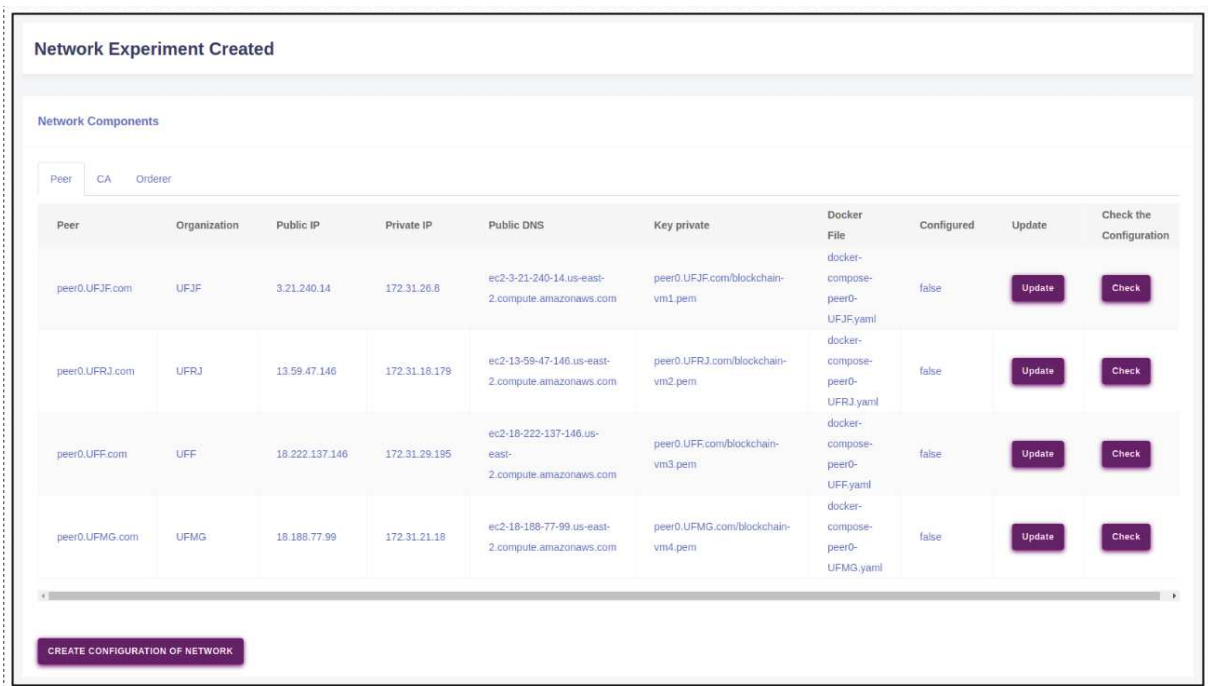
Fonte: Elaborada pelo autor.

3. (Figura 38.-C) - Quais organizações e quantidade de pares (pesquisadores) distribuídos iriam colaborar no experimento. Logo, foi definido: .
 - a) (Figura 38-C-1.) Organização UFJF com 1 nó *peer*.
 - b) (Figura 38-C-2.) Organização UFF com 1 nó *peer*.
 - c) (Figura 38-C-3.) Organização UFMG com 1 nó *peer*.
 - d) (Figura 38-C-4.) Organização UFRJ com 1 nó *peer*.
4. Para que houvesse o compartilhamento e transações de proveniência somente entre as partes interessadas, considerando a privacidade dos dados de proveniência compartilhados, um canal foi criado. Este canal foi denominado "sarscovChannel", que conforme a Figura 38-D, a "UFJF" e "UFMG", "UFF", "UFRJ" fazem parte.

Em seguida, as configurações das instâncias de máquinas virtuais na nuvem, como IP público, IP privado, nome de usuário, DNS público e chaves de acesso ao servidor da nuvem, foram especificados. A Figura 39 apresenta a *interface* que detalha as configurações.

Além disso, para que os pesquisadores pudessem colaborar, foi também realizada de forma automática, toda a configuração necessária para iniciar a rede na nuvem, como (i) iniciar *peers*, (ii) criar canais, (iii) criar identidades, (iv) instalar *chaincode*, (v) instanciar *chaincode*, foram especificadas. A Figura 40 apresenta a tela inicial com a instanciação do ambiente colaborativo, a partir da qual os pesquisadores puderam fazer/visualizar todas as configurações necessárias. A partir da instanciação inicial do ambiente, os dados que trafegaram pelo ambiente ficaram **seguros e interoperáveis** e com a captura dos dados de proveniência, conjuntamente com o ambiente *blockchain*, o suporte a **reprodutibilidade** do experimento foi possível, conforme discutiremos adiante.

Figura 39 - Interface do usuário, com todos os componentes da rede *blockchain* onde são especificadas as configurações, de cada PEERS, CAS, Orderes.



The screenshot shows a web interface titled "Network Experiment Created". It features a "Network Components" section with tabs for "Peer", "CA", and "Orderer". Below the tabs is a table listing four peers. Each row in the table contains the following information: Peer name, Organization, Public IP, Private IP, Public DNS, Key private, Docker File, Configured status, an Update button, and a Check the Configuration button. At the bottom of the interface, there is a button labeled "CREATE CONFIGURATION OF NETWORK".

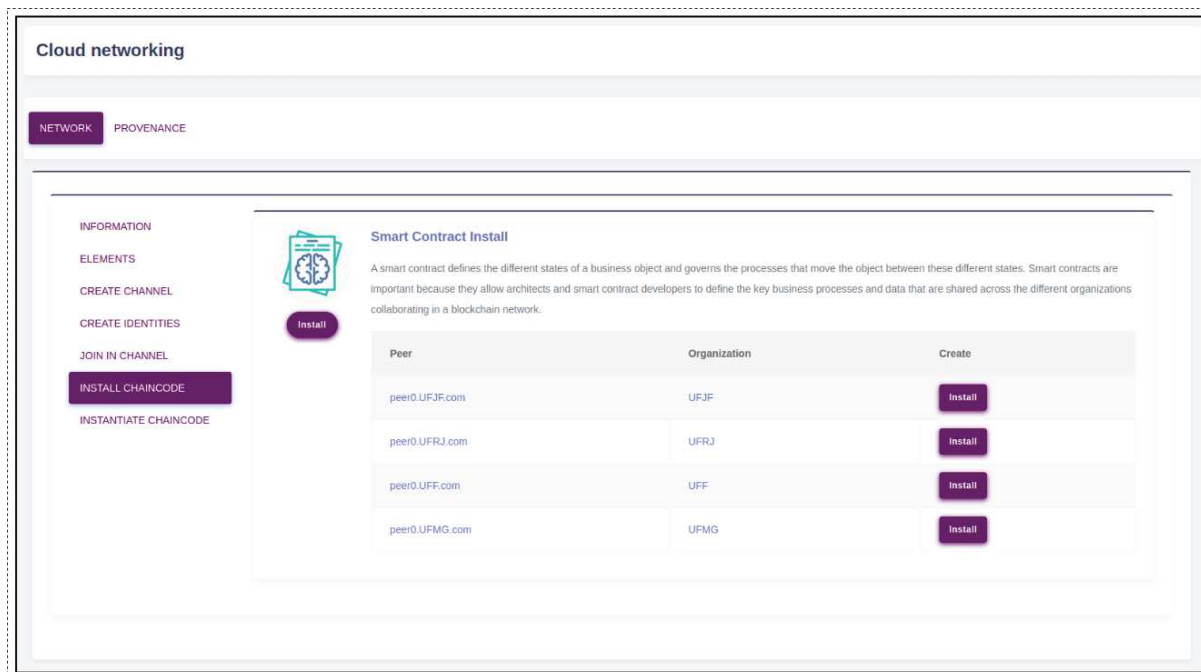
Peer	Organization	Public IP	Private IP	Public DNS	Key private	Docker File	Configured	Update	Check the Configuration
peer0.UFJF.com	UFJF	3.21.240.14	172.31.26.8	ec2-3-21-240-14.us-east-2.compute.amazonaws.com	peer0.UFJF.com/blockchain-vm1.pem	docker-compose-peer0-UFJF.yaml	false	Update	Check
peer0.UFRJ.com	UFRJ	13.59.47.146	172.31.18.179	ec2-13-59-47-146.us-east-2.compute.amazonaws.com	peer0.UFRJ.com/blockchain-vm2.pem	docker-compose-peer0-UFRJ.yaml	false	Update	Check
peer0.UFF.com	UFF	18.222.137.146	172.31.29.195	ec2-18-222-137-146.us-east-2.compute.amazonaws.com	peer0.UFF.com/blockchain-vm3.pem	docker-compose-peer0-UFF.yaml	false	Update	Check
peer0.UFMG.com	UFMG	18.188.77.99	172.31.21.18	ec2-18-188-77-99.us-east-2.compute.amazonaws.com	peer0.UFMG.com/blockchain-vm4.pem	docker-compose-peer0-UFMG.yaml	false	Update	Check

Fonte: Elaborada pelo autor.

4.6.1 Coleta e Armazenamento de dados de proveniência

Para capturar os dados de proveniência e interoperar esses dados entre os pesquisadores que colaboram no experimento, a BlockFlow fornece um serviço *web* através da camada *API RESTful Web Service*. A captura de proveniência é feita em tempo real e é independente do SWfMS. Para capturar os dados, é necessário instrumentalizar com um serviço *web*, cada atividade (tarefa) do *workflow*. A Figura 41 apresenta o *workflow* SciPhy com suas diferentes tarefas instrumentalizadas no SWfMS Taverna e a Figura 42

Figura 40 - Interface do usuário, para que os pesquisadores possam (i) iniciar *peers*, (ii) criar canais, (iii) criar identidades, (iv) instalar *chaincode*, (v) instanciar *chaincode*.



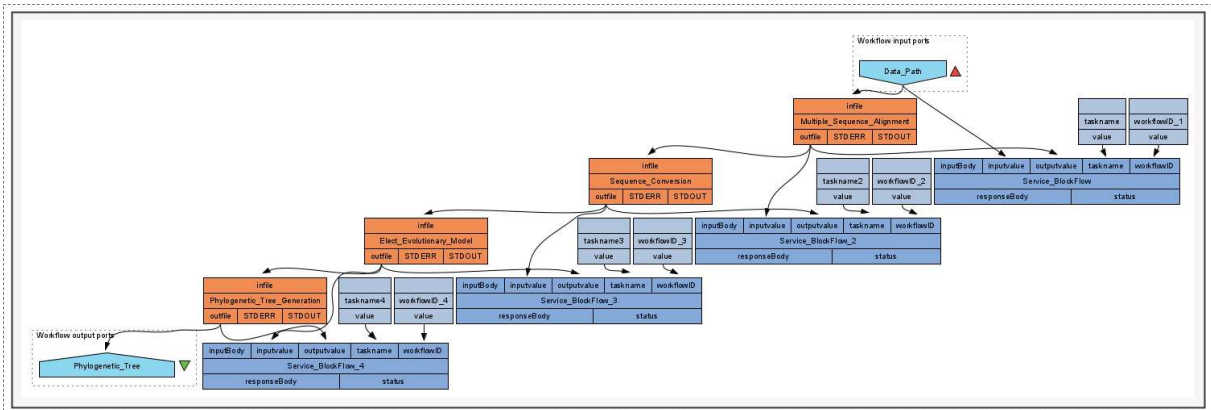
Fonte: Elaborada pelo autor.

apresenta o *workflow* ViReport instrumentalizado no SWfMS Kepler. Além disso, para uma completa tradução de proveniência para o modelo ProvONE, a proveniência em tempo de execução (proveniência retrospectiva) deve ser vinculada à proveniência prospectiva. Assim, foram seguidos os passos detalhados a seguir para se instrumentalizar os *workflows*. É importante ressaltar que esta tarefa não é trivial e os pesquisadores precisaram de suporte de pessoal especializado para alguns passos dessa instrumentalização.

Foram cadastrados quais *workflow(s)* fariam parte do experimento e que teriam sua proveniência coletada. Assim, através da *interface web* da Blockflow, conforme a Figura 43, foram criados cada um dos *workflows*. Conforme a Figura 43, foi especificado:

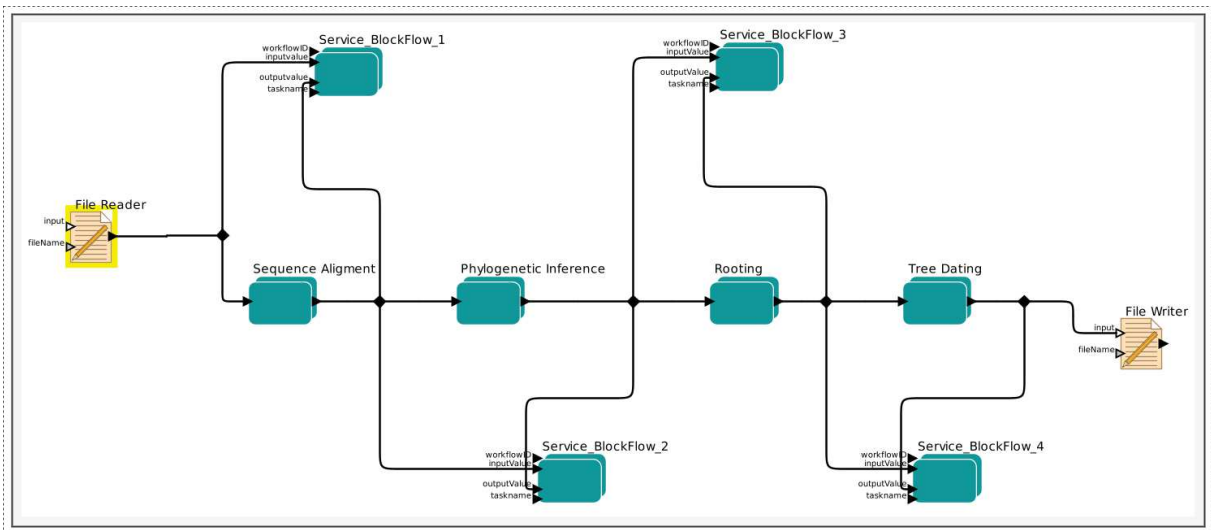
1. (Figura 43-A) - Nome do *Workflow*.
2. (Figura 43-B) - Descrição do *Workflow*.
3. (Figura 43-C) - Suas Tarefas.
4. (Figura 43-D) - *wasDerivedFrom* (informação para especificar se um *workflow/* é derivado de um outro *workflow*).

Figura 41 - *Workflow* Sciphy instrumentalizado com serviço da *web* da BlockFlow.



Fonte: Elaborada pelo autor.

Figura 42 - *Workflow* Sciphy instrumentalizado com serviço da *web* da BlockFlow.



Fonte: Elaborada pelo autor.

Em seguida, os dados foram salvos na rede *blockchain*. A Figura 44 apresenta os dados cadastrados dos *workflows* executados.¹³

¹³ Esses dados são importantes, pois durante a coleta da proveniência retrospectiva, o dado *hash* da transação, ou o id do *workflow* deve ser enviado ao serviço *web*, assim como *token* do usuário recebido durante a autenticação. Essa é uma forma de relacionar o *workflow* e ao pesquisador (*user*) com a sua respectiva execução (proveniência retrospectiva). Além disso, o *token* de pesquisador (*user*) é uma maneira de confirmar a autenticidade e recuperar a identidade do usuário como um nó pertencente à rede *blockchain*, além de garantir que os

Figura 43 - Tela de cadastros dos *Workflows* (a) Sciphy (b) ViReport.

The image shows two side-by-side screenshots of a 'Workflow' registration form. The left form (a) is for 'SciPhy_Workflow' and the right form (b) is for 'Vireport_Workflow'. Both forms have four red circles labeled A, B, C, and D pointing to the Name Workflow, Description, Activities/Tasks, and wasDerivedFrom fields respectively. Each form has an 'Add' button at the bottom right.

Fonte: Elaborada pelo autor.

Além disso, é requerido, porém não obrigatório, armazenar os dados que serão utilizados, como entrada ou gerados como saída durante a execução de uma tarefa ou de um *workflow*. A Figura 45 apresenta um *upload* dos dados utilizados, conforme a Figura 45 foi especificado:

1. (Figura 45-A) - Nome do *Workflow* em que os dados foram utilizados ou gerados.
2. (Figura 45-B) - Nome da Tarefa que os dados foram utilizados ou gerados.
3. (Figura 45-C) - Os dados são de entrada ou de saída.
4. (Figura 45-D) - Uma descrição para os dados.
5. (Figura 45-E) - Campo de *upload* dos dados.

Nessa prova de conceito, como arquivos de entrada para os *workflows* executados, foram utilizadas 25 sequências completas do genoma de diferentes cepas de coronavírus (incluindo SARS, MERS e SARS-CoV-2) e 61 cepas de coronavírus de diferentes países e

dados sejam posteriormente assinados na rede. Se a identidade do usuário não puder ser verificada, as informações serão rejeitadas e não consideradas, e os dados de proveniência não serão registrados.

Figura 44 - Tela com cadastros dos *Workflows* SciPhy, ViReport.

Hash Blockchain ID	Workflow	Description	Activities/Tasks	WasDerivedFrom	View All Provenance
28b31185-db3a-4596-b5e9-562fd1aaf7ea	SciPhy_Workflow	Scientific workflow of phylogenetic analysis. SARS-CoV-2	1) MSA Construction, 2) MSA Format Conversion, 3) Evolutionary Model, 4) Phylogenetic Tree Construction	-	View
9-345221453033289b5763762199303328	Vireport_Workflow	Scientific workflow for phylogenetic analysis in viral sequences. SARS-CoV-2	1) Sequence Alignment, 2) Phylogenetic Inference, 3) Rooting, 4) Tree Dating	-	View
b-21351116849639512627f24w963350	SciPhy_Workflow	Scientific workflow of phylogenetic analysis. SARS-CoV-2	1) MSA Construction, 2) MSA Format Conversion, 3) Evolutionary Model, 4) Phylogenetic Tree Construction	28b31185-db3a-4596-b5e9-562fd1aaf7ea	View

Fonte: Elaborada pelo autor.

regiões. Todas as sequências, foram obtidas no GISAID e NCBI GenBank. Após o *upload*, um *hash* para cada arquivo foi gerado. A Figura 46 apresenta os dados armazenados.

Essa opção é interessante considerando a **reprodutibilidade** do *workflow*, pois ela permite analisar se o objeto de pesquisa (dados) usados ou gerados em um experimento possui o conteúdo equivalente ao que foi publicado e compartilhado no experimento. Desta forma esses dados podem ser comparados com os dados salvos nas classes (*Entity*) ProvONE armazenados no *blockchain* durante a execução do *workflow* que é imutável¹⁴.

Após esses passos, o SciPhy e ViReport foram executados nas diferentes máquinas virtuais, onde foi coletada a proveniência. A camada *Wrapper* da arquitetura BlockFlow transformou cada dado para o formato ProvONE conforme detalhado na seção 3.2.2, e em seguida, os enviou como transações para rede *blockchain*. Essa operação garantiu a imutabilidade (integridade) e transparência nas informações de proveniência.

¹⁴ Na seção a seguir durante a análise de proveniência, será feita uma análise em relação à proveniência coletada e comparação sobre o *hash* coletado

Figura 45 - Tela *FrontEnd* de *upload* de arquivos de entrada e de saída dos *workflows*.

The image shows a web form titled "File Inputs and Output Workflow" with a close button (X) in the top right corner. The form contains five main input sections, each highlighted with a red box and a red lettered circle (A-E) on the left side:

- A:** Workflow: A text input field containing "SciPhy_Workflow".
- B:** Activity/Task: A text input field containing "MSA".
- C:** Type: A text input field containing "input".
- D:** Description: A text input field containing "EPI_ISL_402131".
- E:** Choice File: A text input field with a "Browse" button next to it.

At the bottom right of the form is a purple "Add" button.

Fonte: Elaborada pelo autor.

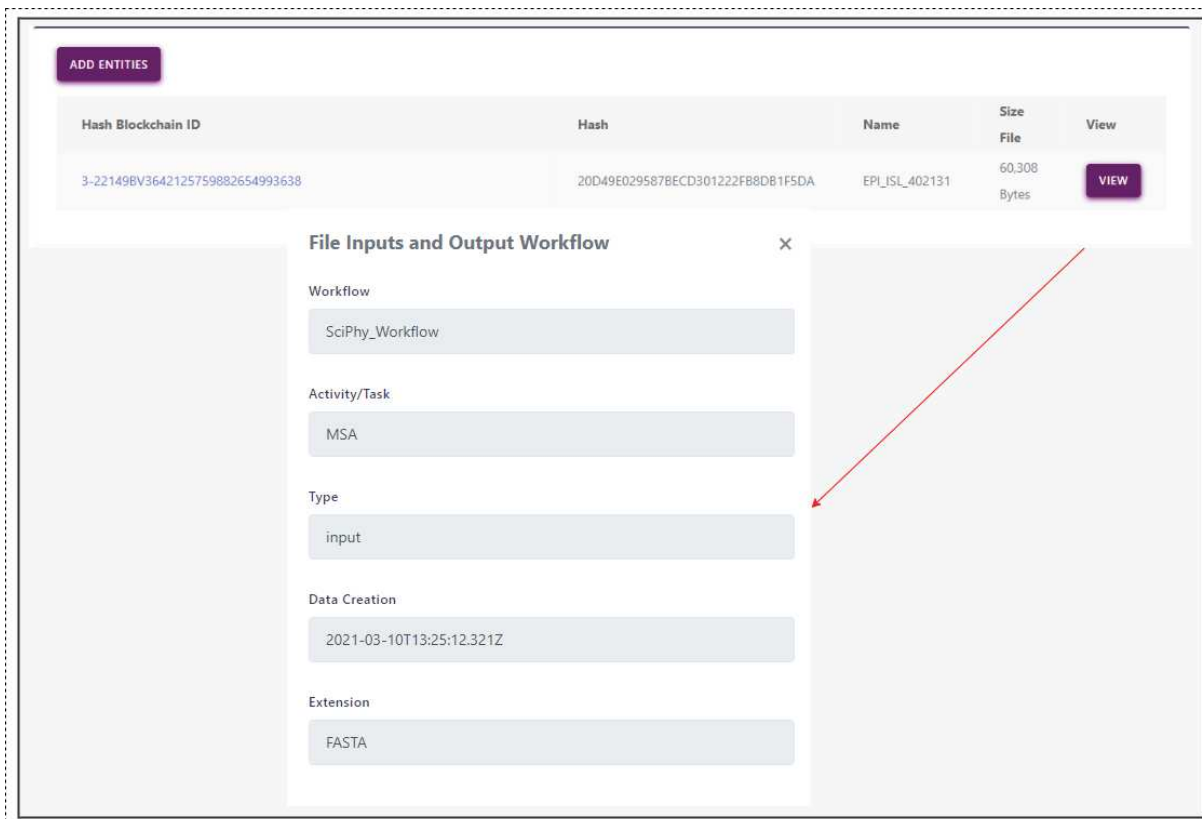
As árvores geradas podem ser visualizadas conforme as Figuras 47 e 48. A proveniência coletada durante a execução dos *workflows*, pode ser visualizada conforme a Figura 49. Essa proveniência compartilhada, devido à natureza distribuída e imutável do *blockchain* é **transparente**. Todos os nós (cientistas), conectados a rede que compõem o experimento, podem verificar e visualizar como a proveniência foi criada na cadeia (*blockchain*) ao longo do tempo. Assim, conforme detalhado na próxima seção, todas as atualizações de dados podem ser rastreadas entre nós. Além disso, os dados da pesquisa podem ser analisados e revisados pelos pares de uma maneira comprovada.

4.6.2 Análise e Consultas de dados de Proveniência

Para obter uma visão geral dos dados de proveniência coletados durante a execução do experimento, os pesquisadores puderam realizar consultas a partir da *interface web* da BlockFlow, conforme a Figura 50. Estas consultas podem ser com componentes fixos, conforme a Figura 50 - A, ou através de consultas no formato do banco de dados *CouchDB*, conforme a Figura 50 - B.

Na BlockFlow, a visualização dos dados retornados através das consultas executadas podem ser feita através de uma tabela, conforme a Figura 49, ou através do formato JSON, este último permitindo a interoperabilidade com a plataforma E-SECO ou qualquer outra

Figura 46 - Tela FrontEnd com arquivos armazenados.



Fonte: Elaborada pelo autor.

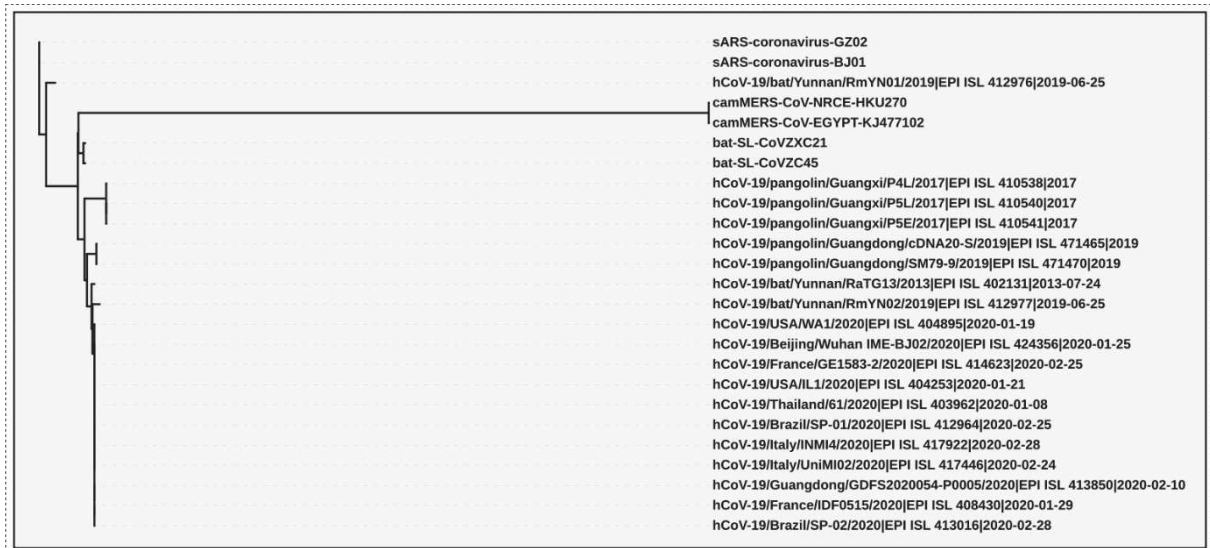
aplicação que necessite consumir dados da Blockflow.

A seguir apresentamos diferentes consultas executadas pelos pesquisadores no conjunto de dados de proveniência coletado durante a execução desta PoC, considerando os *workflows* SciPhy e ViReport.

Q1) *Recuperar todas as tarefas com suas portas de entrada e saída para o workflow SciPhy.* A Figura 51 A apresenta a consulta especificada a partir dos componentes fixos. Nesta foram escolhidos a classe (*Program*) do modelo ProvONE, o parâmetro (*workFlowID*) para especificar o (id) do *workflow* específico, ao qual, os pesquisadores queriam recuperar os dados, e logo abaixo especificaram os campos que gostariam que fossem retornados, (*idProgram*, *NameProgram*, *hasOutPort*, *hasInPort*). A Figura 51 B representa a mesma consulta, porém no formato do banco de dados *CouchDB*. A Figura 52 apresenta o resultado em formato JSON (estes dados também podem ser visualizados através de uma tabela, conforme a Figura 48).

Q2) *Recupere todas as execuções de atividades com seus dados gerados para o grafo de proveniência do workflow ViReport.* A Figura 53 apresenta a consulta utilizando os

Figura 47 - Árvore filogenética com base nas sequências de 25 genomas, completo de coronavírus, incluindo SARS-CoV-2, SARS-CoV, HCoV, morcego SARS, SARS-like CoV e MERS-CoV.



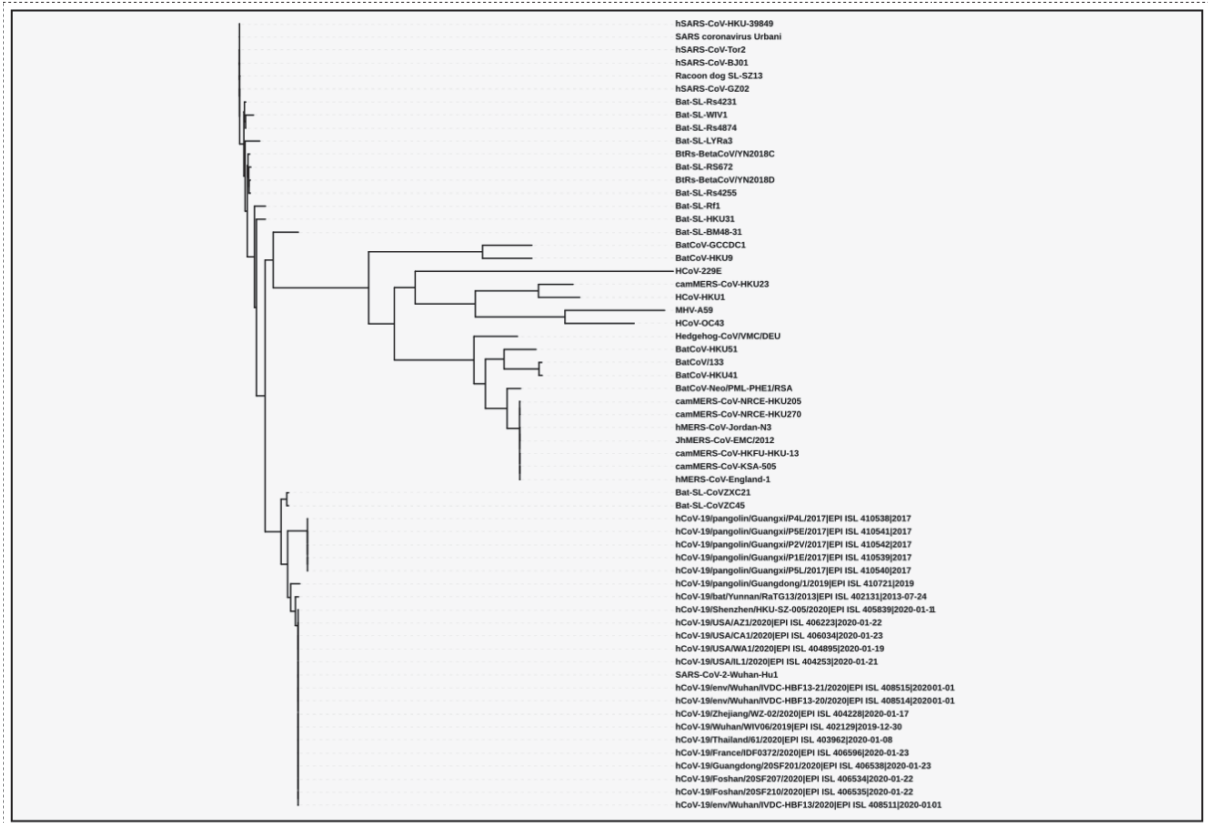
Fonte: Elaborada pelo autor.

componentes fixos. Nesta foi escolhido o relacionamento (*wasGenerationBy*) do modelo ProvONE, o parâmetro (*workFlowID*) para especificar o (*id*) do *workflow* específico, ao qual os pesquisadores queriam acessar os dados e logo abaixo os campos que gostariam que fossem retornados, (*idWasGeneratedBy*, *programExecution*, *programExecutionName*, *idEntity*, *entityValue*). A Figura 53 B representa a mesma consulta, porém no formato do banco de dados *CouchDB*. A Figura 54 apresenta o resultado em formato JSON.

Q3) *Recupere todas as execuções de atividades de ambos workflows, Sciphy e Vireport.* A Figura 55 A detalha a especificação da consulta a partir dos componentes fixos providos pela BlockFlow. Nesta foi escolhido a classe (*Execution*) do modelo ProvONE, o parâmetro (*workFlowID*) para especificar os (*id's*) dos *workflow* específicos, que, os pesquisadores precisavam buscar os dados e logo abaixo especificados os dados a serem retornados, (*programExecutionID*, *programExecutionName*, *startTime*). A Figura 55 B representa a mesma consulta, porém no formato do banco de dados *CouchDB*. A Figura 56 apresenta o resultado em formato JSON.

A seguir, na Tabela 14 apresentamos diferentes consultas também realizadas pelos pesquisadores na condução desta PoC. Os pesquisadores, após o processamento das consultas, puderam fazer *download* dos resultados em formato JSON conforme a Figura 57. Esses dados em formato JSON foram carregados no banco de dados orientado à grafos Neo4j. A Figura 58 apresenta uma pesquisa em *Cypher*, onde foi possível associar e

Figura 48 - Árvore filogenética com base nas seqüências de 61 genomas, completo de coronavírus, incluindo SARS-CoV-2, SARS-CoV, HCoV, morcego SARS, SARS-like CoV e MERS-CoV.



Fonte: Elaborada pelo autor.

retornar um pesquisador com suas respectivas execuções.

Tabela 14 – Consultas.

Especificação de consultas	Consultas
Recupere todas as execuções associadas a um usuário específico.	<code>{"selector": {"docType": {"\$eq": "wa-sAssociatedwith"}, "agent": {"\$eq": "Rai-ane"}}, "fields": ["agent", "idAgent", "programExecutionName", "idProgramExecution_id", "_rev"] }</code>
Recupere toda a proveniência de um <i>Workflow</i> para cada um de suas execuções.	<code>{"selector": {"workflowID": {"\$and": [{"\$eq": "18af6229-94c5-4175-8679-cfa6254fd01"}, {"\$eq": "b-21351116849639512627f24w963350"}] }},</code>

Recupere todas as execuções de atividades com seus dados gerados para cada execução de um <i>workflow</i> .	{"selector":{"docType":{"\$eq":"entity"}}, "workflowID":{"\$eq":"9-345221453033289b5763762199303328"}}, "fields":["idEntity","startTime","typeEntity","valueEntity","_id","_rev"]}
---	--

4.6.3 Discussões

Segundo (MIYAKAWA, 2020), uma das possíveis causas da crise de reprodutibilidade, está relacionada à falta de dados brutos utilizados em pesquisas. Ainda, segundo (BIK; CASADEVALL; FANG, 2016) muitos pesquisadores manipulam sistematicamente dados utilizados em suas pesquisas, para obter resultados satisfatórios. Argumentamos ser importante disponibilizar os dados processados durante a execução dos *workflows*, como é possível ser feito a partir da arquitetura Blockflow.

Na condução desta PoC, utilizando a BlockFlow, durante a coleta de proveniência, foram registradas todas as invocações de tarefas, e os dados processados por estas tarefas foram coletados e organizados segundo o modelo ProvONE através da classe *Entity*. Nas classes *Entity(s)* foram armazenados os *hashs* desses dados, assim como o seu caminho, data e hora de execução.

Assim, fica demonstrado que, na BlockFlow, é possível comparar os dados processados durante a execução de um *workflow* executado por um pesquisador como uma forma de garantir **confiança e reprodutibilidade**. Para isso, os pesquisadores puderam realizar consultas na proveniência coletada e comparar com os dados armazenados e disponibilizados por outros pesquisadores.

Foi também realizado o *upload* de todos os dados usados e gerados durante a execução do *workflow* Sciphy e, em seguida, recuperadas todas as *Entity(s)* (entidades) do *workflow* executado e foi comparado se os dados de *hash* eram os mesmos, considerando os dados disponibilizados na arquitetura. Para recuperar todas as *Entity(s)* geradas durante a execução de um *workflow*, foi executada a consulta, conforme a Figura 59 no formato do banco de dados *CouchDB*.

Os resultados, no formato JSON podem ser visualizados na Figura 60, comparando-se cada resultado. Através dessa comparação, foi possível relacionar cada dado com a proveniência coletada e observar se o objeto de pesquisa (dados) usados ou gerados em um experimento possuía o conteúdo equivalente ao que foi publicado e compartilhado no experimento.

Assim, consideramos que, através desta prova de conceito, foi possível verificar

Figura 49 - Interface de usuário com a proveniência coletada durante a execução do experimento.

ID	Key	View
10891257-g6bgc-1130-cvbb-312cs4563217	1-f56b43396a22097210d757e12a19ca9e	Detail
1765uuio-868cn-c2v7-ebt9-81534936nmvh	1-564e3t39622d0972r1075g7se12y199e	Detail
11991397-gh608-dr46-6cv2-12611624qasy	9-75398klm451932g7091509n4y331e561	Detail
127787u5-364cv-1482-753w-1637m4hw9sti	3-7698010s351213n37244509b2883b891	Detail
331982cv-1309p-1767-dc4d-443e667209c5	5-640180510c5739682452p758o8527584	Detail
1807w190-56719-63mw-wsef-b7g8fvcdwsp0	0-91254a1770431440a471884316937915	Detail
16774736-d2119-Oplo-qwsa-nvc1653938cx8	8-952176c5266341592387583791591635	Detail
1963pl12-po52w-f527-0987-uplxolc00120e7	7-7948235897657319026191351490254f	Detail
1987cvwe-1400i-ulp1-44rd-129cop1672cse6	2-731276780146815973165410b835153r	Detail
0rfg000d-pldgb-133f-273c-16628731151d97	4-371491199497722976086912f369416d	Detail

Fonte: Elaborada pelo autor.

a viabilidade da arquitetura BlockFlow em permitir que pesquisadores geograficamente distribuídos possam colaborar em experimentos científicos em um ambiente confiável e transparente, compartilhando seus dados de forma integrada. Desta forma, considerando o cenário apresentado e os elementos que o compõem, podemos responder às questões previamente delineadas.

(QS1) – *A BlockFlow pode fornecer uma visão geral dos dados de proveniência de forma transparente, onde pesquisadores geograficamente distribuídos, podem verificar como os dados de proveniência foram criados na cadeia (blockchain) ao longo do tempo?*

Como podemos observar durante a execução da PoC, conforme detalhado na seção 4.6.1, devido à natureza distribuída e imutável do *blockchain*, os dados de proveniência compartilhados entre os pesquisadores em um experimento são transparentes, ou seja, são compartilhados entre todos e em tempo real. Conforme detalhado na seção 4.6.2, todas as atualizações de dados de proveniência podem ser rastreadas e visualizadas, através das consultas (*queries*), pelos nós (pesquisadores) geograficamente distribuídos. Assim, podemos verificar que a BlockFlow pode fornecer uma visão geral dos dados de proveniência

Figura 50 - Interface de usuário para executar query(s).

The image shows a user interface for executing queries, divided into two main sections: 'QUERY COMPONENT' (labeled A) and 'QUERY COUCHDB' (labeled B).

QUERY COMPONENT (A):

- Choice Type Classes ProvONE:** A dropdown menu with 'Workflow' selected.
- Parameter ProvONE:** A dropdown menu with 'workflowID' selected.
- Operation:** A dropdown menu with 'equals' selected.
- Value:** A text input field containing '18af6229-94c5-4175-8679-cfa6254fd01f'.
- Add new value in query:** A purple button.
- Choice fields for view in query:** A list of checkboxes:
 - IdProgram
 - NameProgram
 - HasOutPort
 - HasInPort

QUERY COUCHDB (B):

```

1 {
2   "selector": {
3     "docType": {
4       "$eq": "program"
5     }
6   },
7   "workflowID": {
8     "$eq": "18af6229-94c5-4175-8679-cfa6254fd01f"
9   }
10 }

```

Fonte: Elaborada pelo autor.

de forma transparente, onde pesquisadores geograficamente distribuídos, podem verificar como os dados de proveniência foram criados na cadeia (*blockchain*) ao longo do tempo.

(QS2) – *A BlockFlow pode ser usada como um ambiente científico colaborativo e confiável apoiando a interoperabilidade de dados de proveniência advindos de SWfMSs heterogêneos?*

Durante a execução desta PoC conforme a seção 4.6.1, pudemos observar que os *workflows* escolhidos foram executados em diferentes SWfMSs. A Figura 41 apresenta o *workflow* SciPhy com suas diferentes tarefas, instrumentalizado no SWfMS Taverna e a Figura 42 apresenta o *workflow* ViReport instrumentalizado no SWfMS Kepler. Assim, devemos ressaltar que embora existam facilidades oferecidas pelo uso desses SWfMSs no gerenciamento de um experimento *in silico*, esses SWfMSs capturam dados de proveniência em modelos não totalmente interoperáveis. Assim, com intuito de capturar e interoperar os dados de proveniências desses *workflows* entre os pares distribuídos, conforme mencionado na seção 4.6.1, a BlockFlow utilizou um componente de serviço da *web*. A captura foi feita em tempo real e independente do SWfMS. Após a captura dos dados, os mesmos foram convertidos para o modelo ProvONE e estes foram armazenados no *blockchain*.

Figura 51 - Interface de usuário para executar a consulta (Q1).

QUERY COMPONENT

Choice Type Classes ProvONE
 Program

Parameter ProvONE: workflowID Operation: equals Value: 28b31185-db3a-4596-b5e9-562fd1aaf7ea

Add new value in query

Choice fields for view in query:

IdProgram NameProgram HasOutPort HasInPort

QUERY COUCHDB

```

1 {
2   "selector": {
3     "docType": {
4       "$eq": "program"
5     },
6     "workflowID": {
7       "$eq": "28b31185-db3a-4596-b5e9-562fd1aaf7ea"
8     }
9   },
10  "fields": [
11    "idProgram",
12    "nameProgram",
13    "hasInPort",
14    "hasOutPort",
15    "_id",
16    "_rev"
17  ]
18 }

```

Fonte: Elaborada pelo autor.

Conforme a seção 4.6.2, pudemos realizar de maneira integrada a consulta de proveniência dos diferentes *workflows* executados. Assim, pudemos verificar que a BlockFlow pode ser usada de uma maneira confiável apoiando a interoperabilidade de dados de proveniência advindos de SWfMSs heterogêneos.

QS3) – *A BlockFlow pode ser usada como um ambiente confiável de troca de proveniência em workflows intensivos em dados?*

Os *workflows* científicos escolhidos para execução da PoC são *workflows* filogenéticos que são *workflows* intensivos em dados. Esses geralmente precisam ser executados em ambientes colaborativos ou de alto desempenho, como ambientes de computação em nuvem. Para tanto, conforme descrito na seção 4.5, especificamos um ambiente colaborativo utilizamos ambientes de computação em nuvem, provisionando instâncias de máquinas

Figura 52 - Resultado em formato JSON da consulta (Q1).

```

1 {
2   "Workflow":{
3     "_id": "28b31185-db3a-4596-b5e9-562fd1aaf7ea",
4     "_rev": "1-ab7fa45e2e8a14e2e0b5c00ff2bbc0f2",
5     "created": "2021-03-17T17:40:28.174Z",
6     "docType": "workflow",
7     "workFlowName": "SciPhy_Workflow",
8     "description": "Scientific workflow of phylogenetic analysis, SARS-CoV-2",
9     "activities/tasks": "1) MSA Construction, 2) MSA Format Conversion, 3) Evolutionary Model, 4) Phylogenetic Tree Construction",
10    "wasDerivedFrom": null,
11    "program": [
12      {
13        "_id": "c7956d3c-9c66-49c0-b99b-a98bb67bed7a",
14        "_rev": "1-d0d953fd1d2d2c95f393af7bbe1cd77a",
15        "created": "2021-03-17T17:40:28.174Z",
16        "docType": "program",
17        "idProgram": "c7956d3c-9c66-49c0-b99b-a98bb67bed7a",
18        "nameProgram": "msa_construction",
19        "hasInPort": {
20          "docType": "hasInPort",
21          "hasInPortId": "0df56266-855f-4ba5-b521-8e93808b132e",
22          "port": {
23            "portId": "0f698a9d-e9e6-42a9-8a80-916d58454dba",
24            "programID": "c7956d3c-9c66-49c0-b99b-a98bb67bed7a",
25            "programName": "msa_construction",
26            "inputPortHashValue": "20D49E029587BEC301222FB8D81F5DA"
27          }
28        },
29        "hasOutPort": {
30          "docType": "hasOutPort",
31          "hasInPortId": "0df56266-855f-4ba5-b521-8e93808b132e",
32          "port": {
33            "portId": "0df56266-855f-4ba5-b521-8e93808b132e",
34            "programID": "c7956d3c-9c66-49c0-b99b-a98bb67bed7a",
35            "programName": "msa_construction",
36            "outPortHashValue": "4A599A1A2A9D51F831286543A458DAD7"
37          }
38        }
39      }
40    ]
41  }
42 }

```

Fonte: Elaborada pelo autor.

virtuais como *Amazon Elastic Compute Cloud* (Amazon EC2). Este ambiente oferece uma variedade de recursos como *hardware* e *software*, sob elasticidade sem a necessidade dos cientistas adquirirem infraestruturas computacionais. Além disso, para garantir confiança, esse ambiente foi desenvolvido baseado em *blockchain*, onde nenhum dado pode ser alterado e esses dados são transparentes. Devemos ressaltar que toda a configuração na BlockFlow, foi realizada de forma automática, conforme descrito na seção 4.6. Assim, podemos inferir que a BlockFlow pode ser usada como um ambiente confiável de troca de proveniência em

Figura 53 - Interface de usuário para executar a consulta (Q2).

QUERY COMPONENT

Choice Type Classes ProvONE
WasGenerationBy

Parameter ProvONE: workflowID Operation: equals Value: 9-345221453033289b5763762199303328

Add new value in query

Choice fields for view in query:

IdWasGeneratedBy programExecutionId programExecutionName idEntity entityValue

QUERY COUCHDB

```

1 {
2   "selector": {
3     "docType": {
4       "$eq": "wasGeneratedBy"
5     },
6     "workflowID": {
7       "$eq": "9-345221453033289b5763762199303328"
8     }
9   },
10  "fields": [
11    "IdWasGeneratedBy",
12    "programExecutionId",
13    "programExecutionName",
14    "programExecutionName",
15    "idEntity",
16    "entityValue",
17    "_id",
18    "_rev"
19  ]
20 }

```

Fonte: Elaborada pelo autor.

workflows intensivos em dados.

QS4) – *A BlockFlow pode ser usada como um ambiente que fornece privacidade aos dados de proveniência, considerando a propriedade intelectual, onde os dados são compartilhados apenas entre partes ou pessoas autorizadas?*

Com objetivo de responder a esta questão de pesquisa, mencionaremos a diferença entre um *blockchain* sem permissão e com permissão. Em um *blockchain* sem permissão, qualquer nó pode verificar qualquer transação que ocorreu na cadeia. Assim, quando é necessário privacidade ou confidencialidade nas transações, ou seja, entre os dados compartilhados, outros meios criptográficos são necessários. Entretanto, em um *blockchain* com permissão um participante precisa de permissão para fazer ou verificar transações,

Figura 54 - Resultado em formato JSON da consulta (Q2).

```

1 {
2   "Workflow":{
3     "_id": "9-345221453033289b5763762199303328",
4     "_rev": "1-ab7fa45e2e8a14e2e0b5c00ff2bbc0f2",
5     "created": "2021-03-17T17:40:28.174Z",
6     "docType": "workflow",
7     "workFlowName": "Vireport_Workflow",
8     "description": "Scientific workflow for phylogenetic analysis in viral sequences, SARS-CoV-2",
9     "activities/tasks": "1) Sequence Alignment, 2) Phylogenetic Inference, 3) Rooting 4) Tree Dating",
10    "wasDerivedFrom": null,
11    "programExecution": [
12      {
13        "_id": "p1150d9c-8c66-49c0-b74q-a22bb13bed8a",
14        "_rev": "1-d0d473h51j9d3z76f643az7XXe1cd13g",
15        "docType": "programExecution",
16        "startTime": "2021-03-17T16:40:28.174Z",
17        "nameProgramExecution": "sequence_alignment_exe",
18        "wasGeneratedBy": {
19          "docType": "wasGeneratedBy",
20          "hasInPortId": "0zf13332-160r-1ba9-v100-1e14256k142f",
21          "entity": {
22            "docType": "entity",
23            "idEntity": "dbd5ef24df25c8070d8dfdee714d4481",
24            "typeEntity": "data",
25            "valueEntity": "0D12C7EE86EBFB7E915FF635801BF18F"
26          }
27        }
28      },
29      {
30        "_id": "c2232d5p-9c47-79m1-c12c-a39tt22qmk7a",
31        "_rev": "1-d0d953fd1d2d2c95f393af7bbe1cd77a",
32        "docType": "programExecution",
33        "startTime": "2021-03-17T17:10:22.364Z",
34        "nameProgramExecution": "phylogenetic_inference_exe",
35        "wasGeneratedBy": {
36          "docType": "wasGeneratedBy",
37          "hasInPortId": "0bn48304-855f-4ba5-b521-8e56330n132f",
38          "entity": {

```

Fonte: Elaborada pelo autor.

essas transações ocorrem em um ecossistema fechado, os dados da transação permanecem confidenciais e os participantes são conhecidos e autenticados. Assim, devido a questões

Figura 55 - Interface de usuário para executar a consulta (Q3).

A

QUERY COMPONENT

Choice Type Classes ProvONE
 Program Execution

Parameter ProvONE: workflowID
 Operation: equals
 Value: 28b31185-db3a-4596-b5e9-562fd1aaf7ea

Operation: and

Parameter ProvONE: workflowID
 Operation: equals
 Value: 9-345221453033289b5763762199303328

Add new value in query

Choice fields for view in query:

programExecutionId programExecutionName startTime

B

```

1 {
2   "selector":{
3     "docType":{
4       "$eq": "programaExcecution"
5     },
6     "workflowID":{
7       "$and": [
8         {
9           "$eq": "28b31185-db3a-4596-b5e9-562fd1aaf7ea"
10        },
11       {
12         "$eq": "9-345221453033289b5763762199303328"
13       }
14     ]
15   },
16 },
17 "fields":{
18   "programExecutionId",
19   "programExecutionName",
20   "programExecutionName",
21   "_id",
22   "_rev"
23 }
24 }
  
```

Fonte: Elaborada pelo autor.

de privacidade de dados e propriedade intelectual levantados nessa dissertação, *blockchains* com permissão tornaram-se uma opção mais adequada em nosso contexto de colaboração de *workflows* científicos para o compartilhamento de dados de proveniência. Conforme já detalhado, na seção 2.2.21 escolhemos para construção da arquitetura, a *blockchain* com permissão *Hyperledger Fabric*. Essa *blockchain* requer que sejam especificadas, durante a

Figura 56 - Resultado em formato JSON da consulta (Q3).

```

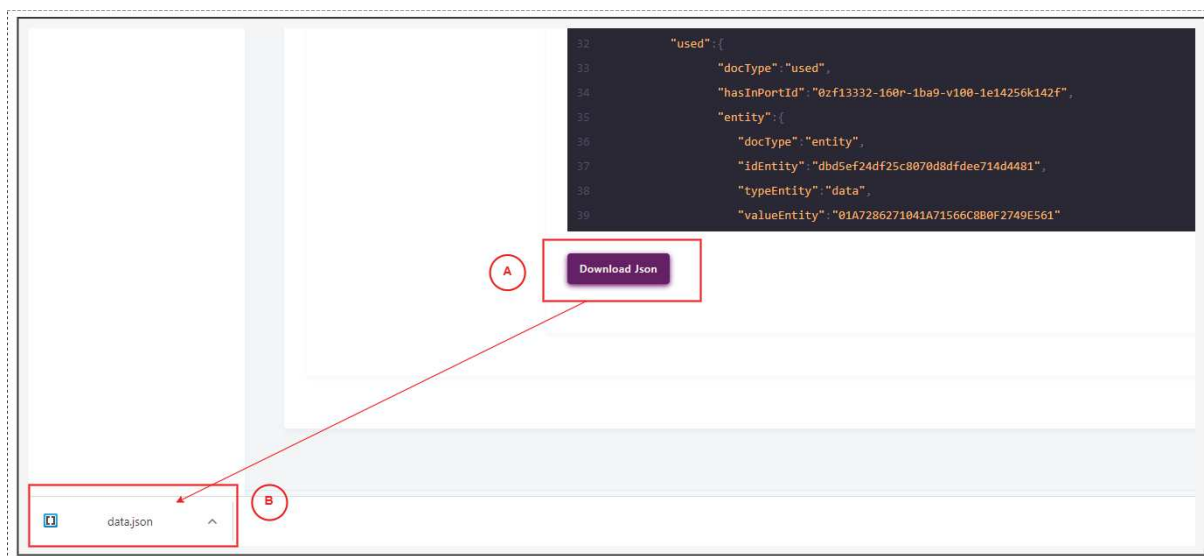
1 {
2   "Workflow": [
3     {
4       "_id": "9-345221453033289b5763762199303328",
5       "docType": "workflow",
6       "workFlowName": "Vireport_Workflow",
7       "programExecution": [
8         {
9
10          "docType": "programExecution",
11          "startTime": "2021-03-17T16:40:28.174Z",
12          "nameProgramExecution": "sequence_alignment_exe",
13        },
14        {
15          "docType": "programExecution",
16          "startTime": "2021-03-17T17:10:22.364Z",
17          "nameProgramExecution": "phylogenetic_inference_exe",
18        }
19      ]
20    },
21    {
22      "_id": "28b31185-db3a-4596-b5e9-562fd1aaf7ea",
23      "docType": "workflow",
24      "workFlowName": "Sciphy_Workflow",
25      "programExecution": [
26        {
27
28          "docType": "programExecution",
29          "startTime": "2021-03-17T17:40:28.174Z",
30          "nameProgramExecution": "multiple_sequence_alignt_ex",
31        },
32        {
33          "docType": "programExecution",
34          "startTime": "2021-03-17T18:02:28.362Z",
35          "nameProgramExecution": "sequence_conversion_exe",
36        }
37      ]
38    },

```

Fonte: Elaborada pelo autor.

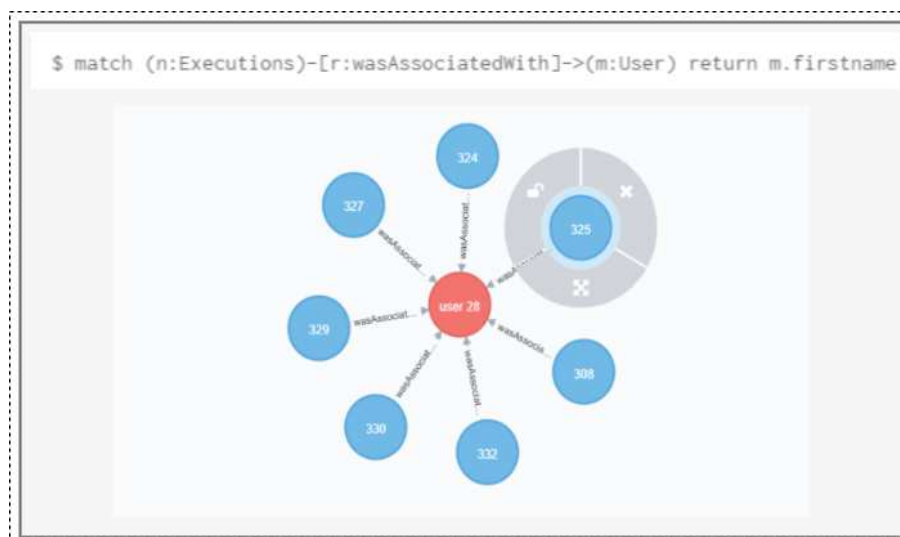
criação do ambiente de execução, as organizações e pares geograficamente distribuídos que poderiam colaborar no experimento e que fariam parte de um canal. Os canais mantêm

Figura 57 - Dowloand dados formato JSON.



Fonte: Elaborada pelo autor.

Figura 58 - Pesquisa em Cypher.



Fonte: Elaborada pelo autor.

privacidade, confidencialidade e isolam atividades entre partes autorizadas. E por fim, também mencionamos durante a execução dessa PoC que para transacionar e enviar dados de proveniência, os nós participantes precisam se inscrever e ter identidades, além de terem que enviar durante a coleta de proveniência o *Token* como uma maneira de confirmar a autenticidade e recuperar sua identidade como usuário pertencente à rede *blockchain*, para então garantir que os dados fossem assinados na rede. Assim, podemos verificar que

Figura 59 - Consulta entity(s) *workflow*.

```

1 {
2   "selector": {
3     "docType": {
4       "$eq": "entity"
5     },
6     "workflowID": {
7       "$eq": "9-345221453033289b5763762199303328"
8     }
9   },
10  "fields": [
11    "idEntity",
12    "startTime",
13    "typeEntity",
14    "valueEntity",
15    "_id",
16    "_rev"
17  ]
18 }

```

Fonte: Elaborada pelo autor.

Figura 60 - Comparação objeto de pesquisa (dados) usados ou gerados em um experimento.

Hash	Name	Size File
0D12C7EE86EBFB7E915FF635801BF18F	sequencias.fastas	1.732,551 Bytes
EE1D2782A1646B72117D0AFA3D08FF7C	sequencias_align.fasta	2.517,152 Bytes
92BF4313A07FA37E1D68D5F2F4BEDBC7	sequencias_align.fasta.phylip	3.363,144 Bytes


```

11  "entity": [
12    {
13      "docType": "entity",
14      "idEntity": "dbd5ef24df25c8070d8dfdee714d4481",
15      "typeEntity": "data",
16      "valueEntity": "0D12C7EE86EBFB7E915FF635801BF18F"
17    },
18    {
19      "docType": "entity",
20      "idEntity": "5fa353395f34221386779d99abe9814c",
21      "typeEntity": "data",
22      "valueEntity": "EE1D2782A1646B72117D0AFA3D08FF7C"
23    },
24    {
25      "docType": "entity",
26      "idEntity": "699586a363e2adfa9dc4dc96dfbd008d",
27      "typeEntity": "data",
28      "valueEntity": "92BF4313A07FA37E1D68D5F2F4BEDBC7"
29    }
30  ]

```

Fonte: Elaborada pelo autor.

a BlockFlow pode ser usada como um ambiente que fornece privacidade aos dados de proveniência, onde os dados são compartilhados apenas entre partes ou pessoas autorizadas,

mantendo assim a confidencialidade.

QS5) – *A BlockFlow pode ser usada como um ambiente de experimentação científica colaborativa, considerando a reprodutibilidade?*

Conforme ressaltado, um aspecto crítico associado a um processo científico é a reprodutibilidade e relacionada a esse princípio está a proveniência, que auxilia na compreensão dos resultados de um experimento científico. Assim, conforme podemos observar na execução dessa PoC, na seção 4.6.1, a Blockflow permite a coleta de proveniência, o seu armazenamento imutável (como uma forma eficaz para proteger a sua integridade, para que para não haja controvérsias ou fraudes indesejáveis) e a consulta de proveniência de uma maneira transparente e confiável. E por último, permite a verificação dos dados utilizados e gerados durante a execução dos *workflows* executados. Assim, podemos verificar que a BlockFlow pode ser usada como um ambiente de experimentação científica colaborativa, considerando a reprodutibilidade, uma vez que permite que pesquisadores possam avaliar proveniência, ou seja, quem?, quando? ou como? um dado foi gerado ao longo da execução de um workflow.

(QP) – **Como a arquitetura BlockFlow pode auxiliar cientistas nos experimentos científicos colaborativos, oferecendo um ambiente confiável apoiando a interoperabilidade, privacidade, transparência e reprodutibilidade de *workflows* científicos?**

Assim, considerando os resultados das questões de pesquisas secundárias, pode se então verificar indícios que a BlockFlow **oferece um ambiente confiável apoiando a interoperabilidade, privacidade, transparência e reprodutibilidade de *workflows* científicos**. A arquitetura oferece componentes que podem facilitar a colaboração na experimentação científica, considerando a reprodutibilidade (QS5), interoperabilidade (QS2), transparência (QS1), privacidade (QS4) e confiabilidade (QS3) de *workflows* científicos intensivos em dados, além da correta interpretação dos dados científicos entre pesquisadores geograficamente distribuídos, a partir da consulta aos dados.

No entanto, é importante observar que novos experimentos devem ser conduzidos de forma a validar os resultados dessa PoC. Além disso, os resultados apresentados só são válidos para este conjunto de dados. No entanto, podemos verificar cenários similares onde resultados semelhantes podem ser alcançados.

5 CONSIDERAÇÕES FINAIS

Este capítulo apresenta as considerações finais dessa dissertação e também, as principais contribuições e limitações, bem como os trabalhos futuros.

Essa dissertação apresentou a Blockflow uma arquitetura baseada em *blockchain*, cujo objetivo é apoiar a confidencialidade, transparência, interoperabilidade e reprodutibilidade na pesquisa colaborativa. A solução proposta está integrada a Plataforma de Ecossistema *Software Científico* chamada E-SECO. E tem como foco prover mecanismos que tragam maior confiabilidade aos dados e processos em *workflows* científicos colaborativos. Para embasar a abordagem proposta, esta dissertação apresentou os principais conceitos relacionados a e-Science, *blockchain*, dados de proveniência, computação em nuvem e a plataforma E-SECO. O presente trabalho também apresentou uma contribuição através de um mapeamento sistemático da literatura, que identificou e categorizou os principais trabalhos existentes no domínio de *blockchain* como mecanismo e benefícios para dados de proveniência.

Através de exemplos e cenários de aplicação, apresentados no capítulo 4 discutimos a viabilidade da proposta em apoiar cientistas a trabalharem de maneira colaborativa e distribuída, compartilhando dados de proveniência de uma maneira mais confiável, com intuito de garantir transparência e a reprodutibilidade dos resultados obtidos. Também, a execução de *workflows* intensivos em dados, ancorados pelo paradigma de computação em nuvem, através de infraestruturas de *cloud*. E por último, apoiar sistemas que necessitam de interoperabilidade e reutilização dos resultados de *workflows* científicos, integrando proveniência proveniente de diferentes Sistemas de Gerenciamento de *Workflows* Científico (*Scientific Workflow Management System - SWffs*) e, no que lhe concerne, aumentando a eficiência na pesquisa colaborativa. Por meio desse cenário, respondemos às questões de pesquisas onde foi possível observar que a solução potencializa a colaboração científica ao fornecer meios de transparência, reprodutibilidade, confiabilidade e reduz a heterogeneidade dos dados compartilhados em *workflow* científicos colaborativos, além de facilitar a interpretação e análise desses dados por pesquisadores geograficamente distribuídos.

O presente trabalho também apresentou as seguintes contribuições:

- Uma *API RESTful Webservice* para que a solução proposta pudesse ser conectada com outros aplicativos e plataformas que têm como objetivo permitir que seus usuários criem redes *blockchain* para colaborar e garantir a confiança e reprodutibilidade de experimentos científicos.
- Um facilitador para criação de ambientes colaborativos e de redes *blockchains* baseado em GUI que permite pesquisadores implementem de forma fácil suas redes *blockchain*

para então colaborarem.

- A especificação de ambientes colaborativos e redes *blockchain* ancorados pelo paradigma de computação em nuvem, através de infraestruturadas de *cloud*, para execução de *workflows* intensivos em dados.
- A Especificação e implementação de um coletor de proveniência que utiliza a tecnologia de serviços *web* e a *API RESTful WebService* para captura de proveniência.
- Um *wrapper* que traduz e integra os dados heterogêneos de proveniência, vindos de diferentes SWMS, para o formato do modelo ProvONE, que é utilizado como modelo padrão e integrador na BlockFlow.
- O armazenamento imutável, o gerenciamento e um facilitador para a consulta, análise e visualizações de informações de proveniência dos experimentos científicos colaborativos, executados.
- A exportação dos dados coletados de proveniência para o modelo JSON, possibilitando que os dados sejam integrados com outras plataformas.
- A possibilidade de *upload* de dados utilizados e gerados durante a execução de um experimento, facilitando analisar se o objeto de pesquisa (dados) usados ou gerados em um experimento possui o conteúdo equivalente ao que foi publicado e compartilhado no experimento.

Este trabalho foi desenvolvido para aumentar a reprodutibilidade, privacidade, transparência e interoperabilidade em ecossistemas de *software* científico. Portanto, dados de proveniência fornecidos por meio desta abordagem são limitados a este objetivo e não pode ser generalizado. No entanto, o conhecimento construído e os resultados obtidos podem ser transferidos para outros contextos. Além disso, uma das desvantagens e limitações da arquitetura está no fato de que em um aplicativo baseado em *blockchain*, o armazenamento de arquivos propriamente ditos não é possível, sendo necessário armazenar *hashes* de informações. Embora esta limitação possa ser superada com a *blockchain* IPFS, no BlockFlow, ainda compartilhamos todos os dados de entrada e saída gerados durante a execução do fluxo de trabalho colaborativo fora da cadeia, como detalhado na seção 4.6.1. Desta forma, é necessário verificar a integridade dos dados conforme detalhado na seção 4.6.3, comparando se o *hash* armazenado corresponde aos dados usados como entrada e saída durante a execução do fluxo de trabalho.

Como trabalhos futuros, pretendemos, realizar a condução de novos estudos de caso, em outros contextos, a fim de avaliar o apoio oferecido pela abordagem.

REFERÊNCIAS

- 1 ANDROULAKI, Elli; BARGER, Artem; BORTNIKOV, Vita; CACHIN, Christian; CHRISTIDIS, Konstantinos; DE CARO, Angelo; ENYEART, David; FERRIS, Christopher; LAVENTMAN, Gennady; MANEVICH, Yacov; MURALIDHARAN, Srinivasan; MURTHY, Chet; NGUYEN, Binh; SETHI, Manish; SINGH Gari; SMITH, Keith; SORNIOTTI, Alessandro; STATHAKOPOULOU, Chrysoula; VUKOLIĆ, Marko; COCCO, Sharon Weed; YELLICK Jason. Hyperledger fabric: a distributed operating system for permissioned blockchains. **In: Proceedings of the thirteenth EuroSys conference**. 2018. p. 1-15.
- 2 AMBRÓSIO, Lenita M; DAVID, José Maria N; BRAGA, Regina MM; CAMPOS, Fernanda; STRÖELE, Victor; ARAÚJO, Marco Antônio. Using Context Elements and Data Provenance to Support Reuse in Scientific Software Ecosystem Platform. **In: ICEIS (2)**. 2018. p. 255-262.
- 3 AMBRÓSIO, Lenita. **Apoiando o Reúso em uma Plataforma de Ecossistema de Software Científico Através do Gerenciamento de Contexto e de Proveniência**. 2018. Dissertação (Mestrado em Ciência da Computação) - Instituto de Ciências Exatas da Universidade Federal de Juiz de Fora - MG. 2018.
- 4 ANSORGE, Wilhelm J. Next-generation DNA sequencing techniques. *New biotechnology*, v. 25, n. 4, p. 195-203, 2009.
- 5 BAKER, Monya. 1,500 scientists lift the lid on reproducibility. **Nature News**, v. 533, n. 7604, p. 452, 2016.
- 6 BAKER, Monya. Reproducibility crisis. **Nature**, v. 533, n. 26, p. 353-66, 2016.
- 7 BEGLEY, C. Glenn; ELLIS, Lee M. Raise standards for preclinical cancer research. **Nature**, v. 483, n. 7391, p. 531-533, 2012.
- 8 BEGLEY, C. Glenn; IOANNIDIS, John PA. Reproducibility in science: improving the standard for basic and preclinical research. **Circulation research**, v. 116, n. 1, p. 116-126, 2015.
- 9 BELL, David A.. . From data properties to evidence. **IEEE Transactions on Knowledge and Data Engineering**, v. 5, n. 6, p. 965-969, 1993.
- 10 BELLOUM, Adam; INDA, Marcia A; VASUNIN, Dmitry; KORKHOV, Vladimir; ZHAO, Zhiming; RAUWERDA, Han; BREIT, Timo M; BUBAK, Marian; HERTZBERGER, Luis O. Collaborative e-science experiments and scientific workflows. **IEEE Internet Computing**, v. 15, n. 4, p. 39-47, 2011.
- 11 BHUYAN, Fahima Amin; LU, Shiyong; REYNOLDS, Robert; ZHANG, Jia; AHMED, Ishtiaq. A Security Framework for Scientific Workflow Provenance Access Control Policies. **IEEE Transactions on Services Computing**, 2019.
- 12 BIK, Elisabeth M.; CASADEVALL, Arturo; FANG, Ferric C. The prevalence of inappropriate image duplication in biomedical research publications. **MBio**, v. 7, n. 3, 2016.

- 13 BOSCH, J. From Software Product Lines to Software Ecosystems. **SPLC**, 2009, Pittsburgh, PA, USA: Proceedings of the 13th International Software Product Line Conference, 2009. p.111– 119
- 14 BUTERIN, Vitalik et al. A next-generation smart contract and decentralized application platform. **white paper**, v. 3, n. 37, 2013.
- 15 CALLAHAN, Steven P; FREIRE, Juliana; SANTOS, Emanuele; SCHEIDEGGER, Carlos E; SILVA, Cláudio T; VO, Huy T. et al. VisTrails: visualization meets data management. In: **Proceedings of the 2006 ACM SIGMOD international conference on Management of data. 2006. p. 745-747.**
- 16 CHEN, Wanghu; LIANG, Xiaoyan; LI, Jing; QIN, Hongwu; MU, Yuxiang; WANG, Jianwu. Blockchain based provenance sharing of scientific workflows. In: **w2018 IEEE International Conference on Big Data (Big Data)**. IEEE, 2018. p. 3814-3820.
- 17 CHIRIGATI, Fernando; RAMPIN, Rémi; SHASHA, Dennis; FREIRE, Juliana. Reprozip: Computational reproducibility with ease. In: **Proceedings of the 2016 international conference on management of data. 2016. p. 2085-2088.**
- 18 CLASSE, Tadeu; BRAGA, Regina; DAVID, José Maria N; CAMPOS, Fernanda; ARAÚJO, Marco Antônio; STRÖELE, Victor. A collaborative approach to support e-science activities. In: **2016 IEEE 20th International Conference on Computer Supported Cooperative Work in Design (CSCWD)**. IEEE, 2016. p. 20-25.
- 19 COELHO, Raiane; BRAGA, Regina; DAVID, José Maria N and DANTAS, Mário; STRÖELE, Victor; CAMPOS, Fernanda. Blockchain for reliability in collaborative scientific workflows on cloud platforms. In: **2020 IEEE Symposium on Computers and Communications (ISCC)**. IEEE, 2020. p. 1-7.
- 20 COELHO, Raiane; BRAGA, Regina; DAVID, José Maria N and DANTAS, Mário; STRÖELE, Victor; CAMPOS, Fernanda. Integrating blockchain for data sharing and collaboration support in scientific ecosystem platform. In: **Proceedings of the 54th Hawaii International Conference on System Sciences**. 2021. p. 264.
- 21 COHEN-BOULAKIA, Sarah; BELHAJJAME, Khalid; COLLIN, Olivier; CHOPARD, Jérôme; FROIDEVAUX, Christine; GAIGNARD, Alban; HINSEN, Konrad; LARMANDE, Pierre; LE BRAS, Yvan; LEMOINE, Frédéric; MAREUIL, Fabien; MÉNAGER, Hervé; PRADAL, Christophe; BLANCHET, Christophe. Scientific workflows for computational reproducibility in the life sciences: Status, challenges and opportunities. **Future Generation Computer Systems**, v. 75, p. 284-298, 2017.
- 22 COSTA, Flavio; DE OLIVEIRA, Daniel; MATTOSO, Marta. Towards an adaptive and distributed architecture for managing workflow provenance data. In: **2014 IEEE 10th International Conference on e-Science**. IEEE, 2014. p. 79-82.
- 23 CUEVAS-VICENTTÍN, Víctor et al. ProvONE: A prov extension data model for scientific workflow provenance. 2015.
- 24 CUEVAS-VICENTTÍN, Víctor; KIANMAJD, Parisa; LUDÄSCHER, Bertram; MISSIER, Paolo; CHIRIGATI, Fernando; WEI, Yaxing; KOOP, David; DEY, Saumen. The PBase scientific workflow provenance repository. 2014.

- 25 DAVIDSON, Susan B.; FREIRE, Juliana. Provenance and scientific workflows: challenges and opportunities. In: **Proceedings of the 2008 ACM SIGMOD international conference on Management of data**. 2008. p. 1345-1350.
- 26 DE OLIVEIRA, Daniel; BAIÃO, Fernanda Araujo; MATTOSO, Marta. Towards a taxonomy for cloud computing from an e-science perspective. In: **Cloud computing**. Springer, London, 2010. p. 47-62.
- 27 DE OLIVEIRA, Daniel; OGASAWARA, Eduardo; BAIÃO, Fernanda; MATTOSO, Marta. Scicumulus: A lightweight cloud middleware to explore many task computing paradigm in scientific workflows. In: **2010 IEEE 3rd International Conference on Cloud Computing**. IEEE, 2010. p. 378-385.
- 28 DE OLIVEIRA, Daniel CM; LIU, Ji; PACITTI, Esther. Data-intensive workflow management: for clouds and data-intensive and scalable computing environments. **Synthesis Lectures on Data Management**, v. 14, n. 4, p. 1-179, 2019.
- 29 DEELMAN, Ewa; MEHTA, Gaurang; SINGH, Gurmeet; SU, Mei-Hui; VAHI, Karan. Pegasus: mapping large-scale workflows to distributed resources. In: **Workflows for e-Science**. Springer, London, 2007. p. 376-394.
- 30 FANELLI, Daniele. Opinion: Is science really facing a reproducibility crisis, and do we need it to?. **Proceedings of the National Academy of Sciences**, v. 115, n. 11, p. 2628-2631, 2018.
- 31 FANNING, Kurt; CENTERS, David P. Blockchain and its coming impact on financial services. **Journal of Corporate Accounting Finance**, v. 27, n. 5, p. 53-57, 2016.
- 32 FERNANDO, Dinuni; KULSHRESTHA, Siddharth; HERATH, J Dinal; MAHADIK, Nitin; MA, Yanzhe; BAI, Changxin; YANG, Ping; YAN, Guanhua; LU, Shiyong. SciBlock: A blockchain-based tamper-proof non-repudiable storage for scientific workflow provenance. In: **2019 IEEE 5th International Conference on Collaboration and Internet Computing (CIC)**. IEEE, 2019. p. 81-90.
- 33 FORSTER, Peter; FORSTER, Lucy; RENFREW, Colin; FORSTER, Michael. Phylogenetic network analysis of SARS-CoV-2 genomes. **Proceedings of the National Academy of Sciences**, v. 117, n. 17, p. 9241-9243, 2020.
- 34 FRASER, Hannah; PARKER, Tim; NAKAGAWA, Shinichi; BARNETT, Ashley; FIDLER, Fiona. Questionable research practices in ecology and evolution. **PloS one**, v. 13, n. 7, p. e0200303, 2018.
- 35 FREIRE, Juliana; CHIRIGATI, Fernando. Provenance and the different flavors of computational reproducibility. **IEEE Data Engineering Bulletin**, v. 41, n. 1, p. 15, 2018.
- 36 FREIRE, Juliana; KOOP, David; SANTOS, Emanuele; SILVA, Cláudio. T. Provenance for Computational Tasks: A Survey, *Computing in Science Engineering*, v. 10, n. 3, p. 11-21, 2008.
- 37 FREITAS, V; DAVID, José Maria N; BRAGA, Regina; CAMPOS, Fernanda. An architecture for scientific software ecosystem. In: **9th Workshop on Distributed Software Development, Software Ecosystems and Systems-of-Systems (WDES 2015)**. 2015. p. 41-48.

- 38 GEORGE, Stephen L.; BUYSE, Marc. Data fraud in clinical trials. **Clinical investigation**, v. 5, n. 2, p. 161, 2015.
- 39 GOECKS, Jeremy; NEKRUTENKO, Anton; TAYLOR, James. Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. **Genome biology**, v. 11, n. 8, p. 1-13, 2010.
- 40 GROTH, Paul; MOREAU, Luc. PROV-overview. An overview of the PROV family of documents. 2013.
- 41 HEIDSIECK, Gaëtan; DE OLIVEIRA, Daniel; PACITTI, Esther; PRADAL, Christophe; TARDIEU, Francois; VALDURIEZ, Patrick. Distributed caching of scientific workflows in multisite cloud. In: **International Conference on Database and Expert Systems Applications**. Springer, Cham, 2020. p. 51-65.
- 42 HERSCHEL, Melanie; DIESTELKÄMPER, Ralf; LAHMAR, Housseem Ben. A survey on provenance: What for? What form? What from?. The **VLDB Journal**, v. 26, n. 6, p. 881-906, 2017.
- 43 HEVNER, Alan R; MARCH, Salvatore T; PARK, Jinsoo; RAM, Sudha. Design science in information systems research. **MIS quarterly**, p. 75-105, 2004.
- 44 HEVNER, Alan R; MARCH, Salvatore T; PARK, Jinsoo; RAM, Sudha. Design science in information systems research. **Management Information Systems Quarterly**, v. 28, n. 1, p. 6, 2008.
- 45 HEY, Tony; TANSLEY, Stewart; TOLLE, Kristin M et al. **The fourth paradigm: data-intensive scientific discovery**. [S.l.]: Microsoft research Redmond, WA, 2009.
- 46 HEY, Tony; TREFETHEN, Anne. The fourth paradigm 10 years on. **Informatik Spektrum**, v. 42, n. 6, p. 441-447, 2020.
- 47 HIMANEN, Lauri; GEURTS, Amber; FOSTER, Adam Stuart; RINKE, Patrick. Data-driven materials science: status, challenges, and perspectives. **Advanced Science**, v. 6, n. 21, p. 1900808, 2019.
- 48 JAIMES, Javier A; ANDRÉ, Nicole M; CHAPPIE, Joshua S; MILLET, Jean K; WHITTAKER, Gary R. Phylogenetic analysis and structural modeling of SARS-CoV-2 spike protein reveals an evolutionary distinct and proteolytically sensitive activation loop. **Journal of molecular biology**, v. 432, n. 10, p. 3309-3325, 2020.
- 49 JANDRE, Eduardo; DIIRR, Bruna; BRAGANHOLO, Vanessa. Provenance in collaborative in silico scientific research: a survey. **ACM SIGMOD Record**, v. 49, n. 2, p. 36-51, 2020.
- 50 KARASTOYANOVA, Dimka; STAGE, Ludwig. Towards collaborative and reproducible scientific experiments on blockchain. In: **International Conference on Advanced Information Systems Engineering**. Springer, Cham, 2018. p. 144-149.
- 51 KHAN, Samiya; ALI, Syed Arshad; HASAN, Nabeela; SHAKIL, Kashish Ara; ALAM, Mansaf. Big data scientific workflows in the cloud: Challenges and future prospects. In: **Cloud computing for geospatial big data analytics**. Springer, Cham, 2019. p. 1-28.

- 52 KIM, Dongwan; LEE, Joo-Yeon; YANG, Jeong-Sun; KIM, Jun Won; KIM, V Narry; CHANG, Hyeshik. The architecture of SARS-CoV-2 transcriptome. *Cell*, v. 181, n. 4, p. 914-921. e10, 2020.
- 53 KIM, Henry M.; LASKOWSKI, Marek. Toward an ontology-driven blockchain design for supply-chain provenance. **Intelligent Systems in Accounting, Finance and Management**, v. 25, n. 1, p. 18-27, 2018.
- 54 KITCHENHAM, Barbara. Procedures for performing systematic reviews. **Keele, UK, Keele University**, v. 33, n. 2004, p. 1–26, 2004.
- 55 KITCHENHAM, Barbara; CHARTERS, Stuart. **Guidelines for performing systematic literature reviews in software engineering**. 2007.
- 56 KOOP, David; FREIRE, Juliana. Reorganizing workflow evolution provenance. In: **6th USENIX Workshop on the Theory and Practice of Provenance (TaPP 2014)**. 2014.
- 57 KUMAR, Swatantra et al. Morphology, genome organization, replication, and pathogenesis of severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2). In: **Coronavirus Disease 2019 (COVID-19)**. Springer, Singapore, 2020. p. 23-31.
- 58 LANDER, Eric S. et al. Initial sequencing and analysis of the human genome. 2001.
- 59 LIANG, Xueping; SHETTY, Sachin; TOSH, Deepak; KAMHOUA, Charles; KWIAT, Kevin; NJILLA, Laurent. Provchain: A blockchain-based data provenance architecture in cloud environment with enhanced privacy and availability. In: **2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)**. IEEE, 2017. p. 468-477.
- 60 LIM, Chunhyeok; LU, Shiyong; CHEBOTKO, Artem; FOTOUHI, Farshad. Prospective and retrospective provenance collection in scientific workflow environments. In: **2010 IEEE International Conference on Services Computing**. IEEE, 2010. p. 449-456.
- 61 LIU, Ji; PACITTI, Esther; VALDURIEZ, Patrick; MATTOSO, Marta. A survey of data-intensive scientific workflow management. **Journal of Grid Computing**, v. 13, n. 4, p. 457-493, 2015.
- 62 Ludäscher, Bertram; Altintas, Ilkay; Berkley, Chad; Higgins, Dan; Jaeger, Efrat; Jones, Matthew; Lee, Edward A; Tao, Jing; Zhao, Yang. Scientific workflow management and the Kepler system. **Concurrency and computation: Practice and experience**, v. 18, n. 10, p. 1039-1065, 2006.
- 63 MAGEE, Andrew F.; MAY, Michael R.; MOORE, Brian R. The dawn of open access to phylogenetic data. **PLoS One**, v. 9, n. 10, p. e110268, 2014.
- 64 MAKEL, Matthew C.; PLUCKER, Jonathan A.; HEGARTY, Boyd. Replications in psychology research: How often do they really occur?. **Perspectives on Psychological Science**, v. 7, n. 6, p. 537-542, 2012.
- 65 MANIKAS, Konstantinos. Revisiting software ecosystems research: A longitudinal literature study. **Journal of Systems and Software**, v. 117, p. 84-103, 2016.

- 66 MARQUES, Philipe; DAVID, José Maria; STRÖELE, Victor; BRAGA, Regina; CAMPOS, Fernanda; ARAÚJO, Marco Antônio. Apoiando a Composição de Serviços em Ecossistemas de Software Científico. In: **Anais do XIV Simpósio Brasileiro de Sistemas Colaborativos**. SBC, 2017. p. 183-197.
- 67 MATTOSO, Marta; WERNER, Claudia; TRAVASSOS, Guilherme Horta; BRAGANHOLO, Vanessa; OGASAWARA, Eduardo; DE OLIVEIRA, Daniel; CRUZ, Sergio; MARTINHO, Wallace; MURTA, Leonardo. Towards supporting the life cycle of large scale scientific experiments. **International Journal of Business Process Integration and Management**, v. 5, n. 1, p. 79-92, 2010.
- 68 MCNUTT, Marcia. Reproducibility. 2014.
- 69 MENDES, Yan; BRAGA, Regina; STRÖELE, Victor; DE OLIVEIRA, Daniel. Polyflow: A soa for analyzing workflow heterogeneous provenance data in distributed environments. In: **Proceedings of the XV Brazilian Symposium on Information Systems**. 2019. p. 1-8.
- 70 MISSIER, Paolo; LUDÄSCHER, Bertram; BOWERS, Shawn; DEY, Saumen; SARKAR, Anandarup; SHRESTHA, Biva; ALTINTAS, Ilkay; ANAND, Manish Kumar; GOBLE, Carole. Linking multiple workflow provenance traces for interoperable collaborative science. In: **The 5th Workshop on Workflows in Support of Large-Scale Science**. IEEE, 2010. p. 1-8.
- 71 MISSIER, Paolo; WOODMAN, Simon; HIDEN, Hugo; WATSON, Paul. Provenance and data differencing for workflow reproducibility analysis. **Concurrency and Computation: Practice and Experience**, v. 28, n. 4, p. 995-1015, 2016.
- 72 MISSIER, Paolo; SOILAND-REYES, Stian; OWEN, Stuart; TAN, Wei; NENADIC, Alexandra; DUNLOP, Ian; WILLIAMS, Alan; OINN, Tom; GOBLE, Carole. Taverna, reloaded. In: **International conference on scientific and statistical database management**. Springer, Berlin, Heidelberg, 2010. p. 471-481.
- 73 MISSIER, Paolo; BELHAJJAME, Khalid; CHENEY, James. The W3C PROV family of specifications for modelling provenance metadata. In: **Proceedings of the 16th International Conference on Extending Database Technology**. 2013. p. 773-776.
- 74 MIYAKAWA, Tsuyoshi. No raw data, no science: another possible source of the reproducibility crisis. 2020.
- 75 MOREAU, Luc; FREIRE, Juliana; FUTRELLE, Joe; MCGRATH, Robert E; MYERS, Jim; PAULSON, Patrick. The open provenance model: An overview. In: **International provenance and annotation workshop**. Springer, Berlin, Heidelberg, 2008. p. 323-326.
- 76 NAKAMOTO, Satoshi. Bitcoin: A peer-to-peer electronic cash system. 2008.
- 77 NEIVA, Frâncila Weidt; DAVID, José Maria N; BRAGA, Regina; CAMPOS, Fernanda; FREITAS, Vitor. PRIME: Pragmatic interoperability architecture to support collaborative development of scientific workflows. In: **2015 IX Brazilian Symposium on Components, Architectures and Reuse Software**. IEEE, 2015. p. 50-59.

- 78 OCAÑA, Kary ACS; DE OLIVEIRA, Daniel; HORTA, Felipe; DIAS, Jonas; OGASAWARA, Eduardo; MATTOSO, Marta. Exploring molecular evolution reconstruction using a parallel cloud based scientific workflow. In: **Brazilian Symposium on Bioinformatics**. Springer, Berlin, Heidelberg, 2012. p. 179-191.
- 79 OCAÑA, Kary ACS; DE OLIVEIRA, Daniel; OGASAWARA, Eduardo; DÁVILA, Alberto MR; LIMA, Alexandre AB; MATTOSO, Marta. SciPhy: a cloud-based workflow for phylogenetic analysis of drug targets in protozoan genomes. In: **Brazilian Symposium on Bioinformatics**. Springer, Berlin, Heidelberg, 2011. p. 66-70.
- 80 OGASAWARA, Eduardo; DIAS, Jonas; SILVA, Vitor; CHIRIGATI, Fernando; DE OLIVEIRA, Daniel; PORTO, Fabio; VALDURIEZ, Patrick; MATTOSO, Marta. Chiron: a parallel engine for algebraic scientific workflows. *Concurrency and Computation: Practice and Experience*, v. 25, n. 16, p. 2327-2341, 2013.
- 81 OGASAWARA, Eduardo; MURTA, Leonardo; WERNER, Cláudia; MATTOSO, Marta. Linhas de experimento: Reutilização e gerência de configuração em workflows científicos. In: **2 Workshop E-Science**. 2008. p. 31-40.
- 82 OLIVEIRA, Wellington; MISSIER, Paolo; OCAÑA, Kary; DE OLIVEIRA, Daniel; BRAGANHOLO, Vanessa. Analyzing provenance across heterogeneous provenance graphs. In: **International Provenance and Annotation Workshop**. Springer, Cham, 2016. p. 57-70.
- 83 PENG, Roger. The reproducibility crisis in science: A statistical counterattack. *Significance*, v. 12, n. 3, p. 30-32, 2015.
- 84 PEREIRA, Anrafel F; DAVID, José Maria N; BRAGA, Regina; CAMPOS, Fernanda. An architecture to enhance collaboration in scientific software product line. In: **2016 49th Hawaii International Conference on System Sciences (HICSS)**. IEEE, 2016. p. 338-347.
- 85 PETTICREW, Mark; ROBERTS, Helen. **Systematic reviews in the social sciences: A practical guide**. John Wiley Sons, 2008.
- 86 POUCHARD, Line; BALDWIN, Sterling; ELSETHAGEN, Todd; JHA, Shantenu; RAJU, Bibi; STEPHAN, Eric; TANG, Li; VAN DAM, Kerstin Kleese. Computational reproducibility of scientific workflows at extreme scales. *The International Journal of High Performance Computing Applications*, v. 33, n. 5, p. 763-776, 2019.
- 87 PRINZ, Florian; SCHLANGE, Thomas; ASADULLAH, Khusru. Believe it or not: how much can we rely on published data on potential drug targets?. *Nature reviews Drug discovery*, v. 10, n. 9, p. 712-712, 2011.
- 88 RAMACHANDRAN, Aravind; KANTARCIOGLU, Murat. Smartprovenance: a distributed, blockchain based dataprovenance system. In: **Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy**. 2018. p. 35-42.
- 89 REID, Fergal; HARRIGAN, Martin. An analysis of anonymity in the bitcoin system. In: **Security and privacy in social networks**. Springer, New York, NY, 2013. p. 197-223.

- 90 ROBERTS, Keith; ALBERTS, Bruce; JOHNSON, Alexander; WALTER, Peter; HUNT, Tim. *Molecular biology of the cell*. New York: Garland Science, 2002.
- 91 SANTANA-PEREZ, Idafen; PÉREZ-HERNÁNDEZ, María S. Towards reproducibility in scientific workflows: An infrastructure-based approach. **Scientific Programming**, v. 2015, 2015.
- 92 SHULL, Forrest; MENDONÇA, Manoel G; BASILI, Victor; CARVER, Jeffrey; MALDONADO, José C; FABBRI, Sandra; TRAVASSOS, Guilherme Horta; FERREIRA, Maria Cristina. Knowledge-sharing issues in experimental software engineering. **Empirical Software Engineering**, v. 9, n. 1, p. 111-137, 2004.
- 93 SILVA, Claudio T.; FREIRE, Juliana; CALLAHAN, Steven P. Provenance for visualizations: Reproducibility and beyond. **Computing in Science Engineering**, v. 9, n. 5, p. 82-89, 2007.
- 94 SIRQUEIRA, Tássio FM; DALPRA, Humberto LO; BRAGA, Regina; ARAÚJO, Marco Antônio P; DAVID, José Maria N; CAMPOS, Fernanda. E-seco proversion: An approach for scientific workflows maintenance and evolution. **Procedia Computer Science**, v. 100, p. 547-556, 2016.
- 95 SONG, Miranda., MOSHIRI, Niema. (2020). An Analysis of SARS-CoV-2 Using ViReport.
- 96 SZABO, Nick. *Smart contracts*. 1994.
- 97 TENOPIR, Carol; DALTON, Elizabeth D; ALLARD, Suzie; FRAME, Mike; PJESIVAC, Ivanka; BIRCH, Ben; POLLOCK, Danielle; DORSETT, Kristina. Changes in data sharing and data reuse practices and perceptions among scientists worldwide. **PloS one**, v. 10, n. 8, p. e0134826, 2015.
- 98 TERZO, Olivier; MOSSUCCA, Lorenzo (Ed.). **Cloud Computing with E-science Applications**. Crc Press, 2017.
- 99 TOSH, Deepak K; SHETTY, Sachin; LIANG, Xueping; KAMHOUA, Charles; NJILLA, Laurent. Consensus protocols for blockchain-based data provenance: Challenges and opportunities. In: **2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)**. IEEE, 2017. p. 469-474.
- 100 TRAVEL, Telecom; MOHANTY, Debajani. R3 Corda for Architects and Developers.
- 101 TSCHORSCH, Florian; SCHEUERMANN, Björn. Bitcoin and beyond: A technical survey on decentralized digital currencies. **IEEE Communications Surveys Tutorials**, v. 18, n. 3, p. 2084-2123, 2016.
- 102 VAILATI-RIBONI, Mario; PALOMBO, Valentino; LOOR, Juan J. What are Omics Sciences?. In: **Periparturient Diseases of Dairy Cows**. Springer, Cham, 2017. p. 1-7.
- 103 VAN ROSSUM, Joris. Blockchain for research: Perspectives on a new paradigm for scholarly communication. **Digital Science**, November, 2017.

- 104 VAN SOLINGEN, Rini; BASILI, Vic; CALDIERA, Gianluigi; ROMBACH, H Dieter
Goal question metric (gqm) approach. **Encyclopedia of software engineering**,
2002.
- 105 WAGNER, Caroline S.; WAGNER, Caroline S.; GRABER. **Collaborative Era in
Science**. London: Palgrave Macmillan, 2018.
- 106 WAN, Shaohua; LI, Meijun; LIU, Gaoyang; WANG, Chen. Recent advances in
consensus protocols for blockchain: a survey. **Wireless networks**, v. 26, n. 8, p.
5579-5593, 2020.
- 107 WANG, Wenbo; HOANG, Dinh Thai; HU, Peizhao; XIONG, Zehui; NIYATO, Dusit;
WANG, Ping; WEN, Yonggang; KIM, Dong. In A survey on consensus mechanisms
and mining strategy management in blockchain networks. **IEEE Access**, v. 7, p.
22328-22370, 2019.
- 108 WILEY, Edward Orlando; LIEBERMAN, Bruce S. **Phylogenetics: theory and
practice of phylogenetic systematics**. John Wiley Sons, 2011.
- 109 WOHLIN, Claes; RUNESON, Per; HÖST, Martin; OHLSSON, Magnus C;
REGNELL, Björn; WESSLÉN, Anders. **Experimentation in software
engineering**. Springer Science Business Media, 2012.
- 110 WOZNIAK, Justin M.; ARMSTRONG, Timothy G.; WILDE, Michael; KATZ,
Daniel S.; LUSK, Ewing; FOSTER, Ian T. Swift/t: Large-scale application
composition via distributed-memory dataflow processing. In: **2013 13th
IEEE/ACM International Symposium on Cluster, Cloud, and Grid
Computing**. IEEE, 2013. p. 95-102.
- 111 XU, Xiwei; WEBER, Ingo; STAPLES, Mark. **Architecture for blockchain
applications**. Cham: Springer, 2019.
- 112 YAQINUDDIN, Ahmed. Cross-immunity between respiratory coronaviruses may
limit COVID-19 fatalities. **Medical Hypotheses**, v. 144, p. 110049, 2020.
- 113 YIN, R. K., Robert K. (2014). **Case Study Research Design and Methods**. Los
Angeles, CA: Sage.
- 114 ZHANG, Jia; KUC, Daniel; LU, Shiyong. Confucius: A tool supporting collaborative
scientific workflow composition. **IEEE Transactions on Services Computing**, v.
7, n. 1, p. 2-17, 2012.
- 115 ZHANG, Tao; WU, Qunfu; ZHANG, Zhigang. Probable pangolin origin of
SARS-CoV-2 associated with the COVID-19 outbreak. **Current biology**, v. 30, n. 7,
p. 1346-1351. e2, 2020.
- 116 ZHAO, Yong; FEI, Xubo; RAICU, Ioan; LU, Shiyong. Opportunities and challenges
in running scientific workflows on the cloud. In: **2011 International Conference
on Cyber-Enabled Distributed Computing and Knowledge Discovery**.
IEEE, 2011. p. 455-462.

- 117 ZHOU, Hong; CHEN, Xing; HU, Tao; Li Juan; SONG, Hao; LIU, Yanran; WANG, Peihan; LIU, Di; YANG, Jing; HOLMES, Edward C; HUGHES, Alice C; BI, Yuhai; SHI, Weifeng. A novel bat coronavirus closely related to SARS-CoV-2 contains natural insertions at the S1/S2 cleavage site of the spike protein. **Current Biology**, v. 30, n. 11, p. 2196-2203. e3, 2020.
- 118 ZHOU, Peng; YANG, Xing-Lou; WANG, Xian-Guang; HU, Ben; ZHANG, Lei; Zhang, Wei; SI, Hao-Rui; ZHU, Yan; LI, Bei; HUANG, Chao-Lin; CHEN, Hui-Dong; CHEN, Jing; LUO, Yun; GUO, Hua; JIANG, Ren-Di; LIU, Mei-Qin; CHEN, Ying; SHEN, Xu-Rui; WANG, Xi; ZHENG, Xiao-Shuang; ZHAO, Kai; CHEN, Quan-Jiao; DENG, Fei; LIU, Lin-Lin; YAN, Bing; ZHAN, Fa-Xian; WANG, Yan-Yi; XIAO, Geng-Fu; SHI, Zheng-Li. A pneumonia outbreak associated with a new coronavirus of probable bat origin. **nature**, v. 579, n. 7798, p. 270-273, 2020.