Unioeste - Universidade Estadual do Oeste do Paraná
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
Colegiado de Ciência da Computação
Curso de Bacharelado em Ciência da Computação

Qual o estado da arte das abordagens de desenvolvimento de software orientado a *blockchain*? - Um mapeamento sistemático.

Gilberto Antunes Monteiro Junior

CASCAVEL
2019

Gilberto Antunes Monteiro Junior

Qual o estado da arte das abordagens de desenvolvimento de software orientado a *blockchain*? - Um mapeamento sistemático.

Monografia apresentada como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação, do Centro de Ciências Exatas e Tecnológicas da Universidade Estadual do Oeste do Paraná - Campus de Cascavel

Orientador: Ivonei Freitas da Silva

CASCAVEL
2019

Gilberto Antunes Monteiro Junior

Qual o estado da arte das abordagens de desenvolvimento de software orientado a *blockchain*? - Um mapeamento sistemático.

Monografia apresentada como requisito parcial para obtenção do Título de Bacharel em Ciência da Computação, pela Universidade Estadual do Oeste do Paraná, Campus de Cascavel, aprovada pela Comissão formada pelos professores:

Ivonei Freitas da Silva (Orientador)
Colegiado de Ciência da Computação,
UNIOESTE

Edmar André Bellorini
Colegiado de Ciência da Computação,
UNIOESTE

Guilherme Galante
Colegiado de Ciência da Computação,
UNIOESTE

Cascavel, 03 de dezembro de 2019

DEDICATÓRIA

Dedico este trabalho a minha namorada Mariane Tidres, por ter me apoiado durante toda a jornada e não ter me deixado ficar doido, a meu pai Gilberto Antunes, minha mãe Lurdes Lopes e também minha tia Maria Elena Dal Santo por me ajudarem a manter-me firme e forte no caminho para mais esta conquista. Sem vocês jamais teria chegado até aqui. Muito obrigado!

“Let me tell you something you already know. The world ain’t all sunshine and rainbows. It’s a very mean and nasty place and I don’t care how tough you are it will beat you to your knees and keep you there permanently if you let it. You, me, or nobody is gonna hit as hard as life. But it ain’t about how hard ya hit. It’s about how hard you can get hit and keep moving forward. How much you can take and keep moving forward. That’s how winning is done! Now if you know what you’re worth then go out and get what you’re worth. But ya gotta be willing to take the hits, and not pointing fingers saying you ain’t where you wanna be because of him, or her, or anybody! Cowards do that and that ain’t you! You’re better than that!”. Robert “Rocky” Balboa

AGRADECIMENTOS

Agradeço a minha namorada Mariane Tidres, por ter me apoiado durante toda a jornada e não ter me deixado ficar doido, a meu pai Gilberto Antunes, minha mãe Lurdes Lopes e também minha tia Maria Elena Dal Santo por me ajudarem a manter-me firme e forte no caminho para mais esta conquista. Sem vocês jamais teria chegado até aqui. Muito obrigado!

Especial agradecimento ao meu orientador Ivonei Freitas da Silva, pela sua fundamental orientação neste trabalho e pelas longas conversas sobre diversos assuntos nesses últimos anos. Eu posso dizer que a minha formação, inclusive pessoal, não teria sido a mesma sem o professor.

Aos meus colegas de luta, em especial ao Brendo Peres Bizetto, Henrique Tomé Damasio e Murillo Douglas Oliveira Machado, agradeço por todas as noites em claro, finais de semana e feriados estudando para as provas, os churrascos e as risadas compartilhadas. Com certeza serão futuros excelentes profissionais. Foi uma honra compartilhar essa etapa da minha vida com vocês!

Agradeço os professores Adair Santa Catarina, André Luiz Brun, Edmar André Bellorini e Guilherme Galante pelas conversas jogadas fora durante os almoços, corredores e principalmente por todo o conhecimento passado. Vocês são excelentes professores e profissionais.

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

ACM	*Association for Computing Machinery*
API	*Application Programming Interface*
BAL	*Blockchain Access Layer)*
BlockME	*Blockchain-aware Modeling and Execution)*
BOS	*Blockchain Oriented Software*
BPEL	*Business Process Execution Language)*
BPMN	*Business ProcessModel and Notation)*
CE	Critério de Exclusão
CI	Critério de Inclusão
CPU	*Central Processing Unit*
ETH	*Ethereum*
EVM	*Ethereum Virtual Machine*
GDPR	*General Data Protection Regulation*
GUEST	*Go, Uniform, Eevaluate, Solve, Test*
HSS	*Human Social Sciences*
HTTP	*Hypertext Transfer Protocol*
ID	Identificador
IoT	*Internet of Things*
IP	*Internet Protocol*
IPFS	*InterPlanetary File System*
ITP	*Interactive Theorem Proving*
MS	Mapeamento Sistemático
P2P	*Peer-to-Peer*
PoW	*Proof-Of-Work*
RS	Revisão Sistemática
SC	*Smart Contract*
STM	*Science Technology Medicine*
WS	*WebSockets*

Sumário

Resumo

A *blockchain* é uma tecnologia de gerenciamento de dados e transações descentralizada, desenvolvida primeiro para a criptomoeda Bitcoin. O interesse na tecnologia *blockchain* tem aumentado desde que a ideia foi cunhada em 2008. O motivo do interesse na *blockchain* são seus atributos centrais que fornecem segurança, anonimato e integridade de dados sem que nenhuma organização terceirizada controle as transações e, portanto, cria áreas de pesquisa interessantes, especialmente sob a perspectiva de desafios, limitações, e processos. Neste contexto, este trabalho realiza um mapeamento sistemático com o objetivo de conhecer o atual estado da arte no que diz respeito às abordagens de desenvolvimento de software orientado a *blockchain*, bem como fornecer um guia a outros pesquisadores e profissionais a identificar possíveis áreas de pesquisa, identificar lacunas e questões ou ainda auxiliar engenheiros de software na escolha de alguma abordagem que possa ser utilizada em seu dia a dia. Os resultados mostram que a maioria dos estudos científicos ainda estão focados em melhorar a infraestrutura da *blockchain*. Extraiu-se 10 artigos principais de bancos de dados científicos e com base nestes, apresentamos análises referentes a como são construídas as abordagens para desenvolvimento do software orientado a *blockchain*, assim como análise de pontos criticos, limitações, falhas e desafios inerentes a essas abordagens, com isso foi possível verificar limitações na *blockchain* do ponto de vista de privacidade e segurança e ainda muitos outros desafios relacionados à escalabilidade da *blockchain*, incluindo taxa de transferência e latência. Com base nessas descobertas, identificamos algumas lacunas de pesquisa e futuras direções exploratórias que deverão ter um valor significativo para acadêmicos e profissionais.

Palavras-chave: Blockchain; Software Orientado a Blockchain; Blockchain Software; Engenharia de Software; Mapeamento Sistemático.

Capítulo 1

Introdução

Este Capítulo tem como objetivo apresentar a visão geral do trabalho. Na Seção 1.1 é apresentado o contexto sobre a importância da *blockchain* em transações de valores pela rede. Na Seção 1.2, são apresentadas as principais motivações para a realização do trabalho. Em seguida, na Seção 1.3 é apresentado alguns trabalhos relacionados a este e seus objetivos em particular. Na Seção 1.4 é apresentado a proposta e metodologia adotadas e é descrito as contribuições esperadas com o desenvolvimento desta pesquisa. Por fim, na Seção 1.5, é apresentado a estrutura geral e a organização do restante desta monografia.

1.1 Contexto

As transações de valores e/ou informações entre pessoas ou empresas são muitas vezes centralizadas e todos os dados e informações são controladas e gerenciados por uma organização terceirizada, em vez das duas principais entidades envolvidas na transação. Realizar um pagamento digital ou transferência requer um banco como intermediário para concluir a transação além de gerar taxas adicionais. O mesmo processo também se aplica a vários outros domínios, como jogos, música, software, etc (YLI-HUUMO et al., 2016).

A *blockchain* é uma tecnologia descentralizada de gerenciamento de transações e dados desenvolvida inicialmente para a criptomoeda Bitcoin (BOOGARD, 2018). É uma solução de banco de dados distribuída que mantém uma lista crescente de registros de dados que são confirmados pelos nós participantes. Estes dados são registrados em um registro público, incluindo informações de todas as transações já concluídas e sendo compartilhadas e disponíveis para todos os nós. Um nó pode ser qualquer dispositivo eletrônico ativo, com capacidades de pro-

cessamento e armazenamento destes registros, esteja conectado à Internet e, como tal, tenha um endereço IP. A função de um nó é suportar a rede mantendo uma cópia de um *blockchain* e, podendo também, processar transações. Os nós são frequentemente organizados em estrutura de árvores (NAKAMOTO, 2008).

1.2 Motivação

A razão para o interesse na *blockchain* são seus atributos centrais que fornecem segurança, anonimato e integridade de dados sem qualquer organização terceirizada no controle das transações (YLI-HUUMO et al., 2016).

Essa tecnologia fornece uma plataforma ideal para diversos tipos de aplicações. Por exemplo, para a execução de contratos inteligentes (ALHARBY; MOORSEL, 2017). Um contrato inteligente é aquele que é executado automaticamente quando determinadas condições são atendidas. A imutabilidade, a natureza descentralizada e os mecanismos de consenso, que são característicos da tecnologia *blockchain*, tornam o contrato inteligente e seu ciclo de desenvolvimento, um novo campo de estudo em engenharia de software (BOOGARD, 2018).

Tendo como exemplo o domínio de contrato inteligente, observa-se, na visão dos desenvolvedores de software, dificuldades para fazer a transição para a nova abordagem de desenvolvimento de contrato inteligente usando *blockchain*. Da mesma forma, há dificuldades do especialista neste domínio em conhecer as tecnologias para fazer a transição de contratos em papel para contratos inteligentes (BOOGARD, 2018).

Desenvolver software no contexto da abordagem de *blockchain* é um novo desafio para os analistas, gerente projetos, programadores e testadores de software. Olhando novamente o exemplo de domínio de contratos inteligentes, eliminar a terceira parte confiável levanta a questão de quem manterá a integridade do contrato, ou seja, como definir uma abordagem para manter a confiança entre os pares desconhecidos. Isso é ilustrado pelo Problema dos Generais Bizantinos, um experimento mental que ilustra as armadilhas e os desafios de design de tentar coordenar uma ação comunicando-se através de um elo não confiável (LAMPORT; SHOSTAK; PEASE, 1982).

1.3 Trabalhos Relacionados

Embora existam várias revisões de literatura sistemáticas sobre a tecnologia de *blockchain* (TAMA et al., 2017), (BRANDÃO; MAMEDE; GONÇALVES, 2018), argumentamos que as abordagens de desenvolvimento de software orientado a *blockchain* tem recebido pouca atenção ou nenhuma. As aplicações de *blockchains* não são cobertas em sua extensão total, aplicabilidade, tampouco em processos e abordagens para seu desenvolvimento.

Há, de fato, algumas revisões focadas no papel particular da *blockchain*, incluindo o desenvolvimento de aplicativos descentralizados e intensivos de dados para a IoT (CHRISTIDIS; DEVETSIKIOTIS, 2016) e gerenciamento de *big data* de maneira descentralizada (KARAFILOSKI; MISHEV, 2017). Zheng et al. (2016) explica detalhadamente a trajetória de rastreamento de informações compartilhada em aplicações orientadas a *blockchain* (ZHANG, 2016).

Existem ainda estudos que se concentram nas questões de segurança do *blockchain* (KHAN et al., 2017), (LI et al., 2017) e no seu potencial para permitir confiança e descentralização nos sistemas de serviços (SEEBACHER; SCHÜRITZ, 2017) e plataformas P2P (HAWLITSCHEK; NOTHEISEN; TEUBNER, 2018).

Já trabalho desenvolvido por (BOSU et al., 2018), propõem uma melhor compreensão das motivações, desafios e necessidades dos desenvolvedores de softwares orientados a *blockchains* (BOS) de forma a realizar uma análise comparativa do desenvolvimento de BOS com o desenvolvimento não-BOS e ainda uma outra análise comparativa dos resultados do estudo com duas outras pesquisas anteriores para definir seus resultados na perspectiva do campo da engenharia de software. Os autores ainda dão possíveis direções para pesquisadores criarem apoios para o desenvolvimento do BOS e uma caracterização da comunidade de desenvolvimento de BOS. Este trabalho foi localizado a partir da execução do piloto do protocolo proposto no Capítulo 3, os autores seguem a mesma linha de pensamento do trabalho proposto aqui, porém ao invés de buscar exaustivamente em base de dados científicas por artigos publicados, fizeram um *survey* em repositórios de BOS. Uma definição de BOS pode ser vista na Seção 2.3.2

Alguns aspectos técnicos do design *blockchain* como seu protocolo de consenso (SANKAR; SINDHU; SETHUMADHAVAN, 2017), as vulnerabilidades de SCs (ATZEI; BARTOLETTI; CIMOLI, 2017) e outras características técnicas como seu tamanho e largura de banda, usabilidade, integridade de dados e escalabilidade também foram estudados em (YLI-HUUMO et

al., 2016). Além disso, existem outras pesquisas, como (BONNEAU et al., 2015), (MUKHOPADHYAY et al., 2016), (KHALILOV; LEVI, 2018) e (CONTI et al., 2018), que estão mais focados no aspecto monetário das *blockchains* e na segurança e privacidade oferecidas. Contudo nenhum destes estudos focaram em mapear abordagens para desenvolvimento de softwares orientados a *blockchains* (BOS).

1.4 Objetivos e Contribuições

A fim de entender como a área de Engenharia de Software tem encarado os problemas relacionados as abordagens para o desenvolvimento de software orientado a *blockchain*, esta pesquisa visa mapear sistematicamente essas abordagens, bem como, os seus desafios atuais que precisam ser explorados em estudos futuros.

Para atingir este objetivo, selecionamos o estudo secundário, mapeamento sistemático (PETERSEN et al., 2008), como a metodologia para esta pesquisa de forma a levantar segundo os critérios de inclusão e exclusão, as abordagens referente ao desenvolvimento de BOS, identificar as pesquisas incluídas no mapeamento, categorizá-las e extrair dados de tais pesquisas.

Para tal, seguindo o processo de mapeamento sistemático apresentado por (FALBO; SOUZA; FELIZARDO, 2017), que apresenta um tutorial didático e conciso para procurar artigos relevantes em bases de dados científicas, e com isso produzir um mapa da atual pesquisa em abordagens de desenvolvimento de software orientado a *blockchain* com foco em explorar apenas estudos de um ponto de vista técnico.

Algumas razões típicas para a escolha do Mapeamento Sistemático (MS) são (FALBO; SOUZA; FELIZARDO, 2017):

- Para examinar a extensão e a natureza de uma atividade de pesquisa (ARSKEY; O'MALLEY, 2005);
- Avaliar o valor de se realizar uma Revisão sistemática (RS) completa, determinando o potencial esforço necessário para tal (ARSKEY; O'MALLEY, 2005);
- Para coletar e sumarizar a pesquisa existente em um tópico. Isso é fundamental para estudantes de pós-graduação, em especial, estudantes de doutorado iniciando seu traba-

lho, uma vez que eles devem compreender o estado da arte da pesquisa no tema de seu trabalho;

- Para identificar lacunas existentes em um tópico de pesquisa, que apontem subtópicos promissores para um projeto de pesquisa.

Mais detalhes sobre esta abordagem serão apresentados no Capítulo 3.

Se espera ao final que o mapeamento produzido ajude outros pesquisadores e profissionais a identificar possíveis áreas de pesquisa, identificar lacunas e questões, o que é fundamental para estudantes de pós-graduação, e doutorado ou ainda auxiliar engenheiros de software na escolha de alguma abordagem que possa ser utilizada em seu dia a dia.

1.5 Estrutura do trabalho

Em sequência, o trabalho está organizado da seguinte maneira: na Seção 2 apresenta-se o referencial teórico, introduzindo o conceito de processos e abordagens de desenvolvimento de software, além de uma visão geral da tecnologia *peer-to-peer*, assim como da tecnologia *blockchain*, contratos inteligentes e uma definição de BOS. Seção 3 apresenta e detalha a fase de planejamento do mapeamento sistemático. Seção 4 apresenta a condução do mapeamento, enquanto Seção 5 finaliza o trabalho.

Capítulo 2

Referencial Teórico

Neste Capítulo é apresentado o referencial teórico necessário para o entendimento do contexto deste trabalho. Na Seção 2.1 é apresentado conceitos sobre o que é uma abordagem ou processo de desenvolvimento de software. Na Seção 2.2, é introduzido o conceito de *peer-to-peer* e redes de computadores distribuídas. Na Seção 2.3, é apresentado os conceitos necessários para o entendimento da tecnologia *blockchain*, dando ênfase ao funcionamento da moeda virtual Bitcoin como exemplo de uso prático da tecnologia, na Seção 2.3.1, é apresentado os conceitos de contrato inteligente e como são realizados, na Seção 2.3.2 é feito uma explicação do que são softwares orientados a *blockchain* (BOS).

2.1 Abordagens e processos de desenvolvimento de software

Um processo ou abordagem de desenvolvimento de software pode ser entendido, como um conjunto de atividades, métodos, ferramentas e práticas relacionadas que levam à produção de um produto de software, essa prática ainda leva a produção de uma série de artefatos (SOMMERVILLE, 2011). Tais processos devem incluir, segundo Sommervile, quatro atividades fundamentais para a engenharia de software:

1. Especificação de software. A funcionalidade do software e as restrições a seu funcionamento devem ser definidas.

2. Projeto e implementação de software. O software deve ser produzido para atender às especificações

3. Validação de software. O software deve ser validado para garantir que atenda às demandas do cliente.

4. Evolução de software. O software deve evoluir para atender às necessidades de mudança dos clientes.

O autor cita ainda o que também pode ser incluído quanto as descrições de um processo:

1. Produtos, que são os resultados de uma das atividades do processo. Por exemplo, o resultado da atividade de projeto de arquitetura pode ser um modelo da arquitetura de software.

2. Papéis, que refletem as responsabilidades das pessoas envolvidas no processo. Exemplos de papéis são: gerente de projeto, gerente de configuração, programador etc.

3. Pré e pós-condições, que são declarações verdadeiras antes e depois de uma atividade do processo ou da produção de um produto. Por exemplo, antes do projeto de arquitetura ser iniciado, pode haver uma pré-condição de que todos os requisitos tenham sido aprovados pelo cliente e, após a conclusão dessa atividade, uma pós-condição poderia ser a de que os modelos UML que descrevem a arquitetura tenham sido revisados.

Assim esse trabalho busca mapear aquelas abordagens para desenvolvimento de BOS, tais que estejam conceitualmente contidas nessas definições e diretrizes.

2.2 Peer-to-peer e redes de computadores distribuídas

Uma rede cliente / servidor é uma rede distribuída que consiste em um sistema de desempenho mais alto, o servidor e vários sistemas de desempenho mais baixos, os clientes. O servidor é a unidade central de registro, bem como o único provedor de conteúdo e serviço. Um cliente só solicita conteúdo ou a execução de serviços, sem compartilhar seus próprios recursos (SCHOLLMEIER, 2001), e caso o servidor seja removido existe uma perda por completo do serviço.

Como definido por (SCHOLLMEIER, 2001) uma arquitetura de rede distribuída pode ser chamada de rede ponto a ponto (*Peer-to-peer*), se os participantes compartilham uma parte de

seus próprios recursos de hardware (capacidade de processamento, capacidade de armazenamento, capacidade de link de rede, impressoras, etc). Esses recursos compartilhados são necessários para fornecer o serviço e o conteúdo oferecido pela rede (por exemplo, compartilhamento de arquivos ou espaços de trabalho compartilhados para colaboração). Eles são acessíveis por outros pares diretamente, sem passar por entidades intermediárias. Os participantes de tal rede são, portanto, provedores de recursos (Serviço e conteúdo), bem como solicitantes de recursos (Serviço e conteúdo). E pode ser classificada como uma rede ponto a ponto "Pura", se for primeiramente uma rede ponto a ponto totalmente de acordo com a definição previamente citada e, em segundo lugar, se qualquer entidade escolhida arbitrariamente puder ser removida da rede, sem que a rede sofra perda de serviço. Um esquema de comunicações em rede cliente/servidor (a esquerda) e *peer-to-peer* (a direita) pode ser vista na Figura 2.1.

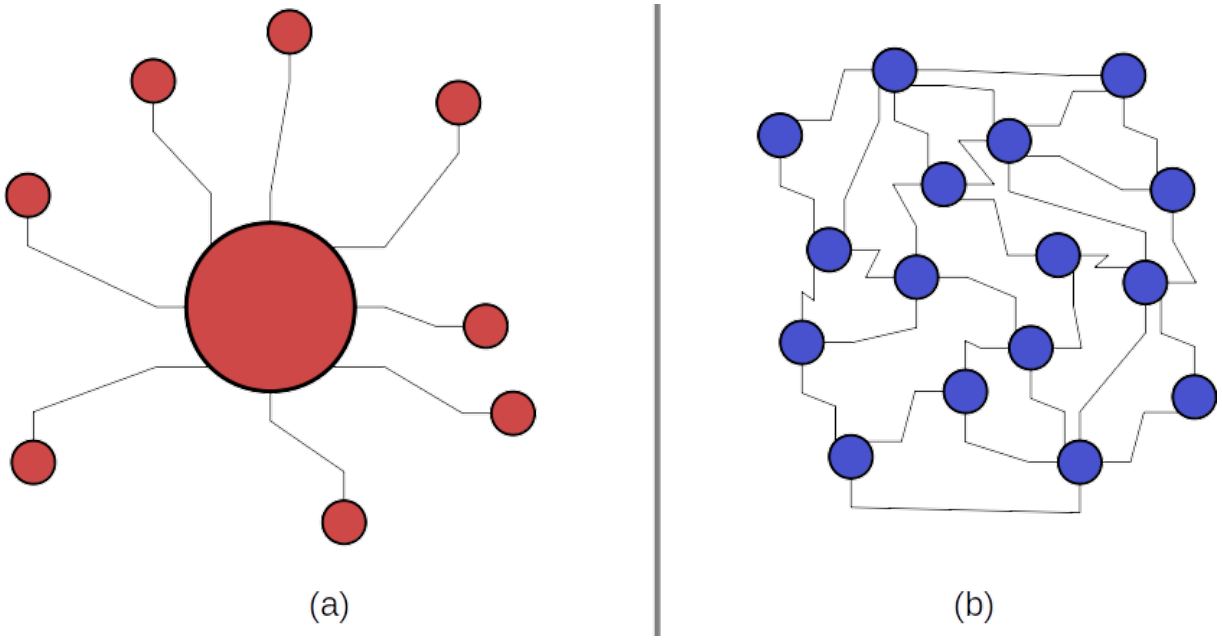

Figura 2.1: Comparação rede cliente-servidor (a) e peer-to-peer (b)

2.3 Introdução à tecnologia blockchain

O nome Blockchain foi originalmente dado ao design que sustenta o funcionamento da moeda digital Bitcoin, ela é uma tecnologia descentralizada de gerenciamento de transações e dados. O criador do Bitcoin nunca usou o termo *blockchain* em seu artigo, e lendo a publicação obtém-se a nítida impressão de que o autor não estava introduzindo uma nova tecnologia

no sentido tradicional do termo, mas um design de software baseado em várias tecnologias existentes para permitir a ele criar uma "versão puramente *peer-to-peer* de dinheiro eletrônico" (NAKAMOTO, 2008).

Como já dito no Capítulo 1, *blockchain* é uma solução de banco de dados distribuída que mantém uma lista crescente de registros de dados que são confirmados pelos nós participantes (ver Figura 2.2). Estes dados são registrados em um registro público, incluindo informações de todas as transações já concluídas e sendo compartilhadas e disponíveis para todos os nós. Um nó pode ser qualquer dispositivo eletrônico ativo, com capacidades de processamento e armazenamento destes registros, esteja conectado à Internet e, como tal, tenha um endereço IP. A função de um nó é suportar a rede mantendo uma cópia de um *blockchain* e, podendo também, processar transações. Os nós são frequentemente organizados em estrutura de árvores (NAKAMOTO, 2008).

A essência da operação *blockchain* do Bitcoin é que sempre que dois membros da rede realizam transações, eles anunciam sua transação para todos os membros da rede, os nós, que por sua vez registram a transação em um bloco com capacidade limitada. Quando o bloco está cheio, os nós executam simultaneamente a *Proof-of-Work* (PoW), operações matemáticas difíceis de resolver, mas cuja solução correta é fácil de verificar. Essas operações matemáticas não estão relacionadas às transações de *blockchain*, mas são indispensáveis para a operação do sistema, pois forçam os nós de verificação a gastar poder de processamento que seria desperdiçado se incluíssem quaisquer transações fraudulentas ou inválidas.

O primeiro nó que consegue resolver um problema de PoW transmite a solução, juntamente com o bloco de transações, para todos os outros nós. Os nós podem verificar de forma rápida e econômica a precisão das transações e soluções, e quando 51% do poder de processamento da rede vota para aprovar um bloco, os nós começam a registrar novas transações em um novo bloco, corrigindo para todos os blocos anteriores. O primeiro nó que resolve o problema da *Proof-of-Work* é recompensado com uma quantidade específica da moeda da rede. Essa recompensa torna a verificação de transações potencialmente lucrativa e faz com que ela seja comumente chamada de "mineração", embora "verificar" seja uma descrição mais funcionalmente precisa. Funcionalmente, a tecnologia Blockchain é uma tecnologia de verificação: como

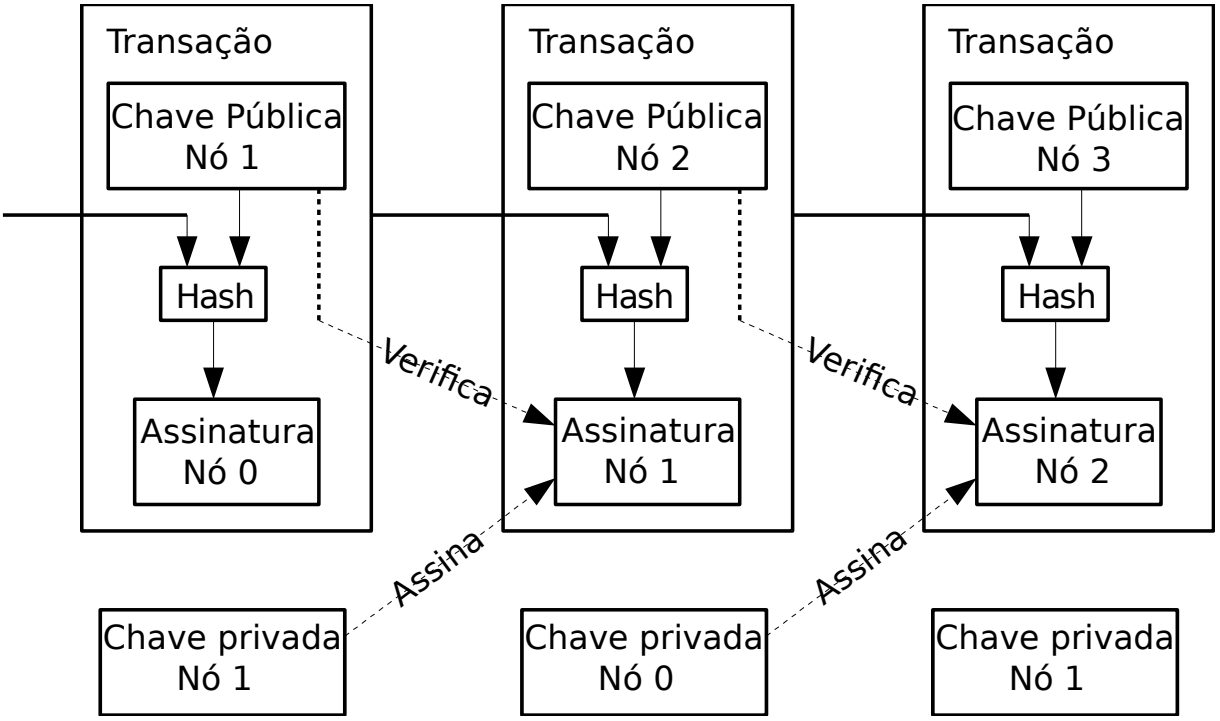

Figura 2.2: A essência da operação *blockchain* do Bitcoin. Traduzido de (NAKAMOTO, 2008)

é muito mais dispendioso resolver a PoW do que verificar sua exatidão, a honestidade é a única estratégia de lucratividade, e o resultado é um registro que é indiscutível (AMMOUS, 2016).

2.3.1 Aplicação de Blockchain: Contratos inteligentes

Na literatura, uma aplicação de blockchain que se destaca são os contratos inteligentes. Atualmente, os contratos são elaborados por advogados, julgados pelos tribunais e executados pela polícia (AMMOUS, 2016). A primeira vez que o termo *Smart Contract* apareceu foi no artigo *"The idea of smart contracts"* do jurista e criptógrafo Nick Szabo (SZABO, 1997). Foi a primeira tentativa para estabelecer a ideia dos contratos em software. O principal objetivo de um contrato inteligente é executar automaticamente os termos de um contrato, uma vez cumpridas as condições especificadas.

Os contratos inteligentes prometem baixas taxas de transação em comparação aos sistemas tradicionais que exigem que um terceiro, confiável por ambas as partes interessadas, aplique e execute os termos de um contrato. No entanto, essa ideia emerge com o surgimento da tecnologia *blockchain*. Um contrato inteligente pode ser considerado como um sistema que libera ativos digitais para todas ou algumas das partes envolvidas, uma vez que regras pré-definidas arbitrá-

rias tenham sido atendidas, um esquema de alto nível pode ser visto na Figura 2.3 (ALHARBY; MOORSEL, 2017).

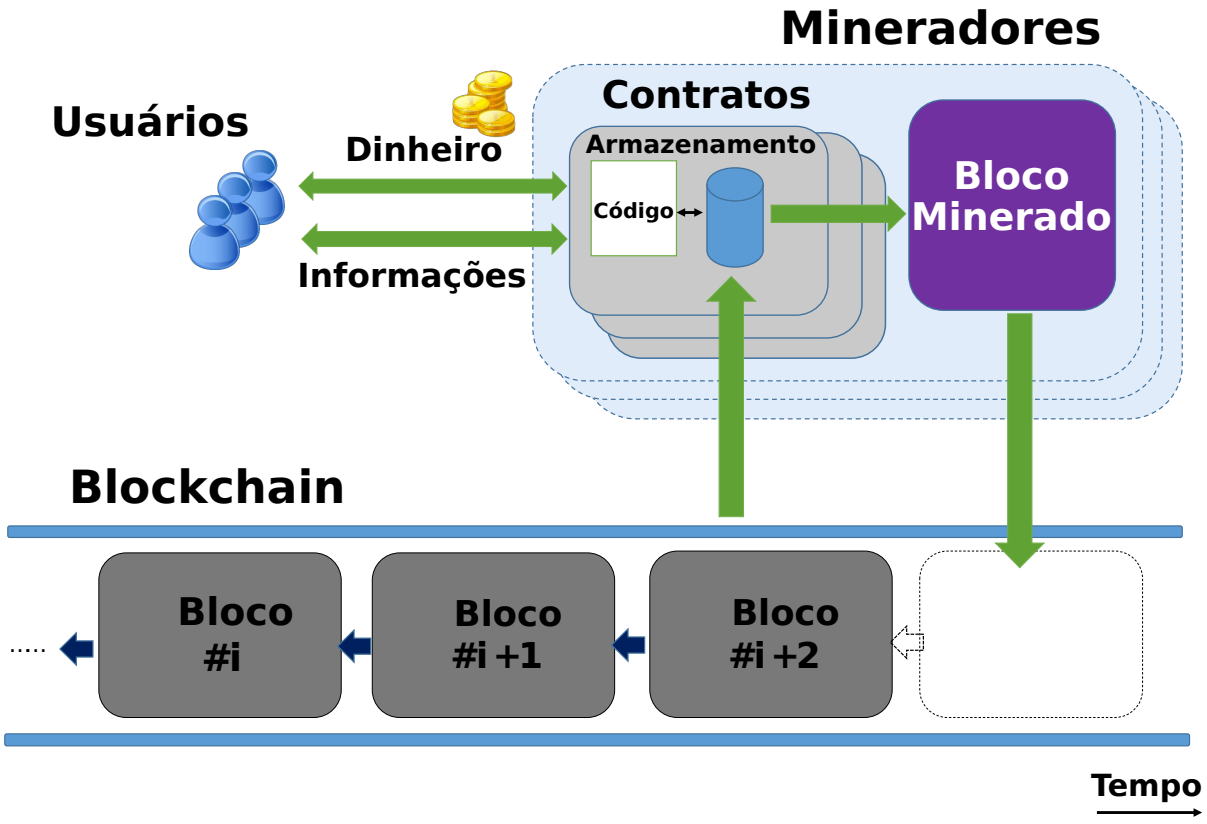

Figura 2.3: Sistema de um Contrato Inteligente. Traduzido de (DELMOLINO et al., 2016)

A Figura 2.3 apresenta o esquema de um sistema de criptomoedas descentralizado com contratos inteligentes. O estado de um contrato inteligente é armazenado no *blockchain* público. Um programa de contrato inteligente é executado por uma rede de mineradores que atingem o consenso sobre o resultado da execução e atualizam o estado do contrato no *blockchain* de acordo. Os usuários podem enviar dinheiro ou dados para um contrato; ou receber dinheiro ou dados de um contrato (DELMOLINO et al., 2016).

Hoje, a ideia de Nick Szabo se tornou realidade a partir da plataforma Ethereum. Sistemas inteligentes de criptografia de contrato, como o Ethereum, codificam contratos em uma *blockchain* para torná-los auto-executáveis, sem possibilidade de recurso ou reversão, além do alcance dos tribunais e da polícia. "O código é a lei" é um lema usado por programadores de contratos inteligentes (AMMOUS, 2016). A plataforma Ethereum, Segundo o Relatório da Bloomberg "*Blockchain Technology and Legal Implications*" está: "desenvolvendo uma infraestru-

tura em *blockchain* em cima do qual aplicações *peer-to-peer* e contratos inteligentes podem ser construídos" (LEE et al., 2015). Desse modo, os contratos inteligentes operam via plataforma Blockchain e dispensam a atuação de terceiros que o validem. Essa concepção é inovadora quando colocado ao lado das noções mais primitivas de contrato (WRIGHT; FILIPPI, 2015).

2.3.2 Softwares orientados a blockchain

Como definido por (PORRU et al., 2017) um software orientado a *blockchain* (BOS) é todo o software trabalhando com alguma implementação de um *blockchain*. Essa definição inclui plataformas *blockchain*, como Bitcoin e Ethereum, e software *blockchain* em geral. Muitos dos autores classifica as aplicações de *blockchain* em aplicações financeiras como o Bitcoin e não financeiras como Medicalchain, uma plataforma de prontuários médicos que se utiliza de uma *blockchain* para assegurar a legitimidade e segurança das informações dos pacientes (CROSBY et al., 2016), uma vez que as criptomoedas representam uma porcentagem considerável das redes de *blockchain* existentes. Outros os classificam de acordo com as versões *blockchain* (ou seja, 1.0, 2.0 e 3.0) (ZHAO; FAN; YAN, 2016).

Capítulo 3

Planejamento do Mapeamento Sistemático

Revisões sistemáticas (RS) e mapeamentos sistemáticos (MS) são tipos de estudos secundários, aqueles que revisam todos (ou quase todos) os estudos primários relacionados a uma questão de pesquisa específica, contribuindo para integrar/sintetizar evidências sobre a questão de pesquisa, diferentemente de estudos primários, que investigam uma questão de pesquisa específica, assim estudos secundários seguem um processo de pesquisa metodologicamente bem definido para identificar, analisar e interpretar as evidências disponíveis relacionadas com um tópico de pesquisa ou fenômeno de interesse, de uma maneira não tendenciosa e até mesmo repetível (KITCHENHAM; CHARTERS, 2007).

Kitchenham e Chartes cria esse conceito a partir de um paralelo feito com a Medicina Baseada em Evidências. Ela mesmo define que o objetivo da Engenharia de Software Baseada em Evidências é prover meios que possibilitem a integração entre as melhores evidências atuais, obtidas através de pesquisas, com a experiência prática e valores humanos no processo decisório relacionado ao desenvolvimento e manutenção de software (KITCHENHAM; CHARTERS, 2007).

Kitchenham e Chartes descreve o MS como um método projetado para fornecer uma ampla visão de uma determinada área de pesquisa, que permite identificar, quantificar e analisar os resultados, estabelecendo evidências sobre um determinado tema. Os autores ainda apontam algumas diferenças entre o MS e a RS, em pontos específicos, como por exemplo sobre as questões de pesquisa, termos usados, extração de dados, etc (KITCHENHAM; CHARTERS, 2007). A Tabela 3.1 sumariza as principais diferenças entre MSs e RSs.

De maneira geral, o objetivo de um estudo secundário é prover aos pesquisadores uma visão geral de uma área de pesquisa e ajudar a identificar lacunas nessa área (WOHLIN et al.,

Tabela 3.1: Diferenças entre MS e RS

Elementos	**Mapeamento sistemático**	**Revisão sistemática**
Objetivos	Prover uma visão ampla de um tópico de pesquisa	Prover uma discussão detalhada a partir da sumarização de evidências dos estudos primários
Questões de pesquisa	Genéricas	Específicas
Processo de busca	Definido pelo tópico de pesquisa e/ou pelas questões de pesquisa	Definido pelas questões de pesquisa
Escopo	Amplo	Focado
Estratégia de busca	*String* de busca mais genérica	*String* de busca mais específica
Avaliação da qualidade	Não é obrigatória	Recomendável
Resultados	Categorização dos estudos primários de acordo com esquemas de classificação	Discussão detalhada sobre as evidências referentes ao tópico de pesquisa

2013). Mapeamentos Sistemáticos estão focados na estruturação de uma área de pesquisa (PETERSEN; VAKKALANKA; KUZNIARZ, 2015) e objetivam classificar os estudos primários relevantes em categorias bem definidas (KITCHENHAM et al., 2011).

A realização de um mapeamento sistemático envolve três fases principais, que são executadas de forma iterativa (ver Figura 3.1) sendo elas: 1) Fase de planejamento, 2) Fase de condução e 3) Fase de publicação dos resultados (FELIZARDO et al., 2017).

Neste Capítulo será descrito a fase de planejamento do MS. Cabe destacar que o processo de um MS não é puramente sequencial. Muitas atividades são iniciadas durante a fase de planejamento e que são refinadas quando a fase de execução efetivamente ocorre, daí a definição dele ser iterativo.

A fase de planejamento do MS é onde os objetivos da pesquisa são listados e as questões de pesquisa são formuladas. Uma vez que o propósito de um MS é revisar um tópico mais amplo de pesquisa e classificar os estudos primários relacionados com este, as questões de pesquisa para um MS são mais gerais (FALBO; SOUZA; FELIZARDO, 2017). E a partir dessas questões de pesquisa que é definida a *string* de busca. Tal *string* será utilizada nas bases de dados que serão definidas no planejamento. Vale destacar que as questões de pesquisa devem ser definidas com cuidado pois implicará diretamente na qualidade dos resultados retornados pela *string* de

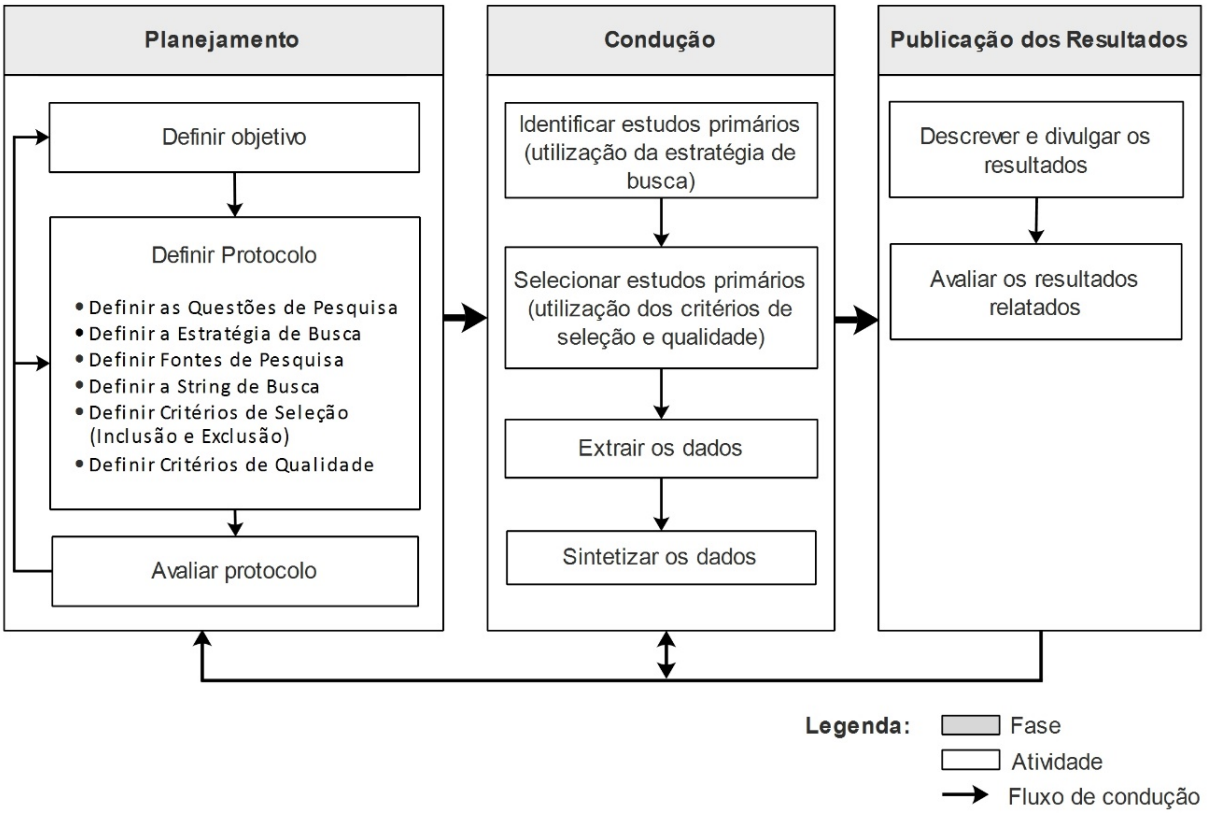

Figura 3.1: Fases e atividades do processo de MS (FELIZARDO et al., 2017)

busca. Assim para o sucesso de um MS, é vital que as questões de pesquisa sejam elaboradas corretamente (FELIZARDO et al., 2017).

3.1 Definição das Questões de Pesquisa

O objetivo aqui é descrever as questões referentes ao tópico de pesquisa abordado, que devem ser respondidas (FELIZARDO et al., 2017). Kitchenham e Chartes propõe que o primeiro passo para definir as questões de pesquisa é descrever o problema, ou seja, a razão pelo qual o MS está sendo feito (KITCHENHAM; CHARTERS, 2007). A descrição do problema abordado por este trabalho já foi conceituado no Capítulo 1. Uma síntese da descrição do problema abordado por este trabalho é a questão expressa no título deste trabalho de conclusão de curso: Qual o estado da arte das abordagens de desenvolvimento de software orientado a *blockchain*?

A partir deste problema, as questões de pesquisa para o protocolo deste mapeamento (este Capítulo) foram elaboradas considerando elementos críticos a serem definidos, pois as questões de pesquisa influenciam os processos de busca, extração e de análise de dados. Analisando a

motivação e a descrição do problema definidas anteriormente, foram especificadas cinco questões de pesquisa:

QP1: Como tem sido construídas as abordagens de desenvolvimento de BOS?

QP2: Quais os pontos críticos e/ou desafios dessas abordagens?

QP3: Como estas abordagens foram validadas?

QP4: Quais os problemas têm sido apontados pelas organizações de software que adotam as abordagens para desenvolvimento de BOS?

QP5: Quais são as tecnologias usadas para prover gestão e desenvolvimento das abordagens para BOS?

As questões de pesquisa QP1 e QP2 tem como objetivo conhecer o estado da arte no que diz respeito à abordagens para o desenvolvimento de BOS. Ou seja, elas são direcionadas para entender como cada abordagem encontrada descreve o processo de criação de BOS e quais são pontos críticos e/ou falhas que podem ser um limitador do trabalho encontrado.

A questão de pesquisa QP3 foi elaborada pelo fato do autor desta pesquisa considerar importante que todo o trabalho científico deve possuir uma estratégia experimental que valide suas teorias. Existem inúmeros tipos de classificação de experimentos, mas cabe destacar três estratégias experimentais: *Survey*, Estudo de Caso e Experimento (SHULL; SINGER; SJØBERG, 2007).

A questão de pesquisa QP4 objetiva levantar quais os problemas mais relatados pelas organizações que adotam as abordagens para desenvolvimento de BOS, de forma a obter uma visão prática de possíveis obstáculos na implementação de BOS.

Por fim, uma ferramenta computacional é uma boa prática para auxiliar a gestão de software. Assim, a questão QP5 é elaborada a fim de verificar se existem tecnologias desenvolvidas especialmente para este fim, ou se ajustes foram feitos em tecnologias já existentes para um melhor aproveitamento em abordagens para desenvolvimento de BOS.

3.2 Definindo a string de busca

Após a definição das questões de pesquisa, o passo seguinte é criar a *string* de busca. Kitchenham e Chartes indicam a utilização da técnica PICO (Population, Intervention, Comparison e Outcome). Técnica que foi desenvolvida originalmente para ser utilizada em RSs na área da Medicina e posteriormente adaptada para a área de Engenharia de Software, ela auxilia os pesquisadores a identificar palavras-chave que serão utilizadas na *string* de busca (KITCHENHAM; CHARTERS, 2007).

As diretrizes médicas recomendam considerar uma pergunta sobre a eficácia de um tratamento sob três pontos de vista: A população, ou seja, as pessoas afetadas pela intervenção. As intervenções, que geralmente são uma comparação entre dois ou mais tratamentos alternativos. Os resultados, isto é, os fatores clínicos e econômicos que serão usados para comparar as intervenções (KITCHENHAM; CHARTERS, 2007).

Em experimentos de engenharia de software, as populações podem ser uma das seguintes: Uma função específica de engenharia de software, por exemplo testadores e/ou gerentes. Uma área de aplicação, por exemplo Sistemas de TI, sistemas de comando e controle. Uma abordagem ou processo como por exemplo BPMN, Modelo Cascata (KITCHENHAM; CHARTERS, 2007).

A intervenção é a metodologia, ferramenta, tecnologia ou procedimento de software que aborda um problema específico, por exemplo, tecnologias para executar tarefas específicas, como especificação de requisitos, teste do sistema ou estimativa de custos de software (KITCHENHAM; CHARTERS, 2007).

A comparação é a metodologia, ferramenta, tecnologia ou procedimento de engenharia de software com a qual a intervenção está sendo comparada. Quando a tecnologia de comparação é a tecnologia convencional ou comumente usada, é frequentemente chamada de tratamento de "controle". A situação de controle deve ser adequadamente descrita. Em particular, “não usar a intervenção” é inadequado como uma descrição do tratamento de controle. As técnicas de engenharia de software geralmente requerem treinamento. Se você comparar pessoas que usam uma técnica com pessoas que não usam uma técnica, o efeito da técnica será confundido com o efeito do treinamento. Ou seja, qualquer efeito pode ser devido ao treinamento e não à técnica espe-

cífica. Este é um problema específico se os participantes forem estudantes (KITCHENHAM; CHARTERS, 2007).

Os resultados devem estar relacionados aos fatores de importância para os profissionais, como confiabilidade aprimorada, custos de produção reduzidos e tempo reduzido de comercialização. Todos os resultados relevantes devem ser especificados. Por exemplo, em alguns casos, exigimos intervenções que melhorem algum aspecto da produção de software sem afetar outro, por exemplo, confiabilidade aprimorada sem aumento de custo. Um problema específico para experimentos de engenharia de software é o uso generalizado de medidas substitutas, por exemplo, defeitos encontrados durante o teste do sistema como substituto da qualidade ou medidas de acoplamento para a qualidade do projeto. Os estudos que usam medidas substitutas podem ser enganosos e as conclusões baseadas nesses estudos podem ser menos robustas (KITCHENHAM; CHARTERS, 2007).

Analisando a descrição do problema e as questões de pesquisa levantadas, nós definimos a estrutura PICO como:

- População: Projetos (experimentais, estudo de casos reais, etc.) que adotam, definam e apliquem as abordagens para desenvolvimento de BOS.
- Intervenção: Adoção (criação, desenvolvimento, avaliação, aplicação) das abordagens de desenvolvimento de BOS.
- Comparação: Não foi considerado no contexto desta pesquisa, por se tratar de um MS e não uma RS.
- Resultado: Benefícios, tecnologias utilizadas, desafios, limitações com a intervenção mencionada.

Sabendo-se que as bases de dado eletrônicas que serão consultadas (Seção 3.3) utilizam a língua inglesa, se faz necessário a tradução de todas as palavras-chave encontradas pela técnica PICO, pois a *string* de busca será em inglês. Cabe ainda destacar que, os sinônimos de cada palavra chave podem ser adicionados na *string* de busca assim como reduções da palavra chave. A Tabela 3.2 mostra toda as palavras-chave e os seus sinônimos e/ou reduções e onde eles estão encaixados na estrutura PICO. Os sinônimos e/ou reduções sempre estarão conectados através do operador lógico “OU” (OR).

Tabela 3.2: Palavras chaves, sinônimos e/ou reduções

ID	Palavra Chave e Sinônimos/Reduções	Estrutura PICO
1	*blockchain OR "blockchain oriented software" OR "blockchain based software"*	*Population*
2	*"peer-to-peer" OR "decentralized network" OR "distributed ledger"*	*Intervention*
3	*approach OR pattern OR method OR model OR framework OR technique OR api OR process OR technology*	*Outcome*

A *string* de busca de um MS é formada por palavras-chave e por operadores lógicos "E" (AND) e "OU" (OR). Quando se utiliza a técnica PICO, a estrutura da *string* de busca fica no seguinte formato: Population AND Intervention AND Comparison AND Outcome, sendo que em cada termo da técnica PICO, as palavras-chave são unidas com o operador lógico "OR". Assim, a string de busca geral do MS é definida por: (#1) AND (#2) AND (#3). A Figura 3.2 apresenta a *string* geral de busca deste MS.

3.3 Seleção das fontes de busca

Um fator importante durante a condução de um MS além de definir bem as questões de pesquisa e a *string* de busca é a escolha das fontes de busca, a escolha irá afetar a cobertura de estudos relevantes do tema pesquisado, assim as fontes devem ser escolhidas de forma minuciosa (FELIZARDO et al., 2017). Então foram definidas três bases de dados eletrônicas a serem utilizadas neste trabalho, sendo elas:

- ACM Digital Library - <http://dl.acm.org/>
- Scopus - <http://www.scopus.com/>
- SpringerLink - <http://link.springer.com/>

Essas bases foram escolhidas levando em consideração a facilidade de uso de seus motores de busca, capacidade de filtrar resultados, reconhecimento no meio acadêmico, e seus motores de busca suportarem todos os conectores lógicos definidos na *string* de busca (ver Seção 3.2).

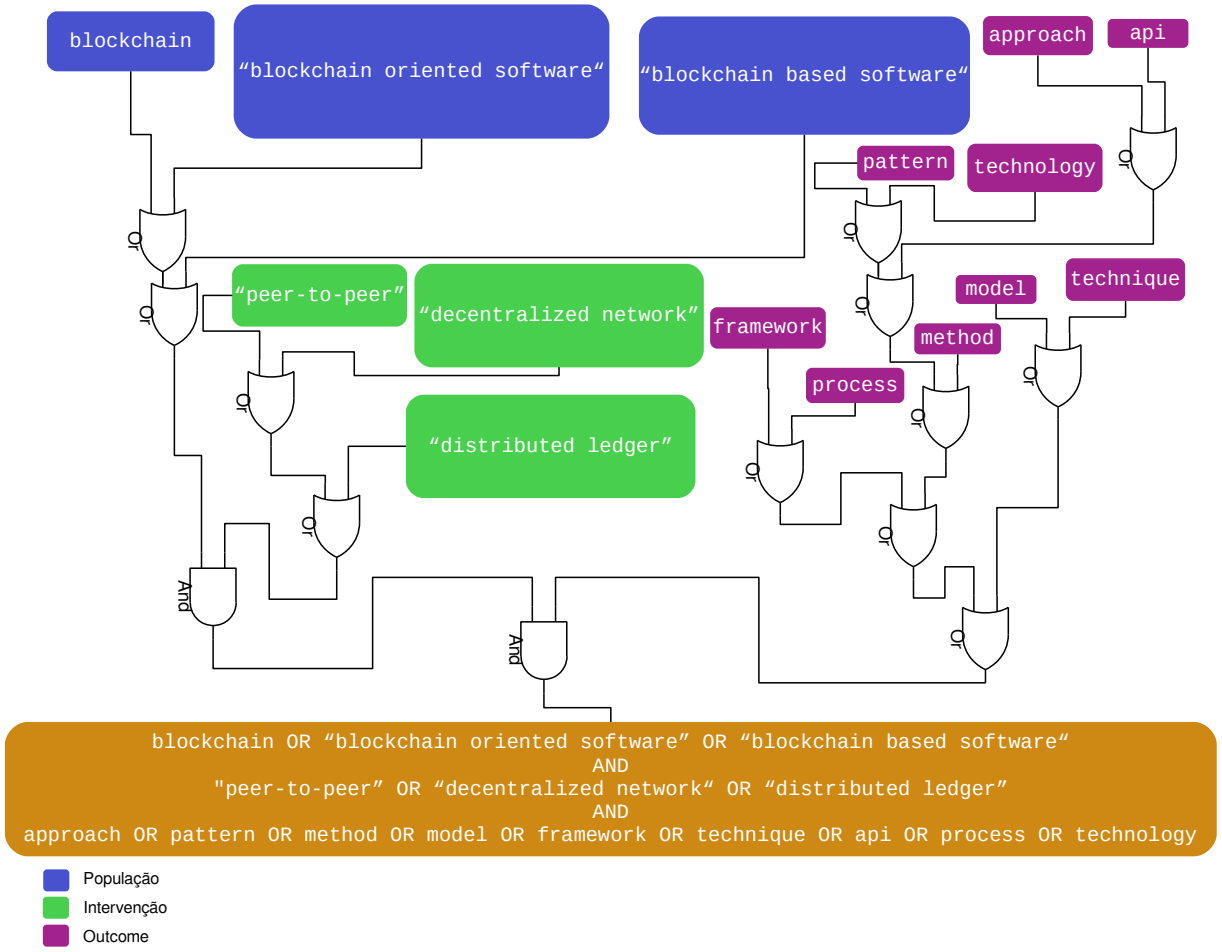

Figura 3.2: *String* geral de busca

3.4 Definição dos estudos

As bases de dados eletrônicas da atualidade ainda possuem problemas como estabilidade de resultados e falta de consistência de nomenclatura das palavras-chave, como dito por (BRERETON et al., 2007) “os motores de busca atuais ainda não estão apropriados para serem utilizados”, no caso, em RS, mas também se aplica a MS. Dessa forma é importante a criação de critérios de inclusão e exclusão, para que se evite estes problemas e ainda outros como estudos duplicados. Nas seguinte Seção são descritos tais critérios de inclusão e exclusão de estudos. Em seguida, na Seção 3.4.2 é apresentada a estratégia adotada para seleção dos estudos. Por fim, na Seção 3.4.3 é apresentada a ferramenta StArt, que será utilizada para facilitar o manuseio e controle dos artigos buscados (LAPES, 2019).

3.4.1 Critérios de Inclusão e Exclusão de Estudos

Os critérios de inclusão e exclusão devem ser baseados nas questões de pesquisa definidas no começo do protocolo (FELIZARDO et al., 2017). Os critérios de inclusão deste MS são definidos como:

1. Artigos científicos publicados em eventos, revistas ou livros que estejam indexados nas bases de dados;
2. Estudos que abordam sobre técnicas, métodos, algoritmos, diretrizes e/ou processos para criação de BOS;
3. Estudos publicados a partir de 2008 até 2019;
4. Estudos escritos em inglês ou português.
5. Estudos disponíveis gratuitamente à todos.
6. Estudos que abordam aspectos de engenharia de software.

O primeiro critério de inclusão é para garantir que estes estudos tenham sido revisados por especialistas da área, de forma que os trabalhos possuam uma qualidade mínima aceitável. O segundo critério de inclusão garantirá que os estudos abordem sobre o contexto deste trabalho. O terceiro critério é utilizado considerando que a ideia inicial de *blockchain* surgiu em meados de 2008 por (NAKAMOTO, 2008) e que este mapeamento está sendo conduzido em 2019. Alguns estudos retornados dessas bases de dados escolhidas podem estar escritos em outros idiomas, assim o quarto critério garantirá apenas aqueles escritos em inglês ou português. Por fim, o sexto critério garantirá que os estudos abordem sobre a perspectiva da engenharia de software, destacando as possíveis contribuições para esta área.

Os critérios de exclusão deste MS são definidos como:

1. Estudos duplicados.
2. Artigos que não são científicos ou não foram publicados em eventos, revistas ou livros que estejam indexados nas bases de dados;

3. Estudos que não abordam sobre técnicas, métodos, algoritmos, diretrizes e/ou processos para criação de BOS;

4. Estudos publicados antes de 2008 ou depois 2019;

5. Estudos que não foram escritos em inglês ou português.

6. Estudos indisponíveis gratuitamente à todos.

7. Estudos que não abordam do ponto de vista de um engenheiro de software.

Todos os critérios de exclusão são o complemento dos critérios de inclusão, com exceção do primeiro critério de exclusão. Este critério garante que não haverá estudos duplicados.

3.4.2 Procedimento de Seleção, Classificação e Extração de Dados

Após a definição dos critérios de inclusão e exclusão de estudos, uma estratégia de seleção dos estudos deve ser criada para aplicar esses critérios. Para que um estudo seja incluído, ele deve satisfazer todos os critérios de inclusão. Por outro lado, basta que apenas um dos critérios de exclusão seja satisfeito para que o estudo não seja considerado para a fase de extração dos dados e excluído do mapeamento. O processo de seleção dos estudos consistem em manter uma lista de estudos excluídos bem como a razão deles não serem considerados, e outra com os estudos selecionados, este processo é iterativo e será baseado na ideia de (PETERSEN et al., 2008). O processo de seleção será dividido em quatro fases:

1ª Todos os estudos retirados das bases eletrônicas serão importados para a ferramenta StArt (LAPES, 2019). Essa ferramenta é descrita com mais detalhes na Seção 3.4.3. Para cada estudo, será criado um arquivo BibTex, que será importado na ferramenta, e ainda auxiliará na remoção de estudos duplicados. Note que não é garantido que a ferramenta remova todos os estudos duplicados.

2ª Leitura e análise de palavras-chave e dos títulos. Se um estudo atender a pelo menos um critério de exclusão, o mesmo será excluído das próximas fases.

3ª Leitura e análise dos *abstracts* (resumos) dos estudos que não foram excluídos até então. Novamente, se um estudo atender a pelo menos um critério de exclusão, o mesmo será excluído da próxima fase.

4ª Leitura e análise das seções de introdução e conclusão e caso exista dúvida sobre aplicação de um dos critérios de exclusão, leitura e análise dos estudos por completo. Os estudos que serão incluídos para a análise na terceira fase do MS devem atender todos os critérios de inclusão.

Um formulário de extração de dados dos artigos (ver Figura 3.3), projetado para coletar as informações necessárias para atender os objetivos deste estudo, após a quarta fase do processo de seleção, documentará as respostas para cada questão de pesquisa apresentada na Seção 3.1 de cada estudo, os objetivos, problemas, resultados, e informações necessárias para categorizar o estudo usando as facetas, que é a divisão das questões de pesquisa em categorias e essas foram definidas na Tabela 3.3. Durante a extração, alguns estudos podem ser classificados em mais de uma categoria. Assim, estudos primários podem aparecer mais de uma vez.

As diferentes facetas e categorias para classificar (mapear) os estudos primários estão descritos na Tabela 3.3.

Tabela 3.3: Facetas e exemplos de categorias

Faceta	**Categoria**
Tipo do estudo	Pesquisa de Validação, Pesquisa de Avaliação, Proposta de Solução, Documentos Filosóficos, Artigos de Opinião e Documentos de Experiência (SILVA et al., 2011), Exemplo (Toy example), Simulação Experimental.
Tipo de contribuição	Processo, Método, Modelo, Abordagem, *Framework*, Métrica,(SILVA et al., 2011) API, *Guidelines* e Plataforma
Limitações na Arquitetura de Blockchain	Mecanismo de consenso, Taxa de transferência, Latência, Tamanho e largura de banda, Recursos desperdiçados (YLI-HUUMO et al., 2016), Privacidade, Sustentabilidade, Segurança, Performance e Eficiência
Tipo de aplicação	Saúde, Criptomoeadas, *Smart contract* (CASINO; DASAKLIS; PATSAKIS, 2018), Open systems
Tecnologias ou Ferramentas	Ethereum, Litecoin, Bitcoin, Multichain, Hyper
Classificação e principais características das redes *blockchain*	Ver Tabela 3.4 (CASINO; DASAKLIS; PATSAKIS, 2018)
Finalidade da abordagem	Desenvolvimento de *blockchain*, Avaliação da adequabilidade do *blockchain* (CASINO; DASAKLIS; PATSAKIS, 2018), Sistemas distribuídos orientados a Blockchain

ID		**Base de dados**	
Título			
Autores			
Palavras-chave			
Resumo			
Questões de pesquisa			
QP1: Como tem sido construídas as abordagens de desenvolvimento de BOS?			
QP2: Quais os pontos críticos e/ou desafios dessas abordagens?			
QP3: Como estas abordagens foram validadas?			
QP4: Quais os problemas têm sido apontados pelas organizações de software que adotam as abordagens para desenvolvimento de BOS?			
QP5: Quais são as tecnologias usadas para prover gestão e desenvolvimento das abordagens para BOS?			
Facetas			
Tipo do estudo			
Tipo de contribuição			
Limitações na Arquitetura de Blockchain			
Tipo de aplicação			
Tecnologias/Ferramentas			
Classificação e principais características das redes blockchain			
Finalidade da abordagem			

Figura 3.3: Formulário de extração de dados

3.4.3 A Ferramenta StArt (*State of the Art through Systematic Review*)

A StArt (*State of the Art through Systematic Review*) é uma ferramenta de apoio ao processo de revisão sistemática desenvolvida e mantida pelo Laboratório de Pesquisa em Engenharia de Software da Universidade Federal de São Carlos. A ferramenta disponibiliza o preenchimento do protocolo para a fase de planejamento, funcionalidades de apoio a condução, seleção e extração de informações para a fase de execução/condução, disponibiliza ainda, gráficos com dados estatísticos para a analise dos dados e permite que o usuário elabore diversos tipos de relatórios, podendo a qualquer momento acessar as informações extraídas de cada estudo. Exemplos desses relatórios são: geração de todas as informações pertinentes à RS, correspondendo ao empacotamento da RS; resumos de todos os estudos importados por meio do arquivo BibTex;

Tabela 3.4: Classificação e principais características das redes *blockchain* (CASINO; DASAKLIS; PATSAKIS, 2018)

Propriedade	**Publica**	**Privada**	**Federada**
Consenso	PoW custosa	PoW leve	PoW leve
Mecanismo	Feito por mineradores	Organização centralizada	Conjunto de nós líder
Identidade	Pseudo anonimato	Usuários identificados	Usuários identificados
Anonimato	Malicioso?	Confiável	Confiável
Eficiência do protocolo	Baixa eficiência	Alta eficiência	Alta eficiência
Consumo	Alto consumo de energia	Baixo consumo de energia	Baixo consumo de energia
Imutabilidade	Quase impossível	*Collision attack* (Ataque de colisão de *hash*)	*Collision attack* (Ataque de colisão de *hash*)
Propriedade	Pública	Centralizada	Semi-centralizada
Gestão	Sem permissão	Lista de permissões permitida	Nós autorizados
Aprovação de transação	Ordem de minutos	Ordem de milissegundos	Ordem de milissegundos

conjunto das informações extraídas pelo usuário de cada um dos estudos aceitos na etapa de Extração de Dados; arquivo no formato BibTex contendo todos os estudos pertencentes à RS possibilitando que eles sejam importados em um gerenciador de referências (LAPES, 2019). A Figura 3.4 mostra a interface da ferramenta.

3.5 Considerações Finais do Capítulo

Neste Capítulo foi apresentado toda a fase de planejamento do protocola que este MS irá executar. Foi mostrada uma breve comparação entre RSs e MSs, e então a descrição do protocolo, começando com a definição das questões de pesquisa, onde suas respostas devem suprir o problema apresentado no Capítulo 1, logo após a definição da *string* de busca, que será utilizada para buscar os estudos nas bases de dados eletrônicas, sendo essas definidas logo em seguida. Então foi descrito como os estudos serão selecionados, definindo critérios de inclusão e exclu-

são, e todo o precedimento que irá decorrer até a fase de extração dos dados daqueles estudos aptos a para tal, e ao final uma breve introdução a ferramenta StArt, que irá auxiliar nestes MS.

Figura 3.4: Interface da ferramenta StArt

Capítulo 4

Condução do Mapeamento

Finalizado o planejamento do MS, o próximo passo é a condução (ver Figura 3.1). O objetivo desta fase é seguir todo o planejamento definido no Capítulo 3, de modo que, estudos primários que respondam as questões de pesquisa definidas e atendam aos critérios de inclusão sejam encontrados.

As 4 fases da condução do MS é apresentada neste Capítulo. A busca e disposição dos estudos encontrados nas bases de dados eletrônicas consiste na primeira fase, as bases que serão utilizadas são aquelas definidas na Seção 3.3. Essa fase é apresentada na Seção 4.1. Continuando, na Seção 4.2 temos a etapa referente ao processo de seleção dos estudos, na qual serão aplicados os critérios de inclusão e exclusão de acordo com a estratégia definida na Seção 3.4. A terceira e quarta etapas são executadas de forma paralela e serão descritas nas seções 4.3 e 4.4 respectivamente. Por fim, a Seção 4.5 apresenta as considerações finais do Capítulo.

4.1 Busca e Disposição dos Estudos

O objetivo aqui é aplicar *string* de busca definidas na Seção 3.3, nas bases de dados eletrônicas especificadas de forma a descrever como foi aplicada a *string* de busca nestas bases e como foram obtidos os arquivos BibTex referentes aos estudos retornados. Ainda foi apresentado imagens da interface de cada base de dados no momento que foi executado a *string* de busca. Ao final, são apresentadas as considerações finais, com um resumo sobre os resultados obtidos das bases de dados.

4.1.1 Obtendo estudos da ACM Digital Library

A *Association for Computing Machinery* (ACM) criou uma biblioteca digital onde milhares de publicações estão disponíveis. Essa biblioteca é considerada a maior coleção de informação da computação e tem em seu acervo jornais, revistas e conferências assim como as próprias publicações da ACM e seu motor de busca possibilita pesquisas básicas e avançadas. Foram encontrados pelo seu motor de busca um total de 168 artigos. A Figura 4.1 mostra o resultado da pesquisa feita no motor de busca da ACM utilizando a *string* de busca definida na Seção 3.2.

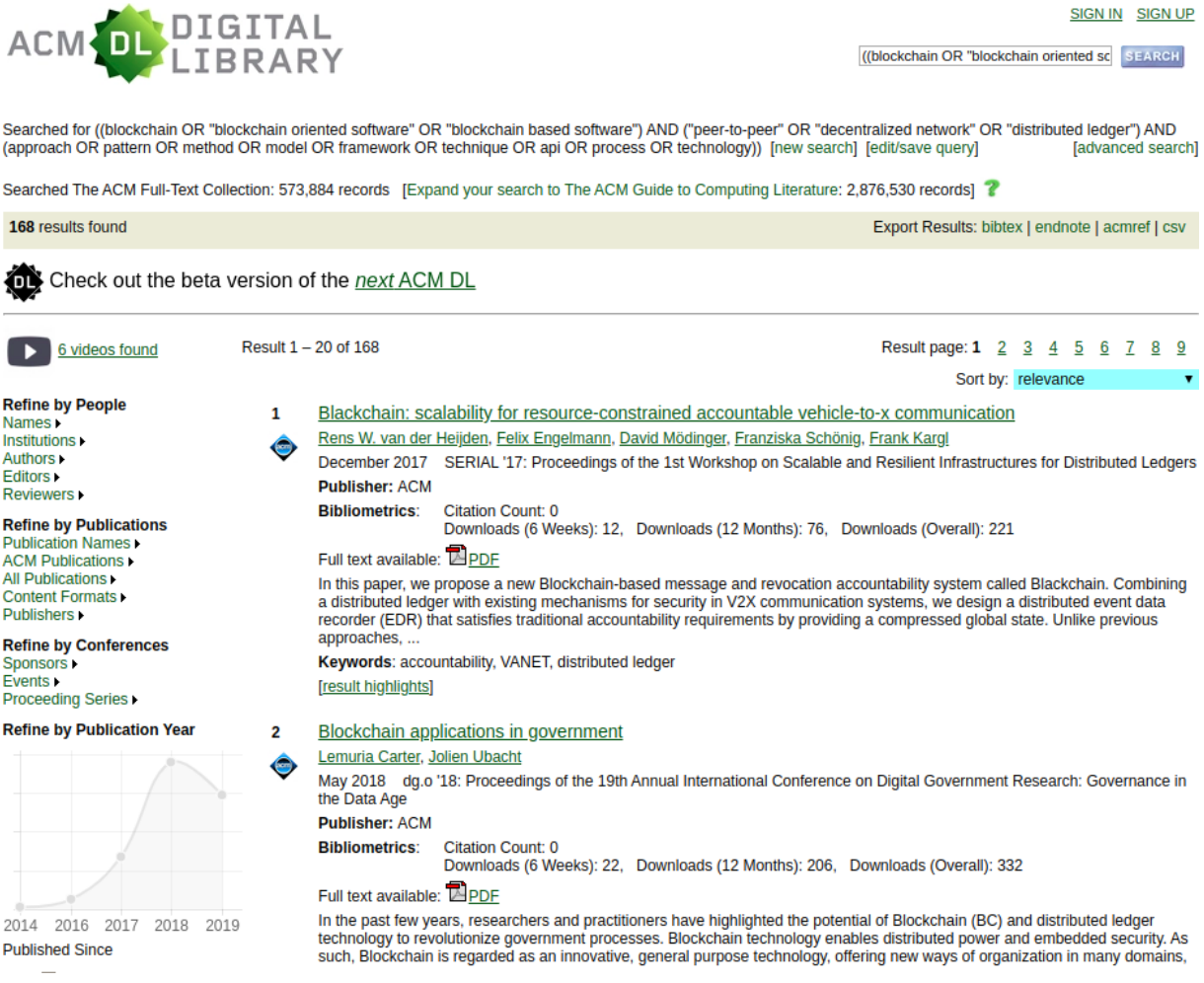

Figura 4.1: Resultados da pesquisa avançada utilizando o motor de busca da ACM

O próximo passo é obter o arquivo BibTex destes resultados. A própria ferramenta de pesquisa da ACM permite que o pesquisador obtenha esse arquivo, o que reduz consideravelmente o esforço para importar estes resultados na ferramenta StArt.

4.1.2 Obtendo estudos da Scopus

A Scopus é uma das maiores bases de dados de resumos e citações de literatura científica, possuindo mais de 24600 títulos de 5000 editoras internacionais (ELSEVIER, 2019). Ela fornece pesquisas por títulos de documentos, autores ou pesquisas avançadas e ainda o seu motor de busca, permite filtros a partir de afiliações. Assim, juntamente com a nossa *string* de busca foi utilizado o filtro "COMP" de forma que apenas resultados da área da Ciência da Computação fosse retornado. A Figura 4.2 mostra a página de pesquisa avançada da Scopus no momento da execução da nossa *string* de busca. A consulta nesta base de dados retornou 407 trabalhos. A Scopus possui o mesmo sistema de exportação de resultados que a ferramenta da ACM, podendo exportar todos os resultados diretamente para um único arquivo no formato BibTex.

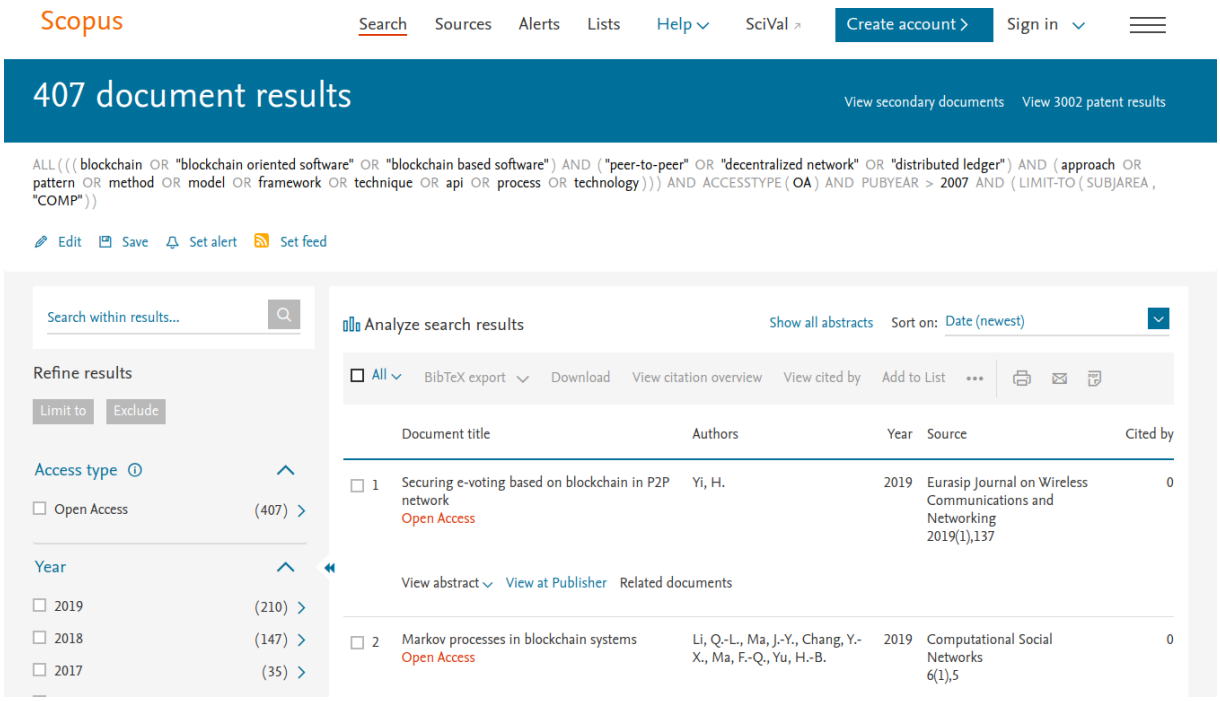

Figura 4.2: Resultados da pesquisa avançada utilizando o motor de busca da Scopus

4.1.3 Obtendo estudos da SpringerLink

SpringerLink é uma plataforma online que fornece acesso rápido e preciso a mais de 10 milhões de documentos científicos de uma coleção on-line contendo livros, periódicos, obras de referência e protocolos nas áreas de STM (Ciência, Tecnologia e Medicina) e HSS (Humanidades e Ciências Sociais), abrangendo uma vasta gama de disciplinas e foi desenvolvida para

auxiliar alunos, docentes, pesquisadores e profissionais das áreas indicadas em suas pesquisas e trabalhos acadêmicos, por meio do armazenamento, busca e recuperação de informação científica e tecnológica (SPRINGERLINK, 2019). A ferramenta permite a filtragem dos resultados por meio de filtros por disciplina, assim foi aplicado um filtro pela disciplina de Ciência da Computação. A Figura 4.3 mostra a página de pesquisa avançada da SpringerLink no momento da execução da nossa *string* de busca. A consulta nesta base de dados retornou 134 trabalhos. A SpringerLink ao contrário das outras bases de dados não possui um sistema de exportação de resultados diretamente para um único arquivo no formato BibTex, ao invés, permite que os resultados sejam exportados em formato de arquivo csv, que significa *"comma-separated-values"* (valores separados por vírgulas), o que não é interessante, tendo em vista que a ferramenta StArt não aceita esse formato de arquivo, dessa forma o pesquisador deve necessariamente criar o arquivo BibTex contendo as referencias para as publicações encontradas pelo seu motor de busca.

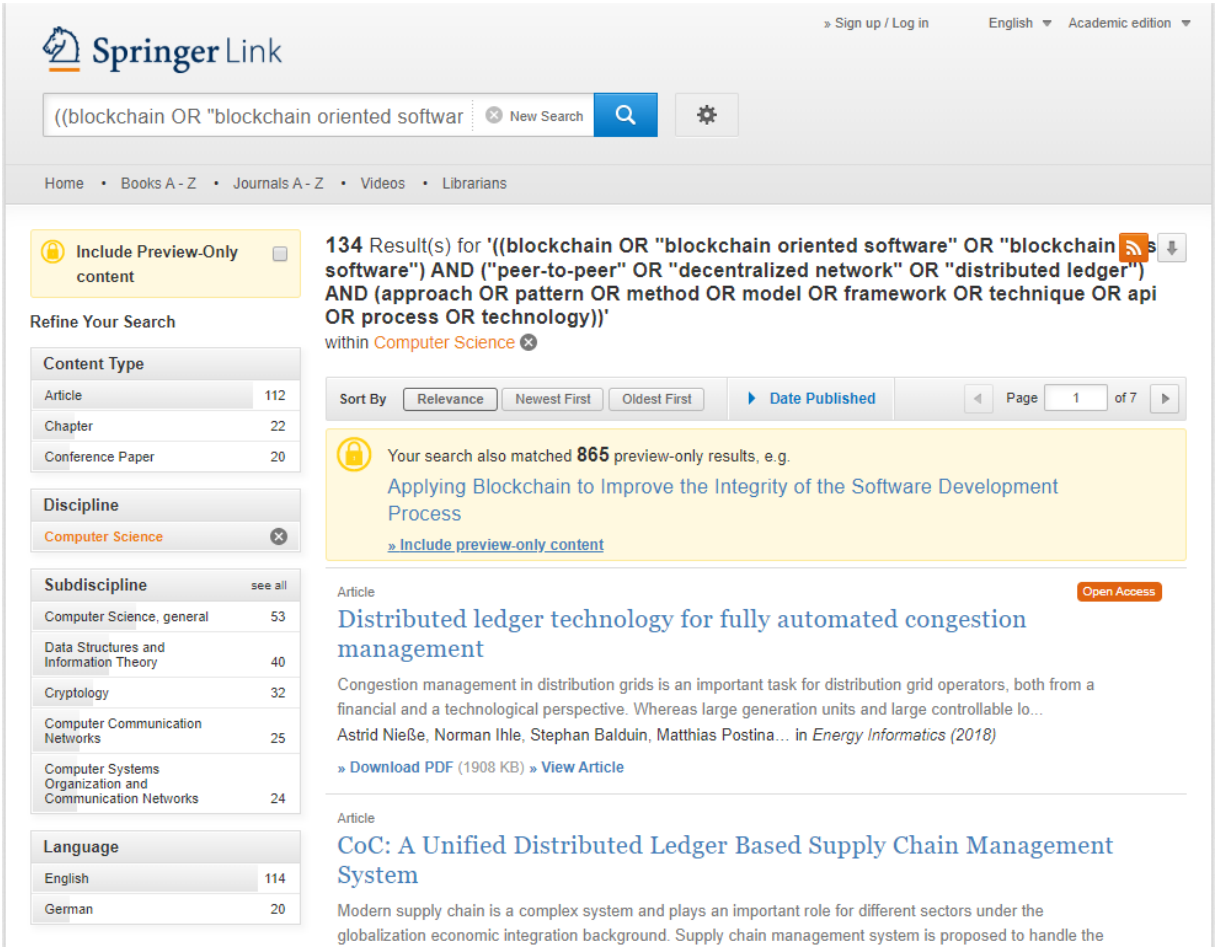

Figura 4.3: Resultados da pesquisa avançada utilizando o motor de busca da SpringerLink

4.1.4 Considerações Finais da Seção

Nesta Seção foram apresentadas as bases de dados eletrônicas e os resultados da *string* de busca aplicada em seus motores de busca. Foram encontrados ao total 709 trabalhos nas três bases de dados eletrônicas. A Tabela 4.1 mostra um resumo de quantos trabalhos foram retornados em cada base. Para uma melhor visualização, a Figura 4.4 mostra um gráfico que destaca o percentual dos resultados que cada base compôs ao total.

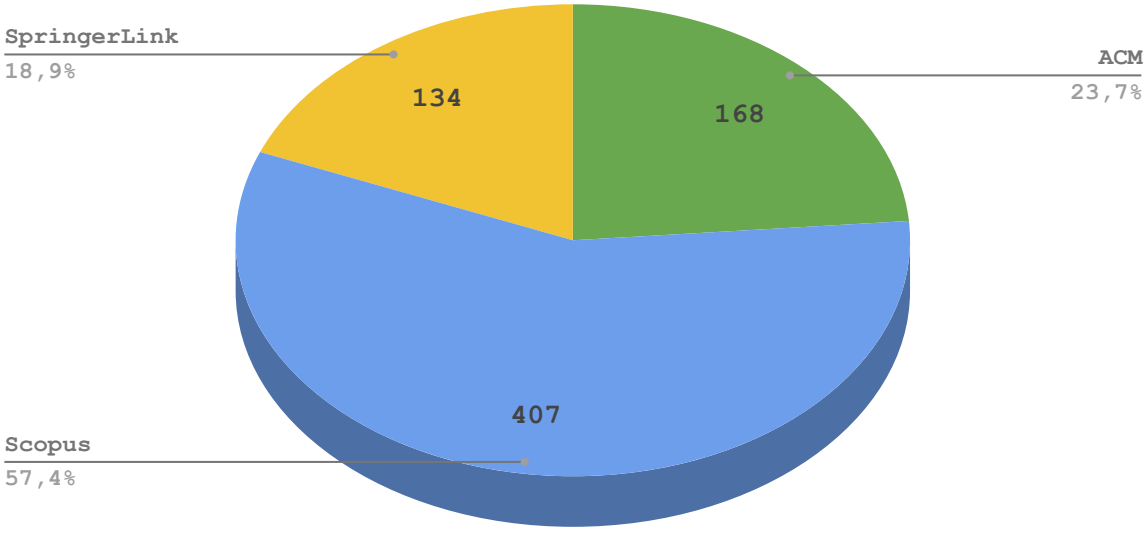

Figura 4.4: Porcentagem dos resultados em relação a cada base de dados

Tabela 4.1: Número de estudos encontrados em cada base de dados eletrônica

Base de dados	**Resultados**
ACM Digital Library	168
Scopus	407
SpringerLink	134
Total	**709**

4.2 Processo de Seleção dos Estudos

Finalizada a primeira fase de condução do MS (ver Figura 3.1) a qual o objetivo é aplicar a *string* de busca nas base de dados eletrônicas e obter os arquivos BibTex referentes aos resultados de cada consulta, a próxima fase é aplicar os critérios de inclusão e exclusão definidos na Seção 3.4.1 juntamente com a estratégia definida. Esta fase é iterativa e consiste em manter uma lista de estudos excluídos bem como a razão deles não serem considerados, e outra com os estudos selecionados. Dividida em quatro fases, o processo foi detalhado nas próximas subseções. Um resumo da execução deste processo pode ser visto na Figura 4.5. Ao final desta Seção, serão apresentadas as considerações finais, apontando algumas características sobre o processo de seleção de estudos.

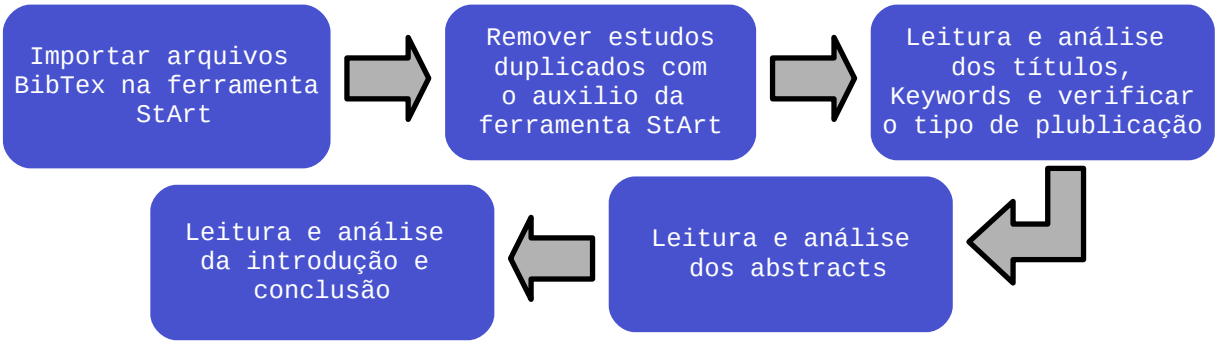

Figura 4.5: Processo de seleção dos estudos

4.2.1 A primeira fase

A primeira fase da seleção dos estudos se caracteriza pela importação dos arquivos BibTex resultantes da primeira fase da condução do MS na ferramenta StArt, a ferramenta por sua vez irá atribuir um número único para cada estudo, que servirá como um identificador. Ela ainda extrairá alguns dados automaticamente, tais como, título, local, ano de publicação e autores. Para importar os arquivos BibTex na ferramenta StArt, foi criado uma sessão de busca, nessa sessão é salvo a *string* de busca utilizada, e alguma descrição que você deseja, e ainda qual a base de dados eletrônica que foi utilizada para gerar o arquivo BibTex a ser incluído, é importante selecionar corretamente, pois cada base possui um padrão específico para seus arquivos BibTex.

Com a sessão de busca configurada, é possível ver todas as informações que a ferramenta extraiu automaticamente. A Figura 4.6 apresenta um exemplo de todas as informações obtidas

automaticamente pela ferramenta. Note que o artigo foi selecionado de forma aleatória a fim de exemplo. Ao final desta etapa foram criadas sessões de busca para todas as bases de dados eletrônicas e importados os arquivos BibTex referentes.

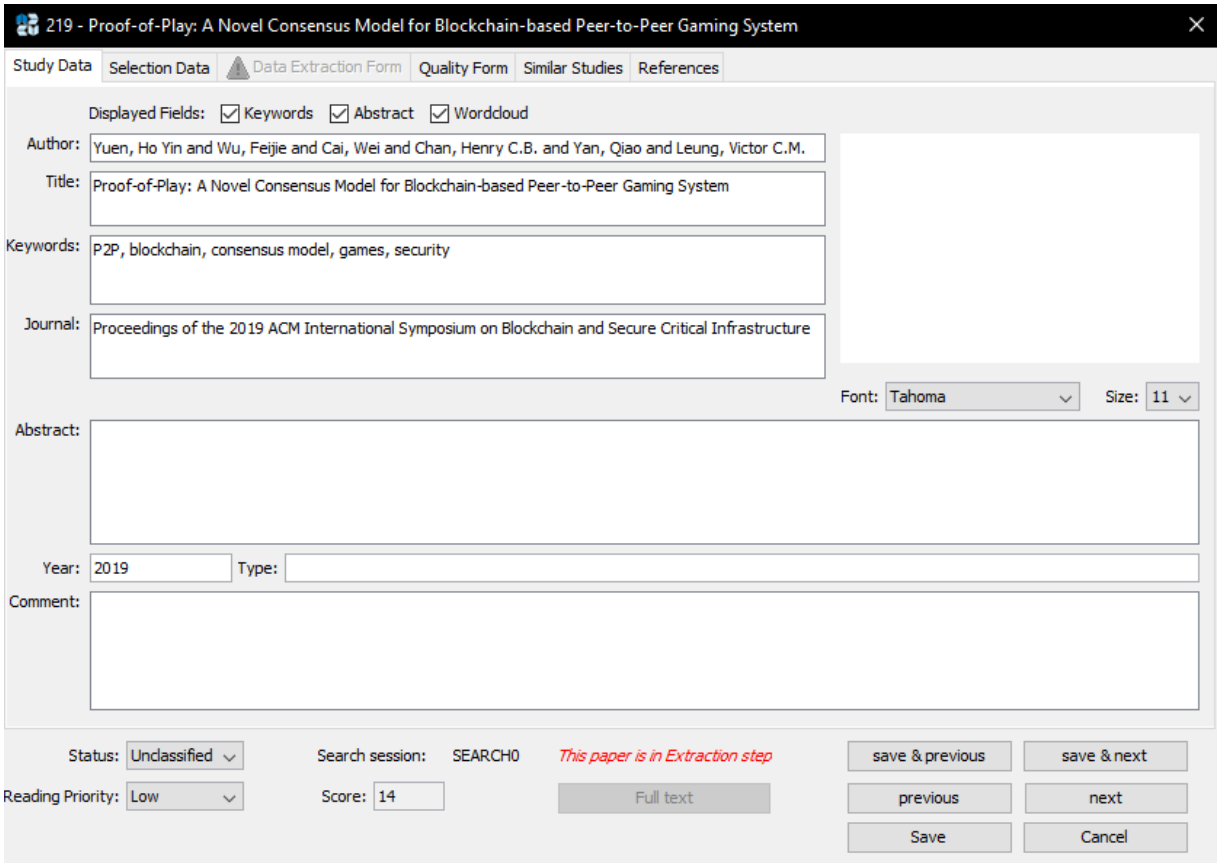

Figura 4.6: Informações extraídas automaticamente de um estudo pela ferramenta StArt

Após a seleção do arquivo BibTex e antes dele ser importado para a ferramenta, a ferramenta solicita ao usuário se ele deseja que a ferramenta identifique os artigos duplicados, ela faz isso utilizando uma comparação de *strings* de alguns campos chave dos estudos, tais como, título, ano de publicação, *abstract*, autores e local de publicação. Ainda existe a forma de eliminar estudos duplicados manualmente. A ferramenta verifica o quanto um estudo é similar a um outro qualquer já importado. Essa similaridade é calculada novamente a partir de comparações de *strings* entre os campos dos estudos. A primeira forma apresentada (forma automática) foi a utilizada neste MS. Ao final desta fase, foram identificados 1 estudo duplicado da base de dados Scopus, 6 estudos duplicados na base de dados SpringerLink e nenhum duplicado na base de dados da ACM. A Tabela 4.2 mostra um resumo dos resultados obtidos ao final desta fase. A

Figura 4.7 mostra uma visão gráfica da proporção de estudos duplicados em relação a cada base de dados.

Tabela 4.2: Número de estudos duplicados encontrados em cada base de dados eletrônica

Base de dados	**Resultados**	**Duplicados**
ACM Digital Library	168	0
Scopus	407	1
SpringerLink	134	6
Total	**709**	**7**

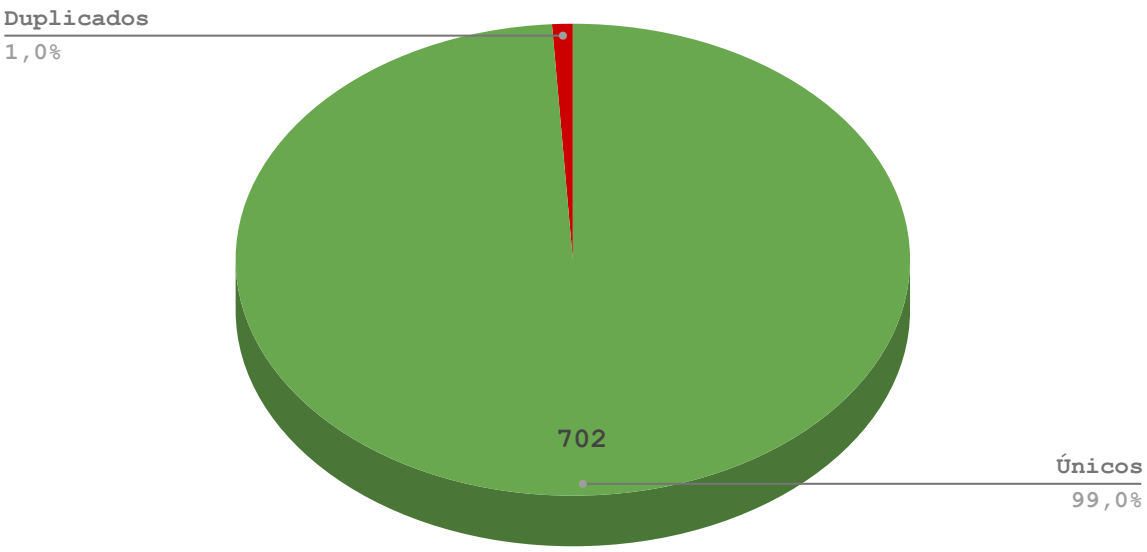

Figura 4.7: Relação entre total de resultados e resultados duplicados

Vale ressaltar que a ferramenta pode não ter encontrados todos os estudos duplicados, sendo assim o critério de exclusão de estudos duplicados ainda deve ser aplicado de forma manual nas próximas etapas.

4.2.2 A segunda fase

Nesta fase é iniciada a aplicação dos critérios de inclusão e exclusão definidos na Seção 3.4.1, estes serão aplicados com base na análise das palavras-chave e dos títulos de cada estudo.

A fim de exemplificar esta etapa foram selecionados aleatoriamente 10 estudos daqueles que não foram excluídos na etapa anterior, ainda caso haja alguma dúvida em encaixar um estudo em um critério de exclusão ou não, foi tomada a decisão de não inclui-lo nesta fase, e sanar esta dúvida nas fases seguintes. A Tabela 4.3 mostra cada estudo e suas respectivas palavras-chave.

Tabela 4.3: Estudos selecionados para exemplificação e palavras-chave encontradas

ID	**Título**	**Palavras-chave**
727	Dandelion: Redesigning the Bitcoin Network for Anonymity	Bitcoin, Cryptocurrency, Network Anonymity, Peer-to-peer
424	Open Peer-to-Peer Systems over Blockchain and IPFS: An Agent Oriented Framework	Blockchain, Decentralization, Distributed Systems, Framework, IPFS, Multi-Agent Systems, P2P Systems
718	Attack and Vulnerability Simulation Framework for Bitcoin-like Blockchain Technologies	Bitcoin, Blockchain, Simulation
787	SCPKI: A Smart Contract-based PKI and Identity System	Não dispõem
708	Mastering Bitcoin: Programming the Open Blockchain	Não dispõem
812	Safe and Irrefutable Decentralized Communication: Bringing Non-Repudiation to Mesh Networks	Blockchain, Information Security, Internet of Things, Mesh Networks
819	Defining Granularity Levels for Supply Chain Traceability Based on IoT and Blockchain	IoT, blockchain, smart contract, supply chain
825	EVA: Fair and Auditable Electric Vehicle Charging Service Using Blockchain	Blockchain, Electric Grids, Electric Vehicles, Energy Systems, Ethereum, Scheduling, Smart Contract
819	Dependable Data Sharing in Dynamic IoT-systems: Subject-oriented Process Design, Complex Event Processing, and Blockchains	S-BPM, events, internet of things, reliable communications, shared input pool, subject orientation
834	Database and Distributed Computing Fundamentals for Scalable, Fault-tolerant, and Consistent Maintenance of Blockchains	blockchain, byzantine faults, distributed consensus

O estudo ID (identificador) 708 não possui palavras-chave para análise, porém o título se refere a programação de *open blockchain*, o que se encaixa no critério de inclusão dessa forma este

estudo permanece em nossa lista de incluídos para próxima fase. Já o estudo identificado pelo ID 787 não possui nenhuma palavra-chave para análise, e seu título refere-se exclusivamente a SC baseados em infra-estrutura de chaves públicas. Logo foi aplicado o terceiro critério de exclusão (CE3), pois este estudo não se refere à abordagens para desenvolvimento de BOS. Vale lembrar que para um estudo ser excluído das próximas fases, basta que se encaixe em apenas um critério de exclusão. O estudo ID 718 se encaixa no critério de exclusão (CE3), pois seu título sugere um *framework* de simulação para tecnologias *blockchain*. O estudo ID 819, discute sobre a rastreabilidade da cadeia de recursos, logo está fora da próxima etapa pelo (CE3). O estudo ID 825 também se enquadra no critério de exclusão (CE3), pelo fato de seu título e palavras-chave sugerirem um estudo relacionado veículos elétricos. Aqueles não citados aqui como excluídos permanecem como incluídos em nossa listagem.

Analisando os 702 estudos restantes da primeira fase, 16 estudos foram excluídos por se encaixarem no primeiro critério de exclusão, outros 5 foram excluídos pelo segundo critério de exclusão, 600 saíram pelo terceiro critério de exclusão e ainda mais 16 foram excluídos pelo quinto critério de exclusão (a descrição dos critérios de exclusão pode ser vista na Seção 3.4.1). A Tabela 4.4 mostra uma suma dos resultados obtidos após a segunda fase.

Tabela 4.4: Quantidade de estudos excluídos por critério de exclusão na segunda fase

Base de dados	**Resultados**	**CE1**	**CE2**	**CE3**	**CE4**	**CE5**	**CE6**	**CE7**
ACM Digital Library	168	2	2	127	0	1	0	0
Scopus	406	5	1	380	0	0	0	0
SpringerLink	128	9	2	93	0	15	0	0
Total	**702**	**16**	**5**	**600**	**0**	**16**	**0**	**0**

Vale destacar a porcentagem de estudos excluídos por cada critério de exclusão em cada uma das bases de dados nessa segunda fase, assim a Figura 4.8 mostra essa informação para a base de dados da ACM, a Figura 4.9 para a base da Scopus e ainda a Figura 4.10 da base de dados da SpringerLink

4.2.3 A terceira fase

Nesta fase é dado continuidade à aplicação dos critérios de inclusão e exclusão, porém agora o que foi levado em consideração é a análise dos *abstracts* (resumos) daqueles estudos que não

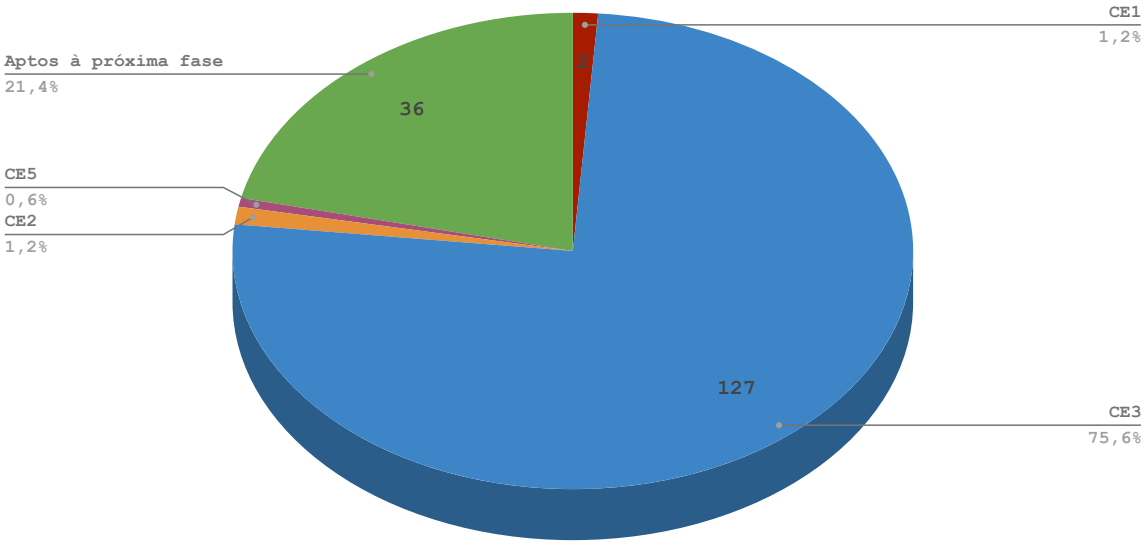

Figura 4.8: Porcentagem de estudos excluídos por critério de exclusão - ACM

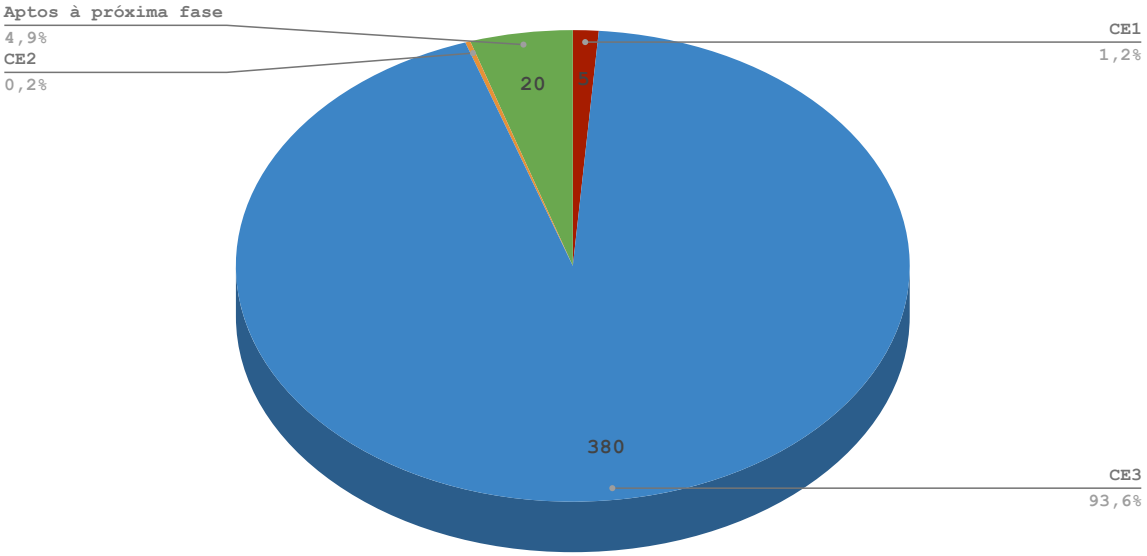

Figura 4.9: Porcentagem de estudos excluídos por critério de exclusão - Scopus

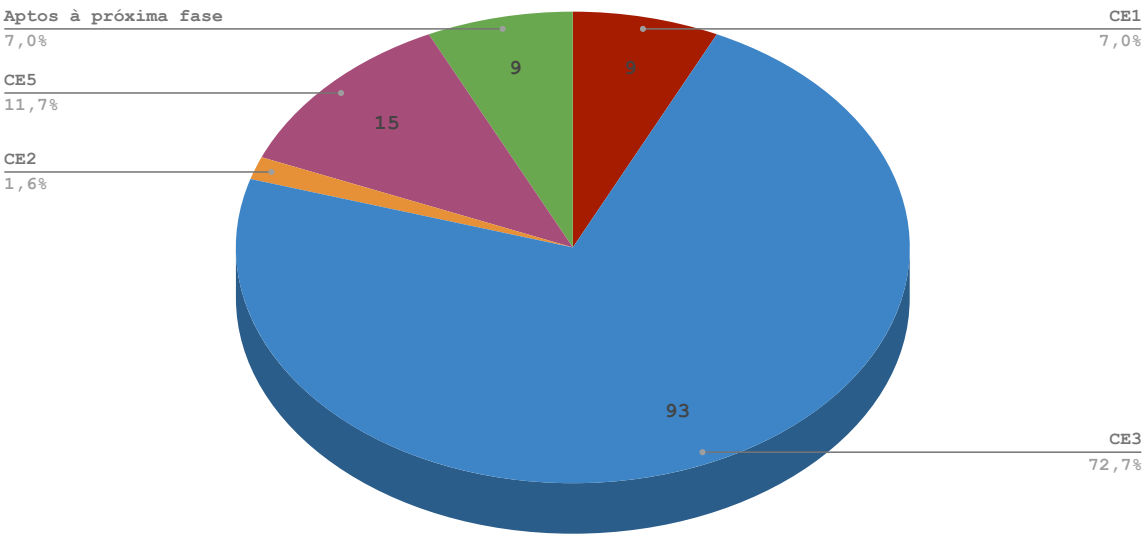

Figura 4.10: Porcentagem de estudos excluídos por critério de exclusão - Springer

foram excluídos nas fases anteriores. A Tabela 4.5 mostra os estudos remanescentes da segunda fase que servirão de exemplo para essa fase

Tabela 4.5: Estudos remanescentes da segunda fase

ID	**Título**	**Palavras-chave**
727	Dandelion: Redesigning the Bitcoin Network for Anonymity	bitcoin, cryptocurrency, network anonymity, peer-to-peer
424	Open Peer-to-Peer Systems over Blockchain and IPFS: An Agent Oriented Framework	Blockchain, Decentralization, Distributed Systems, Framework, IPFS, Multi-Agent Systems, P2P Systems
708	Mastering Bitcoin: Programming the Open Blockchain	Não dispõem
812	Safe and Irrefutable Decentralized Communication: Bringing Non-Repudiation to Mesh Networks	Blockchain, Information Security, Internet of Things, Mesh Networks
834	Database and Distributed Computing Fundamentals for Scalable, Fault-tolerant, and Consistent Maintenance of Blockchains	blockchain, byzantine faults, distributed consensus

O estudo 708 é na verdade um livro, após uma análise de seu sumário foi verificado que este trata-se de um tutorial sobre a moeda virtual Bitcoin. Como o interesse deste MS é sobre abordagem para desenvolvimento de BOS, foi aplicado o terceiro critério de exclusão. Outro estudo que foi aplicado o (CE3) é o 727 que propõe uma nova política de rede para garantia de anonimato. Ou seja, não fala sobre abordagens para desenvolvimento de BOS. Vale destacar que para os estudos de ID 424, 812, 834, não foi aplicado nenhum critério de exclusão. Sendo assim esses estudos são aprovados para a próxima fase.

Ao final, dos 65 estudos remanescentes da fase anterior, 30 foram excluídos após a análise dos *abstracts*, destes 3 através da aplicação do segundo critério de exclusão, 25 aplicando do terceiro critério de exclusão e ainda 2 através da aplicação do sétimo critério de exclusão. A Tabela 4.6 mostra uma suma dos resultados obtidos após a terceira fase.

Tabela 4.6: Quantidade de estudos excluídos por critério de exclusão na terceira fase

Base de dados	**Resultados**	**CE1**	**CE2**	**CE3**	**CE4**	**CE5**	**CE6**	**CE7**
ACM Digital Library	36	0	3	15	0	0	0	0
Scopus	20	0	0	5	0	0	0	2
SpringerLink	9	0	0	5	0	0	0	0
Total	**65**	**0**	**3**	**25**	**0**	**0**	**0**	**2**

4.2.4 A quarta fase

Nesta última fase, à aplicação dos critérios de inclusão e exclusão foi feita novamente, só que desta vez o que foi levado em consideração é a análise da introdução e conclusão de cada estudo, e se após a leitura dessas duas seções ainda existir dúvidas se pode ser utilizado algum critério de exclusão no estudo, então foi feita uma análise do estudo por completo, é claro em cima daqueles que não foram excluídos nas fases anteriores. Vale lembrar que para um estudo ser incluído para a fase de extração de dados, ele deve obrigatoriamente atender todos os critérios de inclusão para assim avançar à próxima próxima fase.

Para exemplificar a aplicação dos critérios de inclusão, foi utilizado os estudos que não foram excluídos nas fases anteriores (ver Tabela 4.7).

Para um estudo ser incluído para a fase de extração de dados, ele deve obrigatoriamente atender todos os critérios de inclusão.

Tabela 4.7: Estudos da remanescentes da terceira fase

ID	Título	Palavras-chave
424	Open Peer-to-Peer Systems over Blockchain and IPFS: An Agent Oriented Framework	Blockchain, Decentralization, Distributed Systems, Framework, IPFS, Multi-Agent Systems, P2P Systems
812	Safe and Irrefutable Decentralized Communication: Bringing Non-Repudiation to Mesh Networks	Blockchain, Information Security, Internet of Things, Mesh Networks
834	Database and Distributed Computing Fundamentals for Scalable, Fault-tolerant, and Consistent Maintenance of Blockchains	blockchain, byzantine faults, distributed consensus

O estudo de ID 424 foi publicado no *CryBlock'18: Proceedings of the 1st Workshop on Cryptocurrencies and Blockchains for Distributed Systems* em 2018. Está redigido em inglês e propõem um *framework* para desenvolvimeto de BOS com uma abordagem orientada a agentes, portanto o estudo atende a todos os critérios de inclusão, dessa forma este estudo foi aceito neste MS. O estudo 812 encontra-se indisponível gratuitamente, o texto completo foi solicitado ao autor, com a escusa de incluí-lo neste mapeamento, porém até o presente momento nenhuma resposta foi obtida. Portanto pelo sexto critério de exclusão, este estudo não foi considerado neste MS. O estudo com identificador 834 é um prólogo de um tutorial totalmente direcionado a área de banco de dados, dessa forma foi desconsiderado por esse mapeamento em acordo com o (CE2).

Após a análise completa dos 35 estudos remanescentes da terceira fase, 24 destes foram excluídos da fase de extração dos dados, destes 1 foi através da aplicação do segundo critério de exclusão, 16 aplicando o terceiro critério de exclusão e outros 8 por conta do sétimo critério de exclusão.

A Tabela 4.8 mostra uma síntese dos resultados obtidos após a ultima fase.

Tabela 4.8: Quantidade de estudos excluídos por critério de exclusão após a ultima fase

Base de dados	Resultados	CE1	CE2	CE3	CE4	CE5	CE6	CE7
ACM Digital Library	18	0	1	13	0	0	0	0
Scopus	13	0	0	2	0	0	0	7
SpringerLink	4	0	0	1	0	0	0	1
Total	**35**	**0**	**1**	**16**	**0**	**0**	**0**	**8**

4.2.5 Considerações Finais da Seção

Completo o processo de seleção dos estudos, foram selecionados 10 trabalhos que atenderam todos os critérios de inclusão para a fase de extração de dados. A Tabela 4.9 mostra a listagem desses trabalhos selecionados. Dentre os 709 estudos analisados, 641 não abordavam o tema, 23 foram considerados estudos duplicados, 9 não eram artigos científicos (*workshop calls*, tabela de conteúdo, descrição de *workshop*), 16 não estavam escritos em inglês ou português e ainda 10 não abordavam o ponto de vista da engenharia de software, essas informações podem ser vistas em forma de gráfico na Figura 4.11.

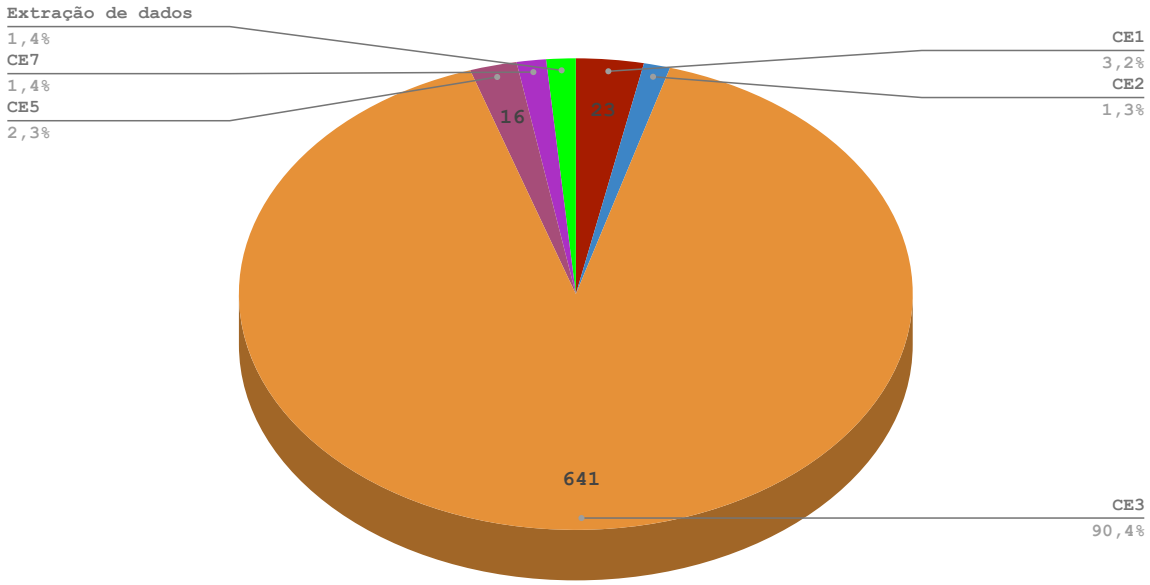

Figura 4.11: Porcentagem de estudos excluídos por critério de exclusão após o processo de seleção

4.3 Extração de Dados dos Estudos Selecionados

A extração de dados dos estudos selecionados objetiva extrair todos os dados necessários para realizar a fase de análise de resultados do MS, essa fase será apresentada no Capítulo 5. Foi extraído de cada estudo informações que respondam as questões de pesquisa, e ainda, serão extraídas algumas informações adicionais que foram descritas na Seção 3.4.2.

Tabela 4.9: Estudos selecionados para a fase de extração de dados

ID	Título
541	A blockchain-based approach for data accountability and provenance tracking
749	A General Framework for Blockchain Analytics
467	Approaches to Front-End IoT Application Development for the Ethereum Blockchain
489	Blockchain in Logistics and Supply Chain: A Lean Approach for Designing Real-World Use Cases
173	Implementing a blockchain from scratch: why, how, and what we learned
628	Interactive verification of architectural design patterns in FACTum
778	LeapChain: Efficient Blockchain Verification for Embedded IoT
608	Modeling and execution of blockchain-aware business processes
424	Open Peer-to-Peer Systems over Blockchain and IPFS: An Agent Oriented Framework
822	Reducing the Execution Time of Unit Tests of Smart Contracts in Blockchain Platforms

4.3.1 ID 541: A blockchain-based approach for data accountability and provenance tracking

Trabalho desenvolvido pelos autores Ricardo Neisse, Gary Steri e Igor Nai-Fovino, publicado em 2017 no *ARES '17 Proceedings of the 12th International Conference on Availability, Reliability and Security*, foi obtido pela base de dados eletrônica da Scopus e propõe o uso de uma abordagem baseada em *blockchain* para prover suporte a responsabilidade de dados e o rastreamento de sua procedência, basendo-se no uso de contratos publicamente auditáveis implementados em uma *blockchain*.

Para prover o rastreamento da procedência dos dados em contratos inteligentes os autores seguem um modelo de dados estruturado, incluindo a especificação de tipos de dados, instanciações de dados e instâncias de dados, assim o modelo proposto permite que tipos de dados primitivos e compostos possam ser especificados. Quando as instâncias de dados sobre um assunto são modificadas por controladores, o modelo de dados deve ser previamente acordado, para isso os autores utilizaram uma abordagem de modelagem de dados usada no Kit de Ferramentas de Segurança Baseado em Modelo - SecKit (NEISSE et al., 2015), já que esse modelo suporta a especificação de identidades de usuários, e já que objetivo é armazenar esse modelo

em uma *blockchain* pública, por motivos de privacidade, ofuscam as referências aos tipos de dados, instanciações e instâncias.

Os *smart contracts* de responsabilidade de dados especificam restrições ao uso dos dados do sujeito transferidos para os controladores, sabendo disso os autores especificam essas restrições usando a linguagem da política de segurança proposta pelo SecKit.

Neste artigo, os autores focam principalmente o design, implementação e desempenho de três modelos propostos usando a plataforma EVM (*Ethereum Virtual Machine*). Em relação ao ciclo de vida do contrato inteligente, identificam três possíveis modelos com diferentes cardinalidades em relação ao número de titulares e controladores de dados e definem os três como sendo:

a) Contrato do titular dos dados para o controlador específico: o sujeito cria um contrato personalizado para cada controlador que gerencia seus dados. O contrato controla os dados compartilhados com o controlador, as políticas regulam o uso dos dados e registra os eventos de uso de dados que representam as atividades executadas pelo controlador usando os dados do sujeito como entrada.

b) Contrato do titular dos dados para dados específicos: o sujeito cria um contrato genérico para cada instância de dados que é compartilhada por todos os controladores que acessam os dados. O contrato contém a lista de controladores que tiveram acesso a uma instância de dados específica e as políticas que eles devem respeitar.

c) Contrato do controlador para vários titulares de dados: o controlador cria um contrato especificando como os dados recebidos de todos os titulares são tratados. Os titulares dos dados então aderem ao contrato, caso aceitem as políticas de uso de dados do controlador.

Levando em consideração esses três possíveis modelos e avaliando suas características individuais os autores fornecem um modelo de controle de procedência e responsabilidade. A Figura 4.12 apresenta a arquitetura de alto nível do modelo. Nesta arquitetura, três entidades principais são representadas seguindo a terminologia do GDPR (Regulamento Geral sobre a Proteção de Dados): o titular dos dados, o controlador de dados e o processador de dados. Quando o sujeito assina um controlador, que normalmente é a função de um provedor de serviços, ele cria um contrato de uso de dados com base em políticas, especificando restrições ao uso

e redistribuição de quaisquer dados obtidos explícita ou implicitamente pelo controlador. Dados explícitos são quaisquer dados fornecidos diretamente por meio de interações com alguém, como os endereços de *email* ou a data de nascimento. Dados implícitos são quaisquer dados adquiridos automaticamente, por exemplo, dados de sensores de dispositivos IoT no ambiente ao redor do sujeito. O contrato neste modelo atua como um rastreador de procedência dos dados, uma entidade de avaliação de políticas e um registrador de eventos que permite ao sujeito verificar facilmente todas as transferências de dados e transações de uso, garantindo que apenas as transações em conformidade com as políticas do contrato sejam autorizadas e registradas na *blockchain*.

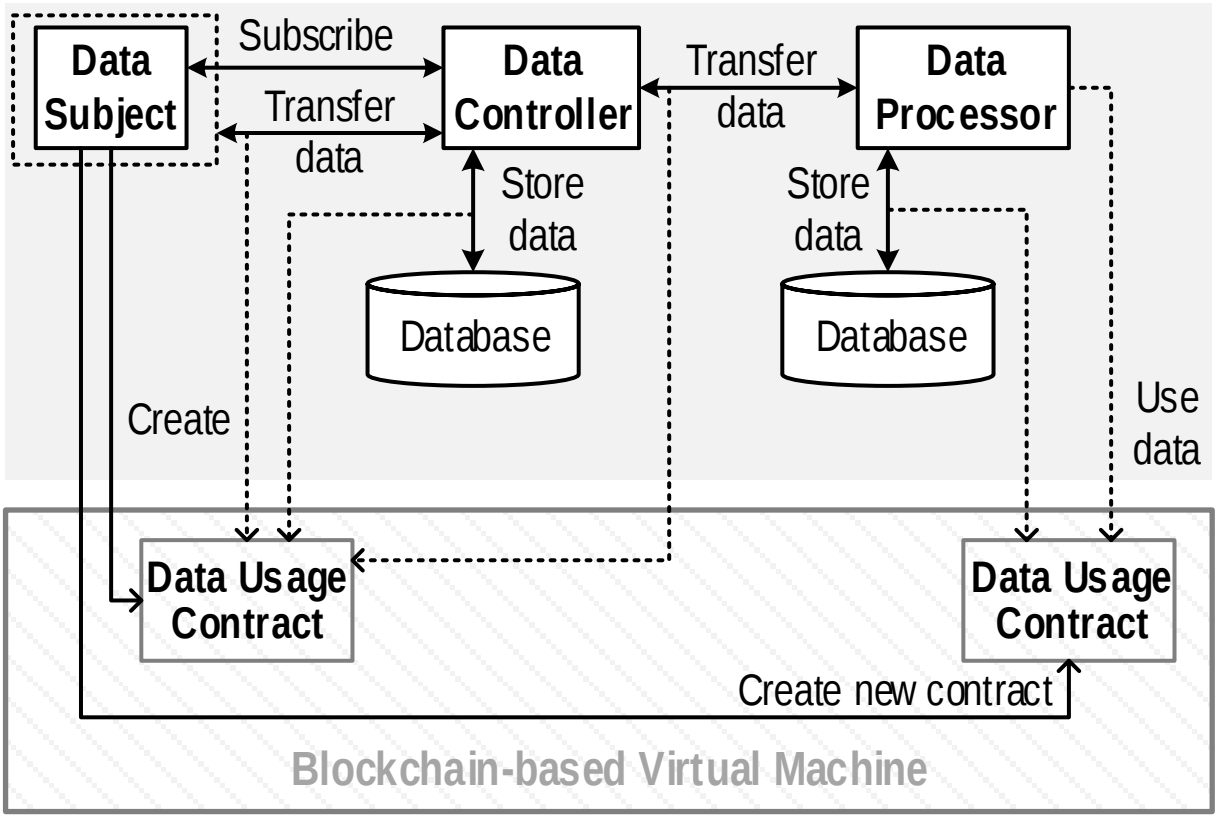

Figura 4.12: Arquitetura de alto nível do modelo de controle de procedência e responsabilidade (NEISSE; STERI; NAI-FOVINO, 2017)

Nesse modelo, o controlador ou processador cria um contrato especificando as restrições compartilhadas no uso e redistribuição de quaisquer dados explícitos ou implícitos obtidos de todos os assuntos que assinam o controlador. O contrato neste modelo atua como um repositório dos modelos de políticas configuráveis. As transações são solicitações para entrar ou sair do

contrato. A Figura 4.13 mostra um exemplo de modelos de aplicação e configuração de políticas que podem ser usados em um contrato de uso de dados neste modelo.

```
Default Enforcement: Deny
PolicyTemplate0
 Variables: Entity(s)
     Event: sendMessage(purpose= billing ,
              isDataSubject(e-mail, s))
 Condition: not(within(30 days, sendMessage(purpose= billing ,
                 isDataSubject(e-mail, s))))
    Action: Allow
ConfigurationTemplate0
   Variables: Entity(s)
 Assignments: s = xpath(\\trigger\user)
       Event: subscribeUser()
   Condition: true
      Action: configure(PolicyTemplate0, s)
```

Figura 4.13: Exemplo de modelos de aplicação e configuração de políticas (NEISSE; STERI; NAI-FOVINO, 2017)

Para a validação da proposta os autores implementaram um contrato de amostra em que a política é configurada como uma sequência e podendo ser uma codificação XML ou JSON usando a linguagem de política SecKit, o código fonte pode ser visto no artigo original.

O formulário de extração de dados deste artigo pode ser visto na Figura 4.14

4.3.2 ID 749: A General Framework for Blockchain Analytics

Trabalho desenvolvido pelos autores Massimo Bartoletti, Andrea Bracciali, Stefano Lande, e Livio Pompianu, publicado no *Proceedings of the 1st Workshop on Scalable and Resilient Infrastructures for Distributed Ledgers*, foi obtido pela base de dados eletrônica da ACM e propõem um *framework* para criar análises de uso geral nas *blockchains* Bitcoin e Ethereum.

Os autores baseiam o design do *framework* em uma pesquisa exaustiva da literatura sobre a análise de *blockchains*. Dentre os resultados dessa pesquisa, destacam a necessidade de processar dados externos além daqueles já presentes na *blockchains*. Para isso, a ferramenta proposta suporta um fluxo de trabalho que consiste em duas etapas: a primeira é a construção de uma

ID	541	**Base de dados**	Scopus
Título	A blockchain-based approach for data accountability and provenance tracking		
Autores	Neisse, R. and Steri, G. and Nai-Fovino, I.		
Palavras-chave	Controllers, Security of data, Block-chain, Data accountabilities, Data controllers, Data subjects, European union, General data protection regulations, Protection requirements, Three models, Data handling		
Resumo	The recent approval of the General Data Protection Regulation (GDPR) imposes new data protection requirements on data controllers and processors with respect to the processing of European Union (EU) residents' data. .ese requirements consist of a single set of rules that have binding legal status and should be enforced in all EU member states. In light of these requirements, we propose in this paper the use of a blockchain-based approach to support data accountability and provenance tracking. Our approach relies on the use of publicly auditable contracts deployed in a blockchain that increase the transparency with respect to the access and usage of data. We identify and discuss three models for our approach with di.erent granularity and scalability requirements where contracts can be used to encode data usage policies and provenance tracking information in a privacy-friendly way. From these three models we designed, implemented, and evaluated a model where contracts are deployed by data subjects for each data controller, and a model where subjects join contracts deployed by data controllers in case they accept the data handling conditions. Our implementations show in practice the feasibility and limitations of contracts for the purposes identified in this paper.		
Questões de pesquisa			
QP1: Como tem sido construídas as abordagens de desenvolvimento de BOS?			
"The solution for data accountability and provenance tracking proposed in this paper relies on a public blockchain-based distributed ledger platform, namely the open source Ethereum Virtual Machine (EVM)", "For the purposes of this paper we adopt the data modeling approach used in the Model-based Security Toolkit (SecKit)".			
QP2: Quais os pontos críticos e/ou desafios dessas abordagens?			
"The adoption of any of these approaches has an impact on the censorship resistance, privacy, anonymity, performance, and scalability". "From a scalability point of view the controller generic approach is the best one, but it also constrains a possible solution to public blockchains due to the high number of transactions".			
QP3: Como estas abordagens foram validadas?			
We have implemented a sample contract where the policy is configured as a string, which could be an XML or JSON encoding of the policy specified using the SecKit policy language.			
QP4: Quais os problemas têm sido apontados pelas organizações de software que adotam as abordagens para desenvolvimento de BOS?			
The recent approval of the General Data Protection Regula- tion (GDPR) imposes new data protection requirements on data controllers and processors with respect to the process- ing of European Union (EU) resident's data.			
QP5: Quais são as tecnologias usadas para prover gestão e desenvolvimento das abordagens para BOS?			
"The solution for data accountability and provenance tracking proposed in this paper relies on a public blockchain-based distributed ledger platform, namely the open source Ethereum Virtual Machine (EVM)", "For the purposes of this paper we adopt the data modeling approach used in the Model-based Security Toolkit (SecKit)".			
Facetas			
Tipo do estudo		Tecnologias/Ferramentas	
Exemplo (Toy example).		Ethereum Virtual Machine (EVM), Model-based Security Toolkit (SecKit), Chave-pública, Ethereum.	
Tipo de contribuição		Classificação e principais características das redes blockchain	
Guidelines		Privada.	
Limitações na Arquitetura de Blockchain		Finalidade da abordagem	
Escalabilidade e privacidade de dados.		Prover suporte a responsabilidade de dados e o rastreamento de sua procedência.	
Tipo de aplicação			
Propósito Geral.			

Figura 4.14: Formulário de extração de dados - A blockchain-based approach for data accountability and provenance tracking

visão da *blockchain*, que já contém os dados externos necessários, e é armazenada em um banco de dados; a segunda etapa consiste na análise da visão criada usando a linguagem de consulta do sistema de gerenciamento de banco de dados.

O *framework* permite que as visões sejam organizadas como um banco de dados MySQL ou uma coleção do MongoDB e ainda suporta os dados externos mais usados, por exemplo, taxas

de câmbio, etiquetas de endereço, identificadores de protocolo, sendo que esses são tratados de uma forma especial sendo discutido detalhadamente suas características e exemplos de códigos-fonte no artigo original. A Figura 4.15 apresenta a utilização da API (interface de programação de aplicações) para construção de uma *blockchain*.

A API da biblioteca proposta fornece as seguintes classes Scala para representar as entidades primitivas da *blockchain*:

- BlockchainLib: classe principal da biblioteca. Ela fornece o método "*getBlockchain*", para iterar sobre objetos do tipo "*Block*".

- Block: contém uma lista de transações e alguns atributos relacionados ao bloco (por exemplo, *hash* do bloco e tempo de criação).

- Transaction: contém vários atributos relacionados (por exemplo, *hash* e tamanho da transação).

```
object MyBlockchain {
    def main(args: Array[String]): Unit = {

    val blockchain = BlockchainLib.getBitcoinBlockchain(
        new BitcoinSettings("user", "password", "8332", MainNet))
    val mongo = new DatabaseSettings("myDatabase", MongoDB, "user", "password
        ")
    val myBlockchain = new Collection("myBlockchain", mongo)

    blockchain.end(473100).foreach(block => {
        block.bitcoinTxs.foreach(tx => {
            myBlockchain.append(List(
                ("txHash", tx.hash),
                ("blockHash", block.hash),
                ("date", block.date),
                ("inputs", tx.inputs),
                ("outputs", tx.outputs)
            ))
        })
    })
    }
}
```

Figura 4.15: Visão básica de uma *blockchain* em Scala (BARTOLETTI et al., 2017)

Embora o *framework* seja geral o suficiente para cobrir a maioria das análises elencadas pela pesquisa exaustiva realizada pelos autores, os mesmos citam algumas limitações que podem ser superadas com extensões futuras. Em particular, algumas análises abordando propagação de informações, bifurcações e ataques exigem a coleta de dados da rede ponto-a-ponto subjacente.

A estrutura é avaliada por meio de um conjunto de casos de uso paradigmáticos, que são distribuídos juntamente com o código fonte do *framework*, sob uma licença de código aberto. Os casos de uso são explorados de forma a avaliar o desempenho dos bancos de dados SQL versus NoSQL para armazenamento e consultas das visualizações de *blockchain*. E, ainda ao final, os autores fornecem uma comparação qualitativa de outras ferramentas para análises de *blockchain* de uso geral.

O formulário de extração de dados deste artigo pode ser visto na Figura 4.16

ID	749	**Base de dados**	ACM Digital Library
Título	A General Framework for Blockchain Analytics		
Autores	Bartoletti, Massimo and Lande, Stefano and Pompianu, Livio and Bracciali, Andrea		
Palavras-chave	Analytics, Bitcoin, Blockchain, Ethereum		
Resumo	Modern cryptocurrencies exploit decentralised blockchains to record a public and unalterable history of transactions. Besides transactions, further information is stored for different, and often undisclosed, purposes, making the blockchains a rich and increasingly growing source of valuable information, in part of difficult interpretation. Many data analytics have been developed, mostly based on specifically designed and ad-hoc engineered approaches. We propose a general-purpose framework, seamlessly supporting data analytics on both Bitcoin and Ethereum — currently the two most prominent cryptocurrencies. Such a framework allows us to integrate relevant blockchain data with data from other sources, and to organise them in a database, either SQL or NoSQL. Our framework is released as an open-source Scala library. We illustrate the distinguishing features of our approach on a set of significant use cases, which allow us to empirically compare ours to other competing proposals, and evaluate the impact of the database choice on scalability		
Questões de pesquisa			
QP1: Como tem sido construídas as abordagens de desenvolvimento de BOS?			
Its main component is a Scala library which can be used to construct views of the blockchain, possibly integrating blockchain data with data retrieved from external sources. Blockchain views can be stored as SQL or NoSQL databases, and can be analysed by using their query languages.			
QP2: Quais os pontos críticos e/ou desafios dessas abordagens?			
Although our framework is general enough to cover most of the analyses, it has some limitations that can be overcome with future extensions. In particular, some analyses addressing e.g. information propagation, forks and attacks.			
QP3: Como estas abordagens foram validadas?			
We illustrate our framework through some case studies, which, for uniformity, have been developed for the Bitcoin case			
QP4: Quais os problemas têm sido apontados pelas organizações de software que adotam as abordagens para desenvolvimento de BOS?			
Não tem.			
QP5: Quais são as tecnologias usadas para prover gestão e desenvolvimento das abordagens para BOS?			
Its main component is a Scala library which can be used to construct views of the blockchain, possibly integrating blockchain data with data retrieved from external sources. Blockchain views can be stored as SQL or NoSQL databases, and can be analysed by using their query languages.			
Facetas			
Tipo do estudo			Tecnologias/Ferramentas
Simulação Experimental			SQL, NoSQL, Bitcoin Core, Parity, Ethereum, Scala, BitcoinJ, web3j, ScalikeJDBC, DSL
Tipo de contribuição			Classificação e principais características das redes blockchain
Framework			Publica
Limitações na Arquitetura de Blockchain			Finalidade da abordagem
Propagação de informações, *forks* e ataques			Desenvolvimento de análises de uso geral nas plataformas blockchain da Bitcoin e Ethereum
Tipo de aplicação			
Propósito Geral.			

Figura 4.16: Formulário de extração de dados - A blockchain-based approach for data accountability and provenance tracking

4.3.3 ID 467: Approaches to Front-End IoT Application Development for the Ethereum Blockchain

Trabalho desenvolvido pelos autores Matevž Pustišeka e Andrej Kosa, publicado no *Procedia Computer Science Volume 129*, foi obtido pela base de dados eletrônica da Scopus e realiza uma análise e apresenta restrições práticas no desenvolvimento de aplicativos de IoT baseados no *blockchain* Ethereum (ETH).

O estudo fornece instruções para os desenvolvedores de aplicativos da IoT, permitindo que eles selecionem o design do sistema apropriado e evitem expectativas irreais impostas aos dispositivos IoT e às tecnologias *blockchain*.

Para tanto os autores elaboram e comparam as abordagens arquiteturais para o design dos aplicativos de dispositivos IoT *front-end* com base no ETH. Segundo eles, a arquitetura das partes *front-end* dos aplicativos depende muito dos recursos e limitações dos dispositivos IoT, onde os *front-ends* serão implantados pois demonstram uma ampla gama de recursos de comunicação (taxa de bits, persistência de conectividade) e computação (CPU, memória volátil). Assim, destacam que é necessário conhecer esses recursos com antecedência.

Destacam ainda que as considerações sobre a arquitetura e as configurações têm como objetivo fornecer uma execução confiável da lógica do aplicativo *front-end* e do fluxo de trabalho das transações ETH (criação, assinatura, envio, monitoramento) relacionadas a um dispositivo específico.

A primeira arquitetura a qual os autores elaboram sobre, é a intitulada "*Stand-alone IoT node architecture*" nela todos os blocos funcionais são executados no mesmo dispositivo físico. Como o cliente ETC "*geth*" também está sendo executado lá, ele impõe uma alta demanda de CPU e memória volátil. Se a sincronização completa do *blockchain* estiver ativada, é necessário contar com vários gigabyte de dados da cadeia ETH para serem transferidos e armazenados no dispositivo. O principal risco nesse caso é a segurança do hardware (chaves roubadas, se a privacidade do dispositivo físico for violada). Segundo os autores essa arquitetura é adequada apenas para os dispositivos IoT mais poderosos. Foi ainda executado o cliente completo no sistema incorporado RPi v3B com uma conexão de internet com fio. Os resultados mostraram segundo os autores que a sincronização da cadeia provou ser altamente confiável, porém tiveram alguns casos onde as sincronizações foram longas, incomuns (a sincronização durou vários dias

e não foi concluída) e interrupções inesperadas na sincronização. Durante a realização desses testes, um cliente de referência era executado em um computador regular (mesmas capacidades de rede IP) não houve problemas com a sincronização. Essa arquitetura é retratada na Figura 4.17 (a).

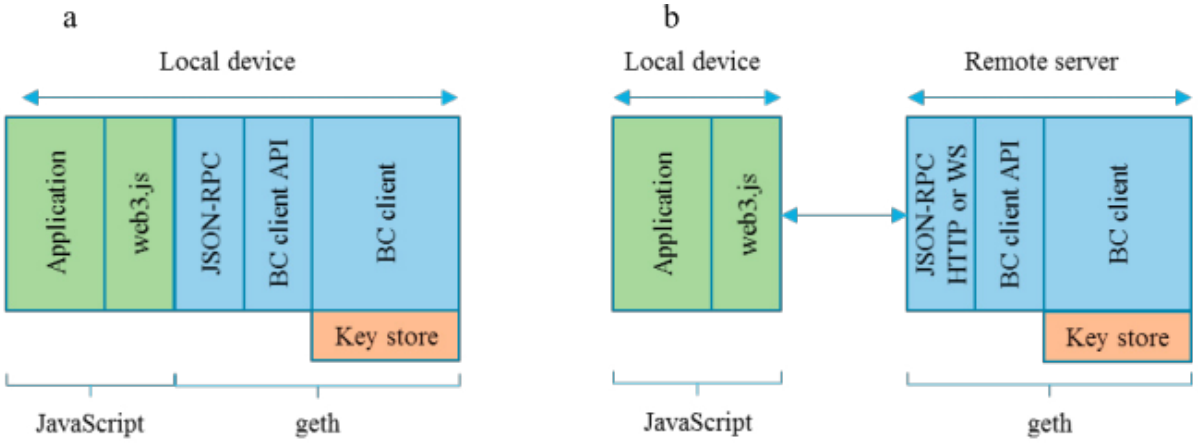

Figura 4.17: . (a) Stand-alone IoT node and (b) Cliente ETC "*geth*" remoto (PUSTIŠEK; KOS, 2018)

Já a segunda arquitetura abordada, o cliente "*geth*" remoto e o armazenamento de chaves remotas são salvos em um servidor separado e sem restrições e a parte JavaScript permanece no dispositivo IoT local ver Figura 4.17 (b). O servidor remoto expõe a funcionalidade "*geth*" pela API JSON-RPC, com HTTP ou Websockets (WS) como o canal de transporte. Esta arquitetura, segundo os autores, mostrou ter um valor prático. Segundo eles o dispositivo local conseguiu executar a parte do aplicativo, enquanto um servidor remoto executou o "*geth*" sem problemas.

A terceira arquitetura é o cliente "*geth*" remoto com arquitetura de armazenamento de chaves local. Como não está no servidor, o risco de segurança do compartilhamento do mesmo servidor "*geth*" por vários dispositivos diminui. Nesse caso, no entanto, o aplicativo deve ainda criar e enviar transações brutas, incluindo a assinatura e aplicar a serialização adequada. Para possibilitar isso, são necessárias bibliotecas JavaScript adicionais.

E, por fim, a última arquitetura abordada é descrita como uma comunicação proprietária de dispositivo local para servidor remoto. Os autores citaram um protocolo de comunicação proprietário entre o dispositivo local da IoT e o servidor remoto ("*geth*"), descartando também os formatos de dados JSON ou RLP existentes. Foram vistos dois benefícios nele, primeiro, os requisitos de largura de banda de comunicação podem ser reduzidos ao mínimo, e segundo,

é possível aplicar controles avançados de acesso ao servidor para minimizar os riscos de segurança descritos no cliente "*geth*" remoto com arquitetura de armazenamento de chaves remotas.

O formulário de extração de dados deste artigo pode ser visto na Figura 4.18

ID	467	**Base de dados**	Scopus
Título	Approaches to Front-End IoT Application Development for the Ethereum Blockchain		
Autores	Matevž Pustišeka, Andrej Kosa		
Palavras-chave	Architecture, Blockchain, Internet protocols, Location, Network architecture, Application architecture, Architectural approach, Ethereum, Front end Internet of thing (IOT), IOT applications, Mobile Technology, Network traffic, Internet of things		
Resumo	There are several distributed ledger protocols potentially suitable for the Internet of things (IoT), including the Ethereum, Hyperledger Fabric and IOTA. This paper briefly presents and compares them from the IoT application development perspective. The IoT applications based on blockchain (BC) can incorporate the on-chain logic the smart contracts and Web, mobile or embedded client front-end application parts. We present three possible architectures for the IoT front-end BC applications. They differ in positioning of Ethereum blockchain clients (local device, remote server) and in positioning of key store needed for the management of outgoing transactions. The practical constraints of these architectures, which utilize the Ethereum network for trusted transaction exchange, are the data volumes, the location and synchronization of the full blockchain node and the location and the access to the Ethereum key store. Results of these experiments indicate that a full Ethereum node is not likely to reliably run on a constrained IoT devices. Therefore the architecture with remote Ethereum clients seems to be a viable approach, where two sub-options exist and differ in key store location/management. In addition, we proposed the use of architectures with a proprietary communication between the IoT device and remote blockchain client to further reduce the network traffic and enhance security. We expect it to be able to operate over low-power, low-bitrate mobile technologies, too. Our research clarifies differences in architectural approaches, but final decision for a particular ledger protocol and front-end application architecture is at strongly based on the particular intended use case.		
Questões de pesquisa			
QP1: Como tem sido construídas as abordagens de desenvolvimento de BOS?			
"Front-end IoT device applications based on ETH BC"			
QP2: Quais os pontos críticos e/ou desafios dessas abordagens?			
The architecture of the front-end applications parts heavily depends on the capabilities and limitations of the IoT devices.			
QP3: Como estas abordagens foram validadas?			
We tried to run the full client on RPi v3B embedded system with a wired internet connection			
QP4: Quais os problemas têm sido apontados pelas organizações de software que adotam as abordagens para desenvolvimento de BOS?			
Não tem.			
QP5: Quais são as tecnologias usadas para prover gestão e desenvolvimento das abordagens para BOS?			
"As the BC client (geth) is running there as well, it imposes high demand on CPU and memory", "We tried to run the full client on RPi v3B embedded system with a wired internet connection", "The JavaScript application part of course remains in the local IoT device.", "A remote server exposes geth functionality over JSON-RPC API, with HTTP or WS as the transport channel", "The rest were message TCP/IP protocol headers"			
Facetas			
Tipo do estudo		Tecnologias/Ferramentas	
Estudo de caso		Blockchain, Ethereum, JSON-RPC, HTTP, WebSockets, IoT, RPi v3B, TCP/IP, JavaScript	
Tipo de contribuição		Classificação e principais características das redes blockchain	
Guidelines		Privada	
Limitações na Arquitetura de Blockchain		Finalidade da abordagem	
Taxa de transferência, Latência, Tamanho e largura de banda, Recursos desperdiçados		Instruir para o desenvolvimento de Sistemas IoT orientados a Blockchain	
Tipo de aplicação			
Sistemas IoT orientados a Blockchain			

Figura 4.18: Formulário de extração de dados - Approaches to Front-End IoT Application Development for the Ethereum Blockchain

4.3.4 ID 489: Blockchain in Logistics and Supply Chain: A Lean Approach for Designing Real-World Use Cases

Trabalho desenvolvido pelos autores Guido Perboli, Stefano Musso e Mariangela Rosano, publicado em 2018 no *IEEE Access (Volume: 6)*, foi obtido pela base de dados eletrônica da Scopus e contribui com a literatura em dois eixos. Primeiro, ele integra as outras contribuições, propondo uma metodologia padrão e repetível para abordar o design da estratégia digital nos projetos *blockchain*. Segundo, discutem resultados de um caso de uso na entrega de alimentos frescos, mostrando os aspectos críticos da implementação de uma solução *blockchain*. O artigo discute ainda como o *blockchain* ajudará a reduzir os custos de logística e otimizar as operações. Um aspecto que os autores frisam é que, ao contrário de outras contribuições baseadas em simulações, a análise realizada por eles se refere a uma solução *blockchain* que foi realmente implementada.

No artigo é usada a metodologia *GUEST* (*GO*, *UNIFORM*, *EVALUATE*, *SOLVE* e *TEST*) para o design de casos de uso. Tal metodologia visa fornecer às empresas uma estrutura inovadora para gerenciamento de negócios. A metodologia *GUEST* controla o processo, desde a ideia original até sua implementação, fornecendo ferramentas conceituais e práticas para os diferentes atores envolvidos, permitindo que eles comuniquem suas visões, dificuldades e oportunidades dentro da mesma estrutura onde cada etapa permite que os atores monitorem seus projetos e, ao mesmo tempo, concede a padronização de documentos e ferramentas que devem ser utilizados para avaliar ideias, sucessos, ações e resultados.

Para aplicar corretamente a metodologia, o primeiro passo, segundo os autores, é definir os diferentes atores envolvidos no processo, identificando para cada ator os trabalhos (o que eles estão tentando alcançar em seu trabalho), os ganhos (os benefícios concretos que estão buscando) e as dificuldades (problemas relacionados ao seu trabalho). Uma vez coletados os trabalhos, ganhos e dificuldades de cada ator, é possível priorizá-los para destacar os mais importantes ou urgentes e visualizá-los através do *Value Ring*, uma ferramenta gráfica capaz de mostrar imediatamente as reais necessidades dos atores. Os autores aplicam a metodologia *GUEST* para projetar o caso de uso relacionado a um varejista de alimentos para comércio eletrônico localizado na Europa.

O formulário de extração de dados deste artigo pode ser visto na Figura 4.19

ID	489	**Base de dados**	Scopus
Título	Blockchain in Logistics and Supply Chain: A Lean Approach for Designing Real-World Use Cases		
Autores	Perboli, G. and Musso, S. and Rosano, M.		
Palavras-chave	Design, Finance, Standards, Supply chains, Systems engineering, Business process model, Digital strategies, Digital supply chain, Financial applications, Hyperledger, Leading technology, Research challenges, Technological parts, Blockchain		
Resumo	The Blockchain technology can be defined as a distributed ledger database for recording transactions between parties verifiably and permanently. Blockchain emerged as a leading technology layer for financial applications. Nevertheless, in the past years, the attention of researchers and practitioners moved to the application of the Blockchain technologies to other domains. Recently, it represents the backbone of a new digital supply chain. Thanks to its capability of ensuring data immutability and public accessibility of data streams, Blockchain can increase the efficiency, reliability, and transparency of the overall supply chain, and optimize the inbound processes. The literature concerning Blockchain in non-financial applications mainly focused on the technological part and the Business Process Modeling, lacking in terms of standard methodology for designing a strategy to develop and validate the overall Blockchain solution and integrate it in the Business Strategy. Thus, this paper aims to overcome this lack. First, we integrate the current literature filling the lack concerning the digital strategy, creating a standard methodology to design Blockchain technology use cases, which are not related to finance applications. Second, we present the results of a use case in the fresh food delivery, showing the critical aspects of implementing a Blockchain solution. Moreover, the paper discusses how the Blockchain will help in reducing the logistics costs and in optimizing the operations and the research challenges.		
Questões de pesquisa			
QP1: Como tem sido construídas as abordagens de desenvolvimento de BOS?			
GUEST aims to provide companies with an innovative framework for business management.			
QP2: Quais os pontos críticos e/ou desafios dessas abordagens?			
Não tem.			
QP3: Como estas abordagens foram validadas?			
An essential aspect of the novelty of this paper is that, on the contrary of other contributions based on simulations, our analysis refers to a Blockchain solution that has been really implemented.			
QP4: Quais os problemas têm sido apontados pelas organizações de software que adotam as abordagens para desenvolvimento de BOS?			
Não tem.			
QP5: Quais são as tecnologias usadas para prover gestão e desenvolvimento das abordagens para BOS?			
The GUEST methodology, used in this paper for the use case design.			
Facetas			
Tipo do estudo		Tecnologias/Ferramentas	
Estudo de caso		GUEST	
Tipo de contribuição		Classificação e principais características das redes blockchain	
Guidelines		Privada	
Limitações na Arquitetura de Blockchain		Finalidade da abordagem	
Não tem.		Metodologia para planejamento e desenvolvimento de BOS	
Tipo de aplicação			
Propósito Geral.			

Figura 4.19: Formulário de extração de dados - Blockchain in Logistics and Supply Chain: A Lean Approach for Designing Real-World Use Cases

4.3.5 ID 173: Implementing a blockchain from scratch: why, how, and what we learned

Trabalho desenvolvido pelos autores Fabian Knirsch, Andreas Unterweger, e Dominik Engel, publicado em 2019 no *EURASIP Journal on Information Security*, foi obtido pela base de dados eletrônica da Scopus e tem como objetivo ser um guia para os outros, mostrando oportunidades e armadilhas em potencial ao implementar uma *blockchain* para um campo de aplicação específico que não seja transações financeiras.

Nele é investigado um caso de uso de duas grandes empresas de serviços públicos austríacos: O legislador nacional, que também se aplica a outros países europeus, exige que, para uma propriedade compartilhada de pequenas usinas fotovoltaicas, como as comumente encontradas em apartamentos de aluguel, os clientes podem trocar partes de sua produção de energia com os vizinhos. Para este caso de uso, é avaliado a necessidade da tecnologia *blockchain* e a aplicabilidade de diferentes tipos de algoritmos de consenso e permissões.

Para isso os autores levaram em consideração várias implementações de *blockchain* existentes, como MultiChain, OpenChain, Ethereum e um *fork* do Bitcoin, mas devido a problemas de escalabilidade no hardware desejado e para obter uma solução leve e simples, implementaram uma *blockchain* do zero, assim descrevem como a fizeram e discutem os resultados ao final.

Uma visão geral da configuração, incluindo nós, servidor e aplicativo, da *blockchain* criada pode ser vista na Figura 4.20. Cada cliente possui um aplicativo conectado a um nó (para enviar novas transações e receber confirmações) e ao servidor (para cobrança). Novas transações (indicadas como Tx) residem em um conjunto de transações não confirmadas até serem extraídas por um nó e anexadas à *blockchain*. O servidor então recupera os dados da *blockchain* e envia informações de cobrança para os aplicativos de *smartphone* dos clientes.

Os nós são implementados em Java 8 e projetados para rodar no Raspberry Pi 2 Modelo B. O hardware foi escolhido segundo os autores devido à sua disponibilidade geral, facilidade de uso e pouco consumo de energia, o que se assemelha bastante aos recursos de hardware de um medidor inteligente. A Figura 4.21 construída pelos autores apresenta uma série de limitações nas diversas implementações de *blockchain* incluindo aquela apresentada nesse estudo.

O formulário de extração de dados deste artigo pode ser visto na Figura 4.22

4.3.6 ID 628: Interactive verification of architectural design patterns in FACTum

Trabalho desenvolvido pelos autores Diego Marmsoler e Habtom Kashay Gidey, publicado em 2019 na *Formal Aspects of Computing*, foi obtido pela base de dados eletrônica da SpringerLink e apresentam um modelo para arquiteturas e técnicas (potencialmente dinâmicas) para especificar padrões de projeto arquiteturais sobre esse modelo. E ainda, introduzem uma estrutura baseada em *Isabelle/Higher-Order Logic* para a verificação interativa de arquiteturas e fornecem um algoritmo para mapear uma especificação de padrão para uma teoria correspon-

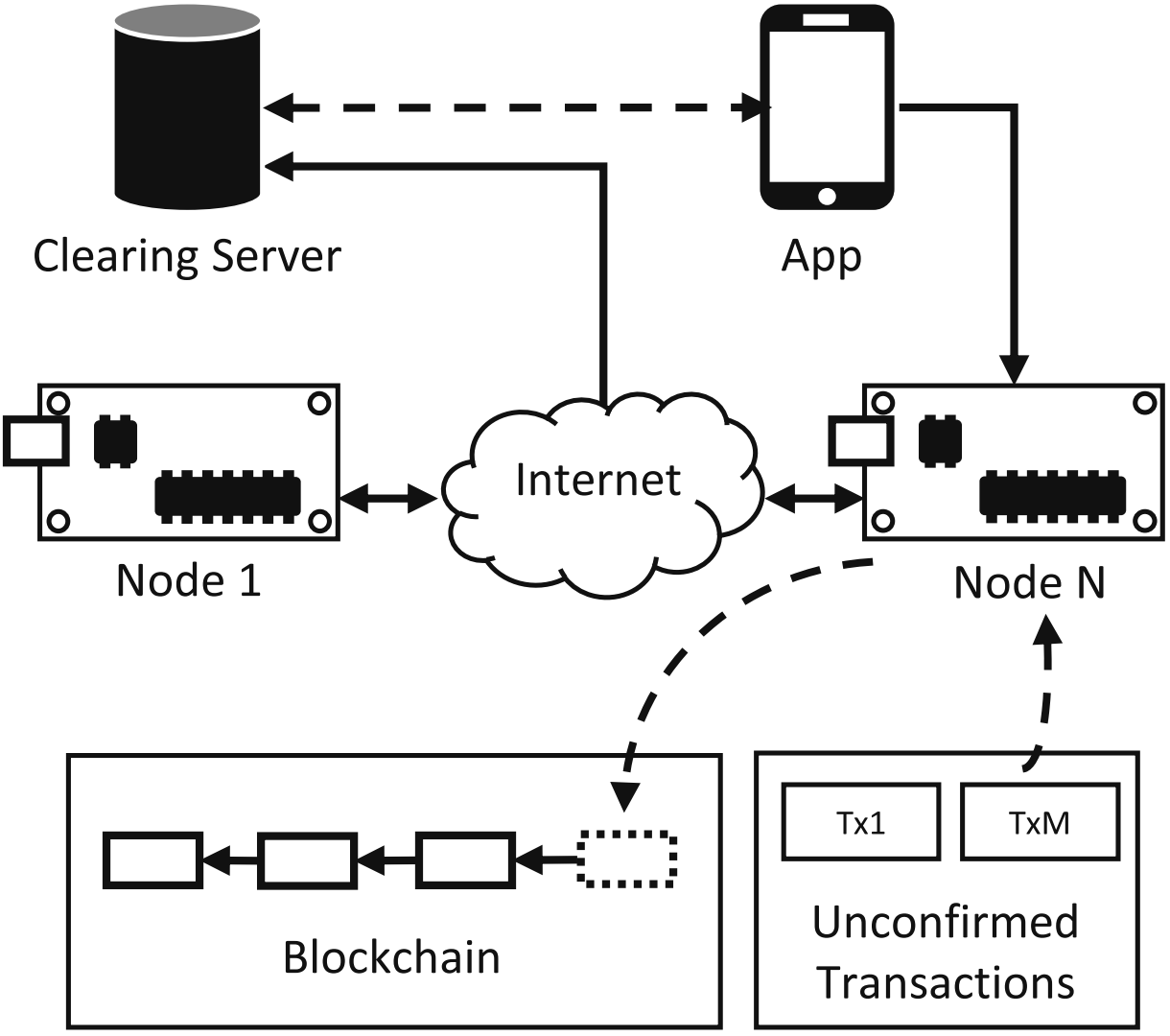

Figura 4.20: Visão geral da configuração, incluindo nós, servidor de compensação, aplicativo, da *blockchain*(KNIRSCH; UNTERWEGER; ENGEL, 2019)

dente em *Isabelle/Higher-Order Logic*. Para avaliar a abordagem, implementam em termos de um aplicativo em Eclipse/EMF e a aplicam para a verificação de quatro padrões de projeto arquiteturais diferentes: o *Singleton*, o *Publisher-Subscriber*, o padrão *Blackboard* e um padrão para arquiteturas *Blockchain*.

Como existem padrões para arquiteturas estática e dinâmica, a abordagem apresentada é baseada em um modelo de arquitetura dinâmica. O modelo apresentado consiste nos seguintes conceitos principais:

- Mensagens e portas (digitadas com conjuntos de mensagens).

Name	Consensus	Permissioned	Limitation
Bitcoin	PoW	–	Extent of modifications infeasible
Ethereum	PoW	–	Complexity exceeds requirements
MultiChain	PoW	√	Limited to high power consumption platforms
OpenChain	PoA	√	PoA algorithm not suitable for use case
Hyperledger Sawtooth	Dynamic	√	Not mature at time of evaluation
Hyperledger Fabric	Dynamic	√	Known security flaws
HAWK	PoW	–	Implementation not available
Corda	PoA	√	PoA algorithm not suitable for use case
Tendermint	BFT	√	Not available at time of evaluation
Stellar	BFT	–	Not mature at time of evaluation
EOS	BFT	√	Not mature at time of evaluation
NEO	BFT	–	Complexity exceeds requirements
OmniLedger	BFT	–	Not available at time of evaluation
ByzCoin	BFT	–	Extent of modifications infeasible
This work	PoW	√	Requires tamper-proof hardware

Figura 4.21: Limitações nas diversas implementações de *blockchain*(KNIRSCH; UNTERWEGER; ENGEL, 2019)

- Interfaces que consistem em portas de entrada e saída.

- Um conjunto de tipos de componentes que consistem em uma interface, parâmetros de componentes avaliados com mensagens e comportamento associado em termos de um conjunto causal de rastreamentos de comportamento, isto é, fluxos de capturas instantâneas de um componente durante a execução.

- Uma especificação de arquitetura que consiste em um conjunto de rastreamentos de arquitetura, isto é, fluxos de capturas instantâneas de uma arquitetura durante a execução.

- Um operador de projeção, que extrai o comportamento de um único componente de um determinado rastreamento de arquitetura.

- Um operador de composição que combina um conjunto de tipos de componentes com uma determinada especificação de arquitetura.

O formulário de extração de dados deste artigo pode ser visto nas Figuras 4.23 e 4.24

ID	173	Base de dados	Scopus
Título	Implementing a blockchain from scratch: why, how, and what we learned		
Autores	Knirsch, F. and Unterweger, A. and Engel, D.		
Palavras-chave	Photovoltaic cells, Energy domain, Financial transactions, Implementation, Photovoltaic power plant, Practical insights, Real-world, User interaction, Blockchain		
Resumo	Blockchains are proposed for many application domains apart from financial transactions. While there are generic blockchains that can be molded for specific use cases, they often lack a lightweight and easy-to-customize implementation. In this paper, we introduce the core concepts of blockchain technology and investigate a real-world use case from the energy domain, where customers trade portions of their photovoltaic power plant via a blockchain. This does not only involve blockchain technology, but also requires user interaction. Therefore, a fully custom, private, and permissioned blockchain is implemented from scratch. We evaluate and motivate the need for blockchain technology within this use case, as well as the desired properties of the system. We then describe the implementation and the insights from our implementation in detail, serving as a guide for others and to show potential opportunities and pitfalls when implementing a blockchain from scratch.		
Questões de pesquisa			
QP1: Como tem sido construídas as abordagens de desenvolvimento de BOS?			
"For achieving a lightweight and simple solution, a blockchain has been implemented from scratch", "Each participant has a node installed in their home which runs the blockchain node software as well as a portable app which allows sending commands to the node, in particular to send and receive portions of solar energy. All nodes are interconnected via the Internet, albeit within a virtual private network (VPN), so that they can communicate with one another despite not using public IP addresses. At each utility's premises, there is a clearing server installed. It is connected to a node listening on the blockchain and reading the portions of the utility's customers in order to calculate the net energy consumption of each participant".			
QP2: Quais os pontos críticos e/ou desafios dessas abordagens?			
Scalability can be improved with Byzantinum Fault Tolerance algorithms when the number of users is small. "Requires tamper-proof hardware"			
QP3: Como estas abordagens foram validadas?			
The implemented solution is currently in use in two locations operated by different energy providers and network operators: the Austrian Smart Grid Model region Köstendorf in Salzburg and Böheimkirchen in Lower Austria.			
QP4: Quais os problemas têm sido apontados pelas organizações de software que adotam as abordagens para desenvolvimento de BOS?			
"Despite the advantages of decentralization, trustlessness, and immutability, there are two major issues with current blockchain technology scalability and power consumption."			
QP5: Quais são as tecnologias usadas para prover gestão e desenvolvimento das abordagens para BOS?			
"The nodes are implemented in Java 8 and designed to run on Raspberry Pi 2 Model B." "The state of the tree is persisted using Java's serialization API, and the state table is stored in an SQLite". "Nodes communicate with each other, the app and the clearing server over TCP/IP and XML messages". "All communication links between nodes as well as between the node and the app are secured with TLS v1.2 using a hybrid encryption scheme based on Elliptic Curve Diffie-Hellman key exchange and AES-256 with CBC"			
Facetas			
Tipo do estudo			Tecnologias/Ferramentas
Estudo de caso			Java 8, Raspberry Pi 2 Model B, SQLite, TCP/IP, XML, TLS v1.2, Elliptic Curve Diffie-Hellman, AES-256
Tipo de contribuição			Classificação e principais características das redes blockchain
Guidelines			Privada
Limitações na Arquitetura de Blockchain			Finalidade da abordagem
Escalabilidade e privacidade de dados			Metodologia para planejamento e desenvolvimento de BOS
Tipo de aplicação			
Propósito Geral			

Figura 4.22: Formulário de extração de dados - Implementing a blockchain from scratch: why, how, and what we learned

4.3.7 ID 778: LeapChain: Efficient Blockchain Verification for Embedded IoT

Trabalho desenvolvido pelos autores Emanuel Regnath e Sebastian Steinhorst, publicado em 2018 no *Proceedings of the International Conference on Computer-Aided Design*, foi obtido pela base de dados eletrônica da ACM Digital Library e trabalha uma abordagem para

ID	628	Base de dados	SpringerLink
Título	Interactive verification of architectural design patterns in FACTum		
Autores	Marmsoler, Diego and Gidey, Habtom Kashay		
Palavras-chave	Architecture design patterns, Interactive theorem proving, Architecture verification, FACTum, Algebraic specification, Isabelle		
Resumo	Architectural design patterns (ADPs) are architectural solutions to common architectural design problems. They are an important concept in software architectures used for the design and analysis of architectures. An ADP usually constrains the design of an architecture and, in turn, guarantees some desired properties for architectures implementing it. Sometimes, however, the constraints imposed by an ADP do not lead to the claimed guarantee. Thus, applying such patterns for the design of architectures might result in architectures which do not fulfill their intended requirements. To address this problem, we propose an approach for the verification of ADPs, based on interactive theorem proving. To this end, we introduce a model for dynamic architectures and a language for the specification of ADPs over this model. Moreover, we propose a framework for the interactive verification of such specifications based on Isabelle/HOL. In addition we describe an algorithm to map a specification to a corresponding Isabelle/HOL theory over our framework. To evaluate the approach, we implement it in Eclipse/EMF and use it for the verification of four ADPs: variants of the Singleton, the Publisher-Subscriber, the Blackboard pattern, and a pattern for Blockchain architectures. With our approach we complement traditional approaches for the verification of architectures, which are usually based on automatic verification techniques such as model checking.		
Questões de pesquisa			
QP1: Como tem sido construídas as abordagens de desenvolvimento de BOS?			
"we present a model for (potentially dynamic) architectures and techniques to specify ADPs over this model. Then, we introduced an Isabelle/HOL-based framework for the interactive verification of architectures and provided an algorithm to map a pattern specification to a corresponding Isabelle/HOL theory. To evaluate the approach, we implemented it in terms of an Eclipse/EMF application and applied it for the verification of four different ADPs: the Singleton, the Publisher-Subscriber, the Blackboard pattern, and a pattern for Blockchain architectures."			
QP2: Quais os pontos críticos e/ou desafios dessas abordagens?			
it makes the approach also difficult to apply, since ITP comes with a steep learning curve and is not yet well-known in the architecture community. Users still need to have some expertise in ITP to efficiently use the approach and thus it might be difficult to apply for practitioners. Challenges: Advanced support for graphical notation: So far, however, expressivity of graphical annotations is limited to activation and connection constraints. To support the graphical specification of more advanced constraints, future work should investigate possibilities to extend the notion of architecture diagrams. One possible extension, for example, could be the introduction of dependencies between components of a certain type in a notation similar to UML composition. Then, activation and connection annotations could be interpreted relative to such dependencies. Pattern verification language: However, architects are usually not trained in interactive theorem proving and future work should investigate possibilities to further support an architect in the verification process. A first step could be the development of a more abstract proof language which allows an architect to sketch a proof using abstractions he is familiar with. The abstract proof should then be translated to a corresponding Isabelle/Isar proof and verified by Isabelle. Integration into architecture verification: Another crucial step to achieve our vision concerns the integration of verification results obtained for ADPs to support the verification of architectures. Compared to the verification of ADPs (which can be reused for different architectures), verification of architectures against ADPs has to be done for each architecture, which is why future work should investigate possibilities to automate this step.			
QP3: Como estas abordagens foram validadas?			
We evaluated the approach by means of four case studies: the Singleton pattern, the Publisher-Subscriber pattern, the Blackboard pattern, and a pattern for Blockchain architectures.			
QP4: Quais os problemas têm sido apontados pelas organizações de software que adotam as abordagens para desenvolvimento de BOS?			
Não tem.			
QP5: Quais são as tecnologias usadas para prover gestão e desenvolvimento das abordagens para BOS?			
interactive theorem prover Isabelle/HOL, Singleton/Publisher-subscriber/Blackboard archiectural design patterns, Eclipse/EMF (Xtext, Sirius, Xtend)			

Figura 4.23: Formulário de extração de dados parte (a) - Interactive verification of architectural design patterns in FACTum

verificação de inclusão e integridade de blocos dentro de uma *blockchain* em dispositivos IoT embarcados altamente restritos.

Como solução os autores propõem uma estrutura de dados de *blockchain* genérica, aplicável a qualquer tipo de tecnologia *blockchain* e seus correspondentes algoritmos para verificação

Facetas	
Tipo do estudo	Tecnologias/Ferramentas
Pesquisa de avaliação	Isabelle/HOL para o provador de teoremas interativo, Padrão de Projeto "Blockchain", Ecosiistema Eclipse para implementação do framework
Tipo de contribuição	Classificação e principais características das redes blockchain
Framework/Algoritmo	POS (mas abordagem deve funcionar para todas)
Limitações na Arquitetura de Blockchain	Finalidade da abordagem
1) mining frequencies in blockchain architectures. To apply the pattern, one needs to ensure that it will indeed be highly unlikely that the mining frequency of untrusted nodes exceeds the mining frequency of trusted nodes by the number of confirmation blocks. Otherwise, entries of a blockchain may be subject to modification by untrusted entities and the pattern would fail its guarantee. 2) many descriptions of blockchain architectures require the data entries to be financial transactions with corresponding private and public keys. However, these assumptions are not required to guarantee persistence of entries and they unnecessarily restrict the application scope of the pattern.	Especificação e verificação de um design pattern "Blockchain architectures"
Tipo de aplicação	
Geral	

Figura 4.24: Formulário de extração de dados parte (b) - Interactive verification of architectural design patterns in FACTum

eficiente de blocos e denominam LeapChain. O conceito reduz as etapas de verificação por meio de *backlinks* adicionais e permite que os dispositivos embarcados verifiquem o conteúdo do *blockchain* com um reduzido uso de espaço de armazenamento volátil (ver Figura 4.25). Os autores apresentam um novo padrão de interligação usando apenas um *backlink* adicional por bloco, e ainda fornecem métodos para a construção da estrutura, que permite ainda a verificação de consenso para a PoW usando apenas uma quantidade logarítmica de blocos.

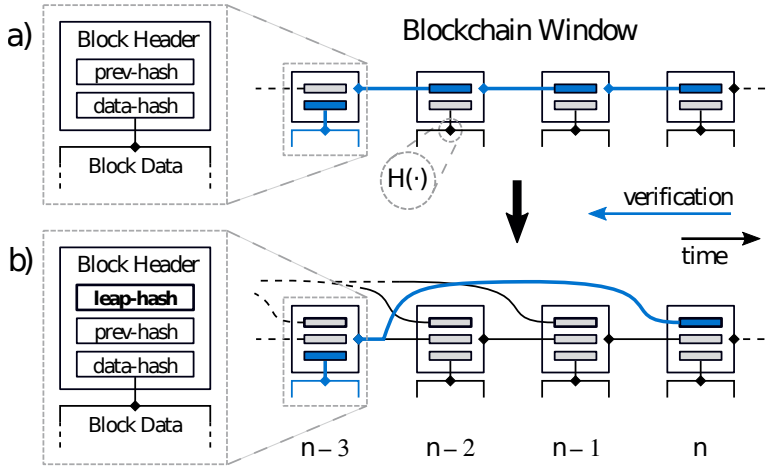

Figura 4.25: LeapChain Verification (REGNATH; STEINHORST, 2018)

Para a validação da proposta os autores primeiro introduzem limites superiores aos requi-

sitos de recursos, que permitem selecionar uma plataforma de hardware integrada apropriada e depois, comparam experimentalmente o LeapChain com trabalhos relacionados em uma simulação que ilustra os ganhos gerais de desempenho e o hardware embarcado para sustentar a viabilidade da LeapChain. Tal simulação é feita em Python nos cabeçalhos de blocos de *blockchains* gerados aleatoriamente usando a estrutura. A proposta foi ainda testada no chipset ESP32 (2 x 240 MHz) usando MicroPython v1.8.6 com 57 kB de SRAM disponível na placa WiPy 2.0.

O formulário de extração de dados deste artigo pode ser visto na Figura 4.26

ID	778	**Base de dados**	ACM Digital Library
Título	LeapChain: Efficient Blockchain Verification for Embedded IoT		
Autores	Emanuel Regnath e Sebastian Steinhorst		
Palavras-chave	SPV, Blockchain, Embedded, Internet of Things		
Resumo	Blockchain provides decentralized consensus in large, open networks without a trusted authority, making it a promising solution for the Internet of Things (IoT) to distribute verifiable data, such as firmware updates. However, verifying data integrity and consensus on a linearly growing blockchain quickly exceeds memory and processing capabilities of embedded systems. As a remedy, we propose a generic blockchain extension that enables highly constrained devices to verify the inclusion and integrity of any block within a blockchain. Instead of traversing block by block, we construct a LeapChain that reduces verification steps without weakening the integrity guarantees of the blockchain. Applied to Proof-of-Work blockchains, our scheme can be used to verify consensus by proving a certain amount of work on top of a block. Our analytical and experimental results show that, compared to existing approaches, only LeapChain provides deterministic and tight upper bounds on the memory requirements in the kilobyte range, significantly extending the possibilities of blockchain application on embedded IoT devices.		
Questões de pesquisa			
QP1: Como tem sido construídas as abordagens de desenvolvimento de BOS?			
"The main foundation of this paper is our blockchain extension that inserts additional connections with a special backlink pattern to speed up traversing the chain without weakening its integrity guarantees."			
QP2: Quais os pontos críticos e/ou desafios dessas abordagens?			
Não tem.			
QP3: Como estas abordagens foram validadas?			
"We simulated our LeapChain approach in Python on the block headers of randomly generated blockchains using our block structure". "In order to compare the approaches on a real embedded IoT platform, we tested them on the ESP32 chipset (2 × 240 MHz) using MicroPython v1.8.6 with 57 kB available SRAM on the WiPy 2.0 board"			
QP4: Quais os problemas têm sido apontados pelas organizações de software que adotam as abordagens para desenvolvimento de BOS?			
Não tem.			
QP5: Quais são as tecnologias usadas para prover gestão e desenvolvimento das abordagens para BOS?			
"We simulated our LeapChain approach in Python on the block headers of randomly generated blockchains using our block structure". "In order to compare the approaches on a real embedded IoT platform, we tested them on the ESP32 chipset (2 × 240 MHz) using MicroPython v1.8.6 with 57 kB available SRAM on the WiPy 2.0 board. For the implementation we used SHA-256 of the uhashlib"			
Facetas			
Tipo do estudo		Tecnologias/Ferramentas	
Simulação Experimental		Python, Sha256, MicroPython	
Tipo de contribuição		Classificação e principais características das redes blockchain	
Abordagem (Algoritmo)		Privada	
Limitações na Arquitetura de Blockchain		Finalidade da abordagem	
Mecanismo de consenso		Verificação de blockchain para IoT	
Tipo de aplicação			
Geral			

Figura 4.26: Formulário de extração de dados - LeapChain: Efficient Blockchain Verification for Embedded IoT

4.3.8 ID 608: Modeling and execution of blockchain-aware business processes

Trabalho desenvolvido pelos autores Ghareeb Falazi, Michael Hahn, Uwe Breitenbücher e Frank Leymann, publicado em 2019 no *SICS Software-Intensive Cyber-Physical Systems*, foi obtido pela base de dados eletrônica da SpringerLink e propõe uma extensão para modelagem de processos de negócios que captura as particularidades das *blockchains*, levando em consideração que as linguagens para modelagem de processos de negócios existentes não fornecem um suporte ideal para modelar intuitivamente as várias interações com *blockchains*. Os autores ainda mostram como transformar as construções propostas em modelos padrão e apresentam uma arquitetura de integração que permite que aplicativos externos se comuniquem com as *blockchains*.

Os autores apresentam o BAL (Blockchain Access Layer), que fornece um acesso assíncrono, independente da tecnologia, a certas operações *blockchain*, e foi desenvolvido levando em consideração quatro quesitos:

- Deve suportar o tratamento da incerteza da *blockchain*.
- Deve desempenhar o papel de uma camada de unificação independente da tecnologia.
- Deve oferecer suporte à extensibilidade, o que significa que ele deve ser projetado de maneira a permitir que novos tipos de *blockchains* se comuniquem com ele, para que sua cobertura do domínio possa aumentar gradualmente.
- Deve fornecer uma API assíncrona

Sendo assim, o BAL permite que aplicativos externos, como mecanismos de processo, acessem suas operações por meio de uma API assíncrona e não os force a bloquear a espera pelo resultado.

Ainda no estudo é apresentado o método *Blockchain-aware Modeling and Execution* (BlockME), este fornece uma visão abrangente da abordagem proposta. Primeiro, uma linguagem de modelagem de processo que suporta as extensões propostas, como *Business Process Model and Notation* (BPMN), é usada para especificar um modelo de processo com construções de modelagem compatíveis com *blockchain* como por exemplo um modelo *BlockMEprocess*. E

como os autores não pretendem desenvolver uma nova linguagem de modelagem de processos ou um mecanismo de processo especial compatível com *blockchain*, na segunda etapa, é feita a transformação dos modelos *BlockMEprocess* em modelos de processos executáveis e compatíveis com alguns padrões, como o BPMN 2.0 ou *Business Process Execution Language* (BPEL).

A abordagem é avaliada por meio de uma implementação prototípica para comprovar sua viabilidade prática. A Figura 4.27 mostra a arquitetura do sistema proposto para suportar os modelos BlockME. A arquitetura é dividida em três camadas principais: Camada Blockchain, camada de Acesso Blockchain e Camada de processo compatível com Blockchain. A camada Blockchain representa o conjunto de redes de *blockchain* que o sistema pretende oferecer suporte. A figura mostra duas dessas redes, a rede Bitcoin e a rede Ethereum. Para se comunicar com uma rede *blockchain*, um aplicativo precisa ter acesso a um dos nós da rede. Um nó *blockchain* é um processo que executa o protocolo ponto a ponto específico desse *blockchain*. Um nó *blockchain* geralmente é executado no mesmo ambiente que o aplicativo de acesso. Como parte da *Blockchain Access Layer*, adaptadores com implementações específicas da tecnologia são introduzidos para unificar o acesso aos nós da *blockchain*. O BAL também tem a responsabilidade de gerenciar assinaturas, armazenando identificadores de assinatura e *end-points* de retorno de chamada e usando-os para direcionar corretamente as mensagens de resposta. Finalmente, a terceira camada da arquitetura é a Camada de Processo compatível com Blockchain, responsável pela execução dos modelos BlockME.

O formulário de extração de dados deste artigo pode ser visto na Figura 4.28

4.3.9 ID 424: Open Peer-to-Peer Systems over Blockchain and IPFS: an Agent Oriented Framework

Trabalho desenvolvido pelos autores Antonio Tenorio-Fornés, Samer Hassan e Juan Pavón, publicado em 2018 no *CryBlock'18: Proceedings of the 1st Workshop on Cryptocurrencies and Blockchains for Distributed Systems*, foi obtido pela base de dados eletrônica da ACM Digital Library e trabalha uma abordagem orientada a agentes.

O trabalho apresenta um *framework* para construir sistemas abertos *peer-to-peer* em forma de sistemas multi-agentes, permitindo acesso aos dados, descoberta de dados e confiança em dados em uma infraestrutura descentralizada. Os autores propõem diretrizes de projeto para avaliar se uma ferramenta de coordenação é necessária (baseado no teorema CAP e principio

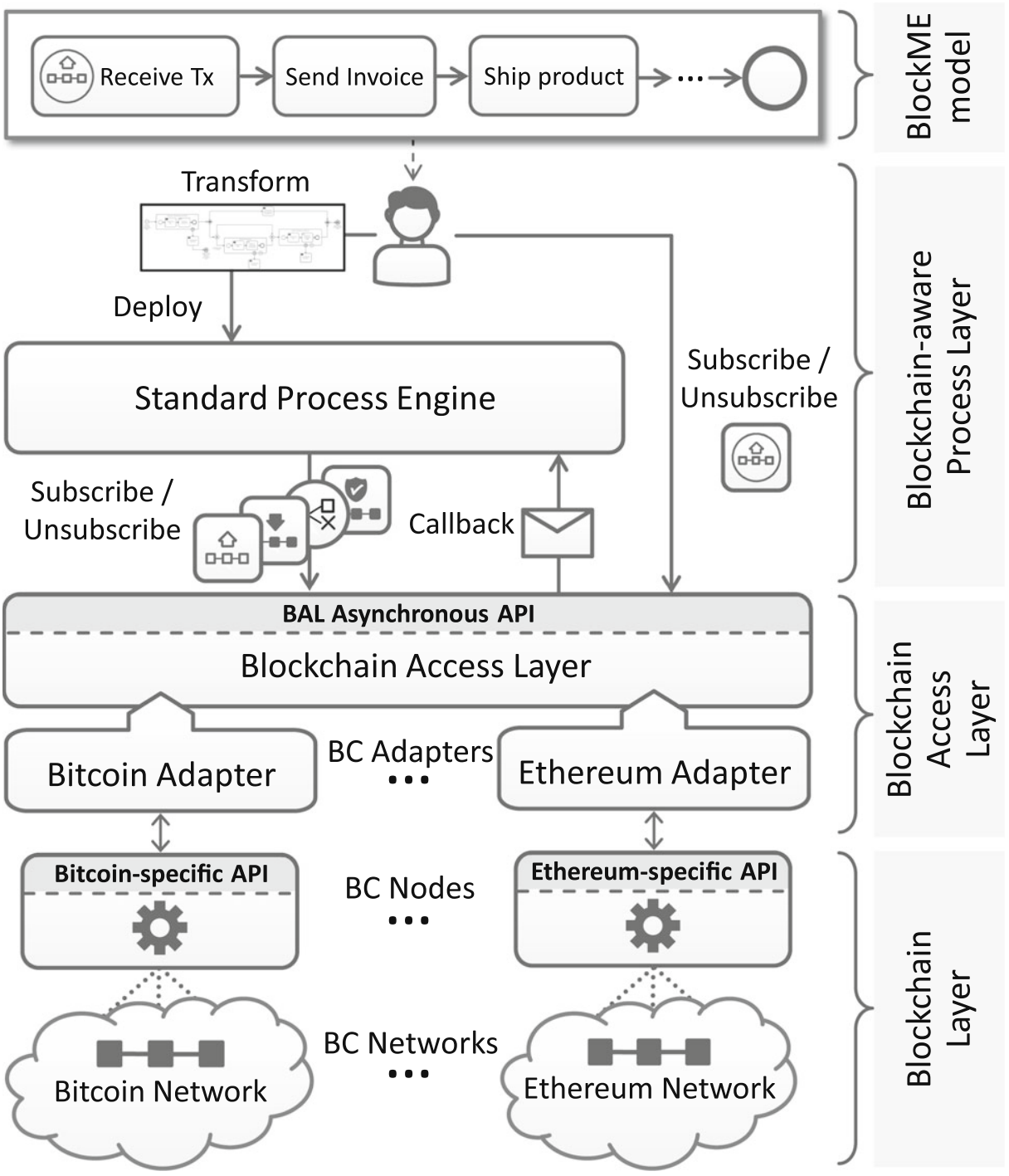

Figura 4.27: Arquitetura do sistema BlockME mostrando suas três principais camadas (FALAZI et al., 2019)

CALM) para fornecer consistência forte no sistema aberto distribuído e propõe o uso de *blockchain* para tais casos. Há ainda a proposta da arquitetura distribuída para a implementação de tais sistemas, tal arquitetura utiliza a IPFS e suas estruturas *merkle linked* para representação

ID	608	**Base de dados**	SpringerLink
Título	Modeling and execution of blockchain-aware business processes		
Autores	Ghareeb Falazi, Michael Hahn, Uwe Breitenbücher, Frank Leymann		
Palavras-chave	Business process management, Blockchain technology, Blockchain-aware business processes		
Resumo	The blockchain is an emerging technology that allows multiple parties to agree on a common state without the need for trusted intermediaries. Moreover, business process technology streamlines the automation of inter- and intra-organizational processes while cutting-down on costs. With the new business opportunities provided by blockchains, it becomes vital to combine both technologies to allow the modeling and execution of blockchain-based interactions within business processes. However, the existing business process modeling languages lack support to intuitively model the various interactions with blockchains. In this paper we address this issue by proposing a business process modeling extension that captures the particularities of blockchains. We also show how to transform the proposed constructs into standard-compliant models, and we present an integration architecture that allows external applications, to communicate with the blockchains. Finally, we validate our approach by providing a prototypical implementation that proves its practical feasibility		
Questões de pesquisa			
QP1: Como tem sido construídas as abordagens de desenvolvimento de BOS?			
"we address this issue by proposing a business process modeling extension that captures the particularities of blockchains. We also show how to transform the proposed constructs into standard-compliant models, and we present an integration architecture that allows external applications, to communicate with the blockchains." " We proposed an extension to BPMN that captures the semantics of blockchain-based systems and assists modeling fine-grained decisions when handling the uncertainty of blockchain transactions. We further showed how to convert each of the proposed modeling extensions into standard-compliant BPMN 2.0 process fragments. Moreover, we designed the Blockchain Access Layer, an integration middleware that allows external applications to communicate with public blockchain systems while taking care of blockchain specificities."			
QP2: Quais os pontos críticos e/ou desafios dessas abordagens?			
Não tem.			
QP3: Como estas abordagens foram validadas?			
Finally, we validate our approach by providing a prototypical implementation that proves its practical feasibility.			
QP4: Quais os problemas têm sido apontados pelas organizações de software que adotam as abordagens para desenvolvimento de BOS?			
However, the existing business process modeling languages lack support to intuitively model the various interactions with blockchains			
QP5: Quais são as tecnologias usadas para prover gestão e desenvolvimento das abordagens para BOS?			
"we enable the transformation of the aforementioned BlockMEprocess models into standard-compliant and executable process models, such as BPMN 2.0 or Business Process Execution Language (BPEL)"			
Facetas			
Tipo do estudo		Tecnologias/Ferramentas	
Proposta de Solução		BPMN, BPMN 2.0, BPEL	
Tipo de contribuição		Classificação e principais características das redes blockchain	
Extensão para BPMN		Pública	
Limitações na Arquitetura de Blockchain		Finalidade da abordagem	
Não tem.		Extensão para modelagem de processos de negócios que captura as particularidades das *blockchains*	
Tipo de aplicação			
Propósito Geral			

Figura 4.28: Formulário de extração de dados - Modeling and execution of blockchain-aware business processes

e distribuição de dados, criptografia de chave pública para fornecer confiança aos dados distribuídos, e a tecnologia Ethereum Blockchain é proposta como ferramenta de coordenação para suportar os requisitos de consistência não monotônica dos sistemas. Os autores apresentam como exemplo de implementação um simples sistema de Perguntas e Respostas (ver Figura 4.29).

O *framework* ainda propõe o uso de um protocolo de comunicação de consulta que possi-

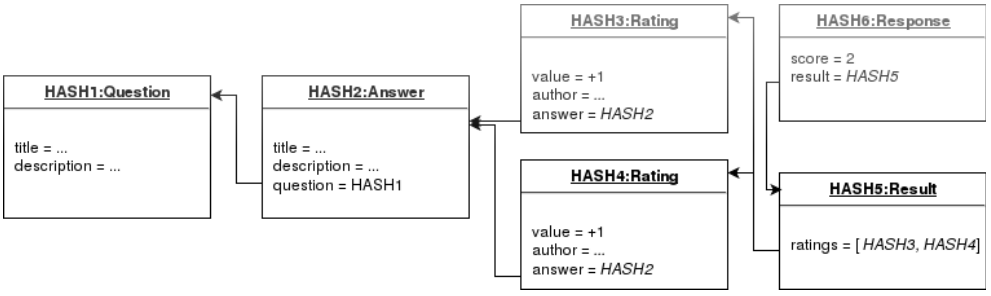

Figura 4.29: Informações Merkle linked em um exemplo de sistema de perguntas e respostas (TENORIO-FORNÉS; HASSAN; PAVÓN, 2018)

bilita a descoberta de dados em sistemas distribuídos abertos e suporta respostas classificadas e verificação sem confiança das respostas. Tal protocolo propõe as definições de consultas como restrições a serem satisfeitas pelas respostas dos dados. As interações do protocolo podem ser vistas na Figura 4.30. O protocolo, conforme descrito pelos autores, tem as seguintes vantagens: Comunicação leve, Comparação/verificação distribuída antecipadas, Classificação e validade sem confiança.

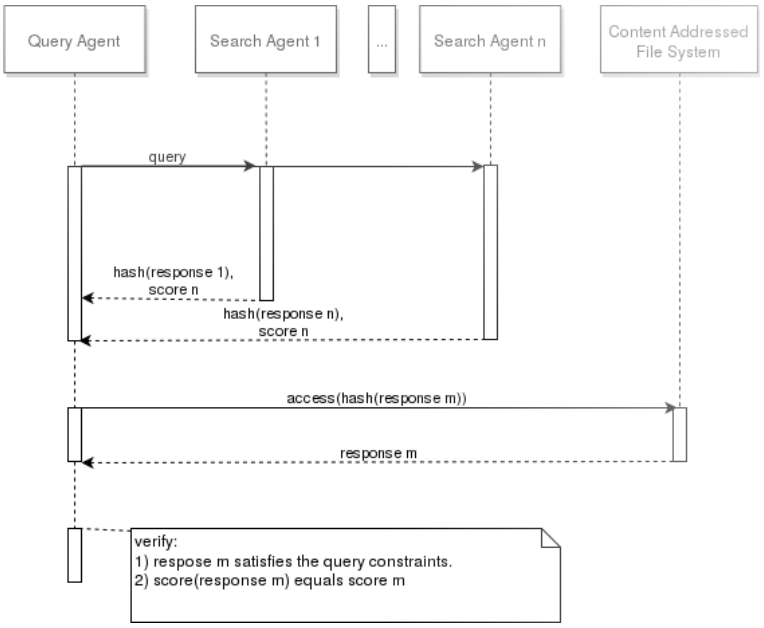

Figura 4.30: Diagrama de sequencia UML do Protocolo de descoberta distribuído (TENORIO-FORNÉS; HASSAN; PAVÓN, 2018)

Como já apontado pelos próprios autores, o *framework* proposto herda os desafios e limitações da tecnologia Blockchain, como privacidade e sustentabilidade. Além disso, alguns problemas de segurança, como ataques *sybil* e ataque de primeira geração.

O formulário de extração de dados deste artigo pode ser visto na Figura 4.31

ID	424	**Base de dados**	Scopus
Título	Open Peer-to-Peer Systems over Blockchain and IPFS: an Agent Oriented Framework		
Autores	Antonio Tenorio-Fornés, Samer Hassan e Juan Pavón		
Palavras-chave	Blockchain, Decentralization, Distributed Systems, Framework, IPFS, Multi-Agent Systems, P2P Systems		
Resumo	In recent years, the increasing concerns around the centralized cloud web services (e.g. privacy, governance, surveillance, security) have triggered the emergence of new distributed technologies, such as IPFS or the Blockchain. These innovations have tackled technical challenges that were unresolved until their appearance. Existing models of peer-to-peer systems need a revision to cover the spectrum of potential systems that can be now implemented as peer-to-peer systems. This work presents a framework to build these systems. It uses an agent-oriented approach in an open environment where agents have only partial information of the system data. The proposal covers data access, data discovery and data trust in peer-to-peer systems where different actors may interact. Moreover, the framework proposes a distributed architecture for these open systems, and provides guidelines to decide in which cases Blockchain technology may be required, or when other technologies may be sufficient.		
Questões de pesquisa			
QP1: Como tem sido construídas as abordagens de desenvolvimento de BOS?			
The proposed implementation of the protocol relies in: IPFS merkle-linked objects to represent the data and provide the responses. Javascript pure functions to express query constraints and score functions, using the JavaScript implementation of IPFS, and a bus model for distributed systems communication over IPFS pub-sub channels.			
QP2: Quais os pontos críticos e/ou desafios dessas abordagens?			
The proposal inherits the challenges and limitations of Blockchainbased and distributed technology such as privacy and sustainability. Moreover, some security issues such as sybil attacks and generation attacks. The performance and efficiency of the proposed framework remains to be studied in future work. The deployment of specialized agents, such as search agents for specific applications, or the proposal of improved network topologies and protocols are some of the performance improvement opportunities to explore.			
QP3: Como estas abordagens foram validadas?			
The architecture of the proposed framework is presented with an example of the implementation of a simple Questions and Answers (Q&A) system, similar to the popular Stack Exchange and its most famous instance Stack Overflow.			
QP4: Quais os problemas têm sido apontados pelas organizações de software que adotam as abordagens para desenvolvimento de BOS?			
Não tem.			
QP5: Quais são as tecnologias usadas para prover gestão e desenvolvimento das abordagens para BOS?			
"In order to enable an easier integration with other parts of the framework, the architecture suggest the use of IPNS or Ethereum identity infrastructure". "The proposed implementation of the protocol relies in: IPFS merkle-linked objects to represent the data and provide the responses. Javascript pure functions to express query constraints and score functions, using the JavaScript implementation of IPFS, and a bus model for distributed systems communication over IPFS pub-sub channels". "Consistency As Logical Monotonicity (CALM) principle provides a tool to describe which queries can be resolved in a distributed system without coordination". "The architecture uses IPFS as a distributed data store, public-key identities for data trust, and a generic P2P network for communication". "This work presents a framework to build peer-to-peer open systems as a multi-agent systems".			
Facetas			
Tipo do estudo		Tecnologias/Ferramentas	
Exemplo (Toy example).		Multi-agentes autônomos, Principio CALM, IPFS, Chave-pública, IPNS, Ethereum.	
Tipo de contribuição		Classificação e principais características das redes blockchain	
Framework, guidelines.		Publica.	
Limitações na Arquitetura de Blockchain		Finalidade da abordagem	
Privacidade, sustentabilidade, segurança, performance e eficiência.		Sistemas distribuídos orientados a Blockchain.	
Tipo de aplicação			
Open systems (www, sistemas operacionais, IPFS).			

Figura 4.31: Formulário de extração de dados - Open Peer-to-Peer Systems over Blockchain and IPFS: an Agent Oriented Framework

4.3.10 ID 822: Reducing the Execution Time of Unit Tests of Smart Contracts in Blockchain Platforms

Trabalho desenvolvido pelos autores Hallan Medeiros, Patrícia Vilain e Vilmar César Pereira Júnior, publicado no *SBSI'19 Proceedings of the XV Brazilian Symposium on Information Systems*, foi obtido pela base de dados eletrônica da ACM Digital Library e propõe uma abordagem para reutilizar a implantação e a execução da configuração do teste de unidade em contratos inteligentes para reduzir o tempo de execução desses testes sem violar o princípio da independência de teste. A Figura 4.32 ilustra como acontece este reuso.

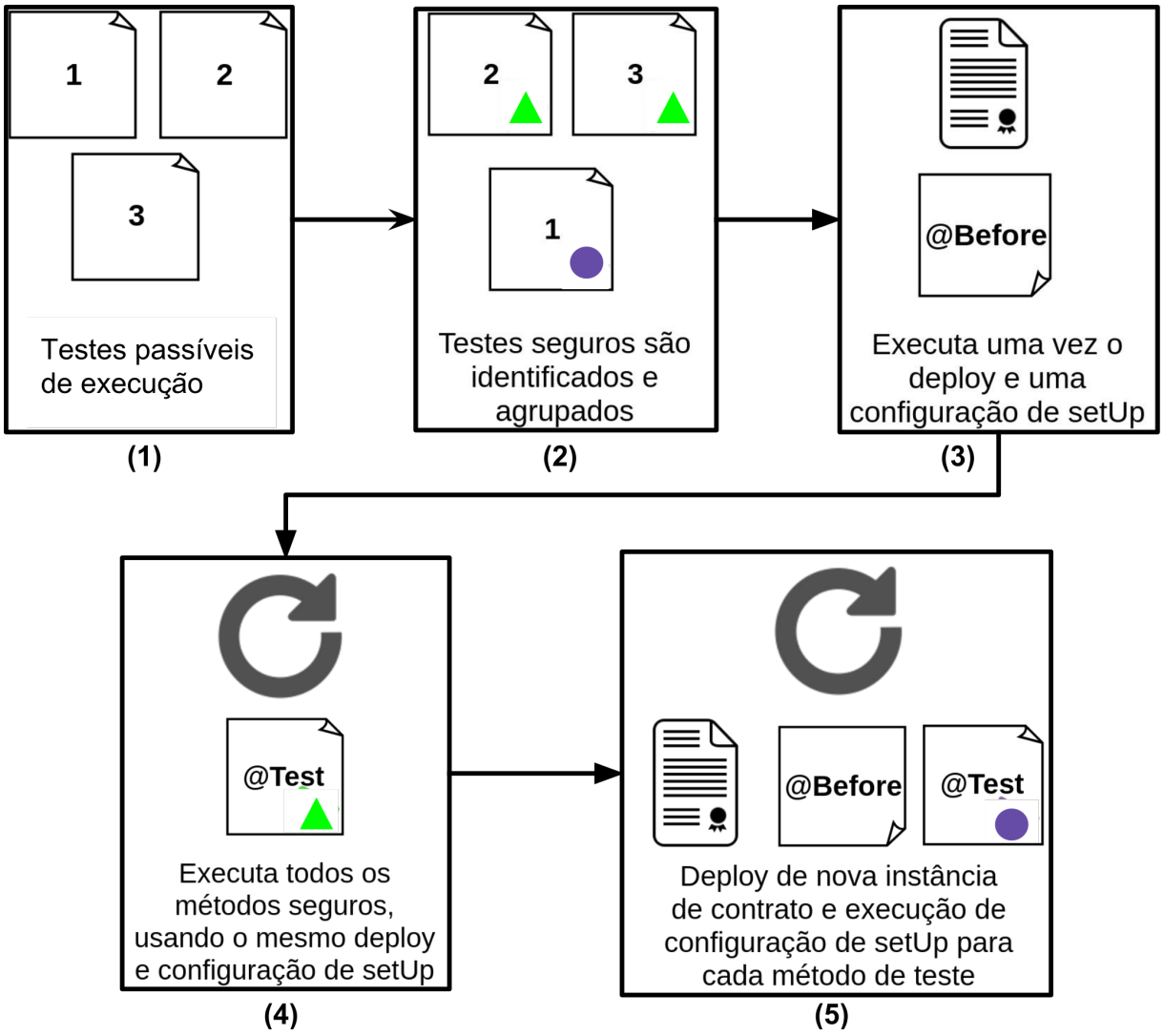

Figura 4.32: Visão geral da proposta de reuso de implantação e de execução de configuração de testes (MEDEIROS; VILAIN; PEREIRA JÚNIOR, 2019)

Para atingir o objetivo os autores apontam a execução de 5 etapas (MEDEIROS; VILAIN; PEREIRA JÚNIOR, 2019):

1 Escrita da lista de testes de unidade a serem executados;

2 Classificação dos testes de acordo com 2 grupos: testes seguros (triângulo) e testes não-seguros (círculo);

3 Execução da implantação e da configuração dos testes uma única vez;

4 Execução de todos os métodos correspondentes aos testes classificados como seguros na mesma instância de contrato, suprimindo a execução da implantação e da configuração de testes antes de cada um destes testes;

5 Execução de todos os métodos correspondentes aos testes classificados como não-seguros, executando novamente a implantação e a configuração de testes uma vez antes da execução de cada um destes testes.

Para dar suporte à proposta do trabalho os autores construíram um *framework* de testes em Java, como uma extensão do JUnit, de forma que, segundo eles, o fluxo básico de execução do ciclo de testes seja o mesmo proposto pelo JUnit. A proposta também utiliza o projeto Web3j para a criação de classes Java que representam contratos Solidity. A Figura 4.33 ilustra esta representação (MEDEIROS; VILAIN; PEREIRA JÚNIOR, 2019).

Para validar o *framework* desenvolvido, os autores realizaram um experimento com um contrato inteligente que implementa um processo de votação. Neste contrato, é possível cadastrar e visualizar propostas que serão votadas, e votar se é a favor ou contra cada proposta. Os resultados obtidos durante o experimento, segundo os autores, mostram que é possível reduzir o tempo de execução dos testes de contratos inteligentes, cerca de até 66% do tempo de execução dos testes de unidade.

O formulário de extração de dados deste artigo pode ser visto na Figura 4.34

```
1  public class Democracy extends Contract {
2
3      protected Democracia(){}
4
5      public RemoteCall<TransactionReceipt> criarProposta(
6              String titulo, String descricao) {}
7
8      public RemoteCall<TransactionReceipt> votar(
9                      BigInteger id, BigInteger voto ){}
10
11     public static RemoteCall<Democracia> deploy(){}
12
13     public static Democracia load(String address){}
14 }
```

Figura 4.33: Código de um contrato inteligente representado através de uma classe Java (MEDEIROS; VILAIN; PEREIRA JÚNIOR, 2019)

ID	822	**Base de dados**	ACM Digital Library
Título	Reducing the Execution Time of Unit Tests of Smart Contracts in Blockchain Platforms		
Autores	Medeiros, Hallan and Vilain, Patrícia and Pereira, Júnior, Vilmar César		
Palavras-chave	Smart Contracts Testing, Test Automation		
Resumo	Smart Contracts are software code that resides within a blockchain, using its infrastructure as an advantage and guarantee of execution. Blockchain and smart contracts are enabling new business models and standards to information systems. However, a smart contract needs to be well tested before to be published in a blockchain, since it cannot be changed after being deployed. The execution time to deploy smart contracts and run their tests is considerable because all transactions must be mined before being added to a new block. This work proposes an approach to reuse the execution of the deployment and the setup of unit test in smart contracts to reduce the execution time of these tests. Experiments have shown a large reduction in the execution time of smart contract unit tests, without breaking the principle of test independency.		
Questões de pesquisa			
QP1: Como tem sido construídas as abordagens de desenvolvimento de BOS?			
"Este trabalho propõe a reutilização da implantação e da execução da configuração de testes de contratos inteligentes, sem que este reuso possa deixar um teste inconsistente", "A proposta consiste nas seguintes etapas: (1) Escrita da lista de testes de unidade a serem executados; (2) Classificação dos testes de acordo com 2 grupos: testes seguros (triângulo) e testes não-seguros (círculo); (3) Execução da implantação e da configuração dos testes uma única vez; (4) Execução de todos os métodos correspondentes aos testes classificados como seguros na mesma instância de contrato, suprimindo a execução da implantação e da configuração de testes antes de cada um destes testes; (5) Execução de todos os métodos correspondentes aos testes classificados como não-seguros, executando novamente a implantação e a configuração de testes uma vez antes da execução de cada um destes testes"			
QP2: Quais os pontos críticos e/ou desafios dessas abordagens?			
Como todo estudo experimental, existem situações que podem ameaçar a validade dos resultados obtidos. Neste trabalho, o fato de haver apenas um sujeito nos experimentos representa uma ameaça à generalização da proposta. Outra ameaça consiste na escassez de bons repositórios públicos contendo contratos e testes de contratos para utilizar nos experimentos			
QP3: Como estas abordagens foram validadas?			
O experimento foi realizado com um contrato inteligente que implementa um processo de votação, a fim de validar o framework desenvolvido. Neste contrato, é possível cadastrar e visualizar propostas que serão votadas, e votar se é a favor ou contra cada proposta			
QP4: Quais os problemas têm sido apontados pelas organizações de software que adotam as abordagens para desenvolvimento de BOS?			
Não tem.			
QP5: Quais são as tecnologias usadas para prover gestão e desenvolvimento das abordagens para BOS?			
"Esta ferramenta foi implementada como uma extensão do framework JUnit, na linguagem de programação Java", " A proposta também utiliza o projeto Web3j para a criação de classes Java que representam contratos Solidity"			
Facetas			
Tipo do estudo		Tecnologias/Ferramentas	
Proposta de Solução / Simulação Experimental		Java, JUnit, Ethereum, Solidity, Web3j	
Tipo de contribuição		Classificação e principais características das redes blockchain	
Abordagem		Pública	
Limitações na Arquitetura de Blockchain		Finalidade da abordagem	
Não tem.		Extenção do JUnit. Testes unitários	
Tipo de aplicação			
Testes unitários			

Figura 4.34: Formulário de extração de dados - Reducing the Execution Time of Unit Tests of Smart Contracts in Blockchain Platforms

Capítulo 5

Análise dos Resultados

A terceira e última fase do processo de um mapeamento sistemático é publicação dos resultados obtidos (ver Figura 3.1), nela é feita uma análise dos resultados a partir da extração de dados dos estudos selecionados no Capítulo 4. Primeiramente será apresentada uma visão geral dos artigos selecionados na Seção 5.1. Uma análise sobre os aspectos das abordagens identificadas é visto na Seção 5.2. Seção 5.3 apresenta uma análise sobre os pontos críticos das abordagens, na Seção 5.4 é apresentada uma análise e pequenos comentários sobre como foram validadas as abordagens. Uma análise dos problemas apontados pode ser vista na Seção 5.5 e por fim na Seção 5.6 é apresentada a análise das tecnologias utilizadas.

5.1 Visão Geral

Após a finalização da fase de execução do MS (ver Capítulo 4), foram selecionados e extraídos dados de 10 trabalhos que descrevem abordagens para desenvolvimento de BOS. Antes de apresentar uma análise mais detalhada, será apresentado uma visão geral dos estudos selecionados. A Tabela 5.1 apresenta uma síntese dos resultados para auxiliar a análise dessa Seção. Dos 10 estudos selecionados, 4 estudos são oriundos da base de dados eletrônica da Scopus, 4 da ACM Digital Library e 2 da SpringerLink. Pode-se observar que 5 estudos foram publicados em revistas, 3 em conferências e 2 são artigos de *workshop*.

Os trabalhos mais antigos foram (NEISSE; STERI; NAI-FOVINO, 2017) (ID 541) e (BARTOLETTI et al., 2017) (ID 749), todos publicado em 2017. Os trabalhos mais recentes foram publicados em 2019 (IDs 173, 628, 608 e 822). Note que são apenas 3 anos de estudos sobre o tema pesquisado.

Tabela 5.1: Síntese de visão geral dos estudos selecionados

ID	Base de dados	Tipo	Ano
541	Scopus	*Conference*	2017
749	ACM Digital Library	*Workshop*	2017
467	Scopus	*Journal*	2018
489	Scopus	*Journal*	2018
173	Scopus	*Journal*	2019
628	SpringerLink	*Journal*	2019
778	ACM Digital Library	*Conference*	2018
608	SpringerLink	*Journal*	2019
424	ACM Digital Library	*Workshop*	2018
822	ACM Digital Library	*Conference*	2019

A Tabela 5.2 apresenta alguns dados dos autores dos estudos suas instituições e países dessas instituições. No geral, o conjunto de estudos são em sua grande maioria de pesquisadores europeus que estiveram envolvidos em 9 dos estudos. Destaque para a *Technische Universitat München*, a qual teve dois estudos de seus pesquisadores incluídos nesse trabalho. Pesquisadores italianos e alemães se destacam por possuírem três pesquisas.

5.2 Análise dos Aspectos das Abordagens

Diversas abordagens diferentes para o desenvolvimento de BOS foram encontras nesse mapeamento sistemático, entretanto muito menos do que era esperado, assim como descrito por (BOSU et al., 2018), apesar da existência de um grande número de projetos de BOS ativos, bem como um enorme interesse de desenvolvedores na tecnologia *blockchain*, houve poucas pesquisas científicas explorando a área da engenharia de software. Durante a execução do protocolo (Ver Capítulo 4), foi constatado que, pela tecnologia *blockchain* ser relativamente nova, grande parte dos estudos estão voltados para o aprimoramento de sua infra-estrutura e seus mecanismos de consenso e estes não foram considerados nesse mapeamento.

Nos estudos selecionados foram levantados alguns aspectos que auxiliaram na construção da resposta da primeira questão de pesquisa deste trabalho. Os autores do trabalho ID 541 utilizam a plataforma *Ethereum Virtual Machine* para avaliar o design, implementação e desempenho dos modelos propostos e ainda utilizam de uma abordagem de modelagem de dados usada no Kit de Ferra-mentas de Segurança Baseado em Modelo (SecKit). No estudo ID 467 os

Tabela 5.2: Dados gerais dos autores

ID	Autor	Instituição	País
541	Ricardo Neisse	*European Commission Joint Research Centre (JRC)*	Itália
541	Gary Steri	*European Commission Joint Research Centre (JRC)*	Itália
541	Igor Nai-Fovino	*European Commission Joint Research Centre (JRC)*	Itália
749	Massimo Bartoletti	*University of Cagliari*	Itália
749	Stefano Lande	*University of Cagliari*	Itália
749	Livio Pompianu	*University of Cagliari*	Itália
749	Andrea Bracciali	*University of Stirling*	Reino Unido
467	Matevž Pustišek	*University of Ljubljana*	Eslovénia
467	Andrej Kos	*University of Ljubljana*	Eslovénia
489	Guido Perboli	*DAUIN, Politecnico di Torino*	Itália
489	Stefano Musso	*DAUIN, Politecnico di Torino*	Itália
489	Mariangela Rosano	*DAUIN, Politecnico di Torino*	Itália
173	Fabian Knirsch	*Salzburg University of Applied Sciences*	Áustria
173	Andreas Unterweger	*Salzburg University of Applied Sciences*	Áustria
173	Dominik Engel	*Salzburg University of Applied Sciences*	Áustria
628	Diego Marmsoler	*Technische Universitat München*	Alemanha
628	Habtom Kashay Gidey	*Technische Universitat München*	Alemanha
778	Emanuel Regnath	*Technische Universität München*	Alemanha
778	Sebastian Steinhorst	*Technische Universität München*	Alemanha
608	Ghareeb Falazi	*University of Stuttgart*	Alemanha
608	Michael Hahn	*University of Stuttgart*	Alemanha
608	Uwe Breitenbücher	*University of Stuttgart*	Alemanha
608	Frank Leymann	*University of Stuttgart*	Alemanha
424	Antonio Tenorio-Fornés	*GRASIA, Universidad Complutense de Madrid*	Espanha
424	Samer Hassan	*GRASIA, Universidad Complutense de Madrid*	Espanha
424	Juan Pavón	*GRASIA, Universidad Complutense de Madrid*	Espanha
822	Hallan Medeiros	Universidade Federal de Santa Catarina	Brasil
822	Patrícia Vilain	Universidade Federal de Santa Catarina	Brasil
822	Vilmar César Pereira Júnior	Universidade Federal de Santa Catarina	Brasil

autores utilizam como base a plataforma *Ethereum* para elaborar e comparar as abordagens arquiteturais para o design dos aplicativos de dispositivos IoT front-end. No estudo ID 489

foi proposto a utilização da metodologia GUEST para auxiliar no design de um BOS. Já no estudo ID 173 os autores avaliam as plataformas *blockchain* existentes e verificam que para existem alguns problemas em relação a elas como escalabilidade, decidem pro criar do zero uma implementação de *blockchain*. No trabalho ID 608 é proposto uma extensão ao BPMN que captura as particularidades das *blockchain*. E por fim no trabalho ID 822 é proposto uma abordagem para reutilizar a implantação e a execução da configuração do teste unitários em contratos inteligentes para reduzir o tempo de execução. Diante desses aspectos levantados a partir dos formulários de extração de dados, uma síntese da resposta para a QP1: *"Como tem sido construídas as abordagens de desenvolvimento de BOS?"*, é a seguinte:

É possível perceber que a utilização de plataformas *blockchain* como Ethereum vêm sendo utilizada para desenvolver certas abordagens, isso se dá ao fato delas já disponibilizarem toda a parte de infraestrutura necessária para a implantação de uma *blockchain* e também como apresentado por (BOSU et al., 2018), possuem uma grande comunidade de desenvolvedores ao redor do mundo, o que facilita o compartilhamento de experiencias, porém como mostrado por (KNIRSCH; UNTERWEGER; ENGEL, 2019) é possível que para objetivos específicos deve-se criar do zero uma *blockchain*, e para isso a metodologia GUEST já se mostrou útil como apontado por (PERBOLI; MUSSO; ROSANO, 2018) e o uso do Kit de Ferramentas de Segurança Baseado em Modelo (SecKit) apresentado por (NEISSE; STERI; NAI-FOVINO, 2017), já para a modelagem de processos uma extensão para BPMN já se mostrou eficaz como relata (FALAZI et al., 2019) e o interesse nessa notação se da ao fato de que no contexto da modelagem de negócio, o modelo considerado padrão atualmente é o BPMN como dito por (HARMON, 2010) e ainda para testes unitários, a metodologia desenvolvida por (MEDEIROS; VILAIN; PEREIRA JÚNIOR, 2019) mostrou uma redução no tempo de execução destes testes de até 66%.

5.3 Análise de Pontos Críticos, Falhas e Desafios

Diversos pontos críticos foram encontrados nos estudos selecionados. Esses pontos devem ter uma atenção especial do usuário na hora de escolher alguma abordagem a ser utilizada. Ainda houve a identificação de falhas e alguns desafios a serem superados, quase todos relaci-

onados aos atributos de qualidade (escalabilidade, segurança, disponibilidade, compreensibilidade, etc) de uma aplicação BOS.

Os autores do trabalho ID 541 relatam que a adoção dos modelos lá citados afetam a privacidade, o anonimato, o desempenho e a escalabilidade sendo que do ponto de vista da escalabilidade, a abordagem genérica do controlador é a melhor, mas também restringe uma possível solução para *blockchains* públicas devido ao alto número de transações, ou seja a escalabilidade é uma de suas limitações a serem vistas com maior atenção pelo usuário. No estudo ID 749 análises que abordam propagação de informações e *forks* não foram cobertas. Os autores do estudo ID 467 relatam que arquitetura das partes *front-end* dos aplicativos depende muito dos recursos e limitações dos dispositivos de IoT onde serão implantados. Já os autores do trabalho ID 173 argumentam que a escalabilidade pode ser aprimorada quando o número de usuários ainda é pequeno, mas que isso ainda é um desafio a ser superado. No trabalho ID 628 é relatado que os usuários precisam ter alguma experiência em ITP (Prova interativa de teoremas) para usar eficientemente a abordagem descrita. No estudo ID 424 é relatado limitações como privacidade e sustentabilidade e ainda problemas com ataques de geração 5. Com isso foi possível obter uma síntese para a resposta da QP2: *"Quais os pontos críticos e/ou desafios dessas abordagens?"*, sendo esta a seguinte:

A escalabilidade é um ponto crítico, relatado por (NEISSE; STERI; NAI-FOVINO, 2017) e (KNIRSCH; UNTERWEGER; ENGEL, 2019), e também já é um dos problemas apontados na revisão sistemática de (YLI-HUUMO et al., 2016), ou seja, é um problema recorrente e que ainda precisa ser resolvido, assim deve ser visto com atenção, há ainda deficiência em análises abordando propagação de informações e *forks* como dito por (BARTOLETTI et al., 2017). Existe outro ponto crítico que deve ser levado em consideração durante a implantação de um BOS, as limitações de hardware e software do dispositivo hospedeiro, como descrito por (PUSTIŠEK; KOS, 2018), há ainda o desafio de superar o não conhecimento de usuários para o entendimento de algumas abordagens como descrito no trabalho de (MARMSOLER; GIDEY, 2019). Por fim, existem falhas como sustentabilidade, prevenção de ataques e privacidade como relatado por (TENORIO-FORNÉS; HASSAN; PAVÓN, 2018), sendo que a privacidade é um atributo essencial no ambiente *blockchain*, devido à sua característica de anonimato e sendo um

fator necessário para evitar ataques e tentativas de perturbar transações no *blockchain* (YLI-HUUMO et al., 2016).

5.4 Análise de como foram validadas as abordagens

Durante a análise dos estudos incluídos nesse mapeamento, diferentes métodos de avaliação/validação dos estudos foram adotados, dentre esses o que mais se destaca é a implementação e desenvolvimento prático da proposta. Os autores do estudo ID 541 implementaram um contrato de amostra para validação dos modelos propostos, assim como os autores do trabalho ID 467 que executaram o cliente completo em um sistema embarcado RPi v3B. No estudo ID 489 os autores também implementaram a solução proposta. No estudo ID 608 foi feita a implementação prototípica para comprovar a viabilidade prática da proposta, o mesmo feito pelos autores dos estudos ID 778, 424 e 822. Houve ainda a validação por meio de casos de uso como nos estudos ID 749, ID 628 e ainda no estudo ID 173 a solução foi implementada e posta em uso em dois locais reais. Sabendo disso a síntese para a resposta da QP3: *"Como estas abordagens foram validadas?"*, fica da seguinte forma:

É unanime a decisão dos autores por uma implementação prática das soluções proposta para validação das mesmas, e como descrito por (SHULL; SINGER; SJØBERG, 2007), essa metodologia de validação permite a identificação de falhas e possíveis melhorias de forma a aperfeiçoar a abordagem para seu objetivo especifico.

5.5 Análise dos problemas apontados pelas organizações de software

Apenas 3 estudos daqueles incluídos nesse mapeamento descreveram problemas nas abordagens para desenvolvimento de BOS. No estudo ID 541 (NEISSE; STERI; NAI-FOVINO, 2017) os autores apontam um problema recente em relação à proteção de dados onde uma recente aprovação do Regulamento Geral de Proteção de Dados (GDPR) impôs novos requisitos de proteção de dados de residentes da União Europeia (UE). No estudo ID 173 os autores apresentam problemas relacionados aos atributos de qualidade já bem relatados anteriormente. E por fim no estudo ID 608 os autores relatam que as BPEL existentes não têm suporte para modelar intuitivamente as várias interações com *blockchains*.

Assim por mais que poucos estudos tenham fornecido informações sobre essa questão, é possível formular uma síntese para a resposta da QP4: *"Quais os problemas têm sido apontados pelas organizações de software que adotam as abordagens para desenvolvimento de BOS?"*, sendo ela:

Problemas relacionados aos atributos de qualidade que vêm sendo relatados pelas novas abordagens são problemas recorrentes, uma explicação plausível para isso é que, pela falta de estudos científicos voltados as soluções de tais problemas como mostrado por (BOSU et al., 2018), a resolução dos mesmos se da pelo desenvolvimento exaustivo por partes isoladas até que se alcance um solução, sem que haja um estudo empírico por trás da proposta, e ainda, problemas relacionados às notações específicas para o desenvolvimento BOS não foram propostas, apesar de extensões para notações bem conhecidas como é o caso do estudo ID 608 já terem sido desenvolvidas, uma notação própria e desenvolvida apenas para o projeto e análise de fluxos BOS ainda não existe.

5.6 Análise das tecnologias utilizadas

Diversas tecnologias diferentes foram usadas pelas abordagens incluídas neste mapeamento. O estudo ID 541 utiliza a plataforma *Ethereum Virtual Machine* e o Kit de Ferramentas de Segurança Baseado no Modelo (SecKit) destacado na Seção 5.2, para o desenvolvimento de sua abordagem. O estudo ID 749 tem seu componente principal construido utilizando a linguagem Scala e ainda utiliza ambos os modelos de banco de dados SQL e NoSQL a plataforma Ethereum e Bitcoin Core como também as bibliotecas web3j e ScalikeJDBC para conexão com os bancos. O estudo ID 467 utiliza também as plataformas Bitcoin e Ethereum, sendo a última também utilizada pelo estudo ID 424 e 822. Sendo assim é possível formular a síntese para a QP5: *"Quais são as tecnologias usadas para prover gestão e desenvolvimento das abordagens para BOS?"*, sendo ela:

As plataformas Ethereum e Bitcoin tiveram grande destaque dentre aqueles estudos incluídos neste mapeamento, sendo seu uso destacado por (NEISSE; STERI; NAI-FOVINO, 2017), (BARTOLETTI et al., 2017), (PUSTIŠEK; KOS, 2018), (TENORIO-FORNÉS; HASSAN; PAVÓN, 2018) e (MEDEIROS; VILAIN; PEREIRA JÚNIOR, 2019). Há ainda Kit de Ferramen-

tas de Segurança Baseado em Modelo (SecKit) destacado por (NEISSE; STERI; NAI-FOVINO, 2017), e ainda a utilização de modelos SQL e noSQL para armazenamento de informações.

Capítulo 6

Considerações finais

Neste trabalho de conclusão de curso foi realizado um mapeamento sistemático com o objetivo de conhecer o atual estado da arte no que diz respeito às abordagens para desenvolvimento de softwares orientados a *blockchain*, bem como fornecer um guia para outros pesquisadores e profissionais, apresentando detalhes sobre essas abordagens e seus principais conceitos e fundamentos. Apresenta-se algumas conclusões obtidas com base nesses resultados. Por fim, são propostos alguns trabalhos futuros relacionados com o tema.

6.1 Conclusões

Com base nos resultados obtidos, podemos afirmar que a engenharia de software está pouco amadurecida a respeito de abordagens para desenvolvimento de softwares orientados a *blockchain* no momento da construção deste documento, isso se justifica pela quantidade de abordagens encontradas e mapeadas. Ainda existe o fato de que, a diferença entre os anos de publicação dos estudos mais antigos e os mais recentes mapeados neste trabalho é de apenas 3 anos, portanto é necessário que mais estudos nessa área sejam realizados.

Cabe destacar ainda desafios relacionados aos atributos de qualidade, principalmente escalabilidade, sustentabilidade e privacidade podendo-se concluir, através dos estudos de (YLI-HUUMO et al., 2016), (NEISSE; STERI; NAI-FOVINO, 2017) (ID 541), (KNIRSCH; UNTERWEGER; ENGEL, 2019) (ID 173) e (TENORIO-FORNÉS; HASSAN; PAVÓN, 2018) (ID 424), que são problemas recorrentes das abordagens para desenvolvimento de BOS e que precisam ser abordados em trabalhos futuros.

Cabe destacar a utilização de plataformas *blockchain* como Ethereum por parte dos autores

dos estudos selecionados. Isso mostra que de fato essas plataformas facilitam o processo de construção de um BOS, por terem toda a infraestrutura básica necessária para a construção de uma *blockchain* como também por terem uma grande comunidade de desenvolvedores. Vale ainda destacar os esforços de (MEDEIROS; VILAIN; PEREIRA JÚNIOR, 2019), sua metodologia para a execução de testes unitários resultou em uma redução de até 66% do tempo de execução destes testes.

Por fim, este trabalho pode servir como um guia para outros pesquisadores e profissionais a identificar possíveis áreas de pesquisa, identificar lacunas e questões ou para auxiliar engenheiros de software na escolha de alguma abordagem que possa ser utilizada em seu dia a dia.

6.2 Trabalhos Futuros

Com base nesta pesquisa, podemos destacar alguns trabalhos futuros:

- Proposta de uma notação específica para o projeto e análise de fluxos BOS para que essa aborde com objetividade todos os detalhes deste tipo de software;
- Estudo comparativo entre os casos de uso gerados pelas abordagens apresentadas neste trabalho realizando uma análise da validez dos processos utilizados;
- Construção de um *framework* para testes unitários em contratos inteligentes da plataforma *blockchain* Bitcoin;
- Replicação futura do protocolo do mapeamento sistemático para encontrar novas abordagens na literatura;

Referências Bibliográficas

ALHARBY, M.; MOORSEL, A. v. Blockchain-based smart contracts: A systematic mapping study. *CoRR*, abs/1710.06372, 2017. Disponível em: <http://arxiv.org/abs/1710.06372>.

AMMOUS, S. H. Blockchain technology: What is it good for? *SSRN Electronic Journal*, 01 2016.

ARSKEY, H.; O'MALLEY, L. Scoping studies: Towards a methodological framework. *International Journal of Social Research Methodology*, v. 8, p. 19–32, 01 2005.

ATZEI, N.; BARTOLETTI, M.; CIMOLI, T. A survey of attacks on ethereum smart contracts (sok). In: . [S.l.: s.n.], 2017. p. 164–186. ISBN 978-3-662-54454-9.

BARTOLETTI, M. et al. A general framework for blockchain analytics. In: *Proceedings of the 1st Workshop on Scalable and Resilient Infrastructures for Distributed Ledgers*. New York, NY, USA: ACM, 2017. (SERIAL '17), p. 7:1–7:6. ISBN 978-1-4503-5173-7. Disponível em: <http://doi.acm.org/10.1145/3152824.3152831>.

BONNEAU, J. et al. Sok: Research perspectives and challenges for bitcoin and cryptocurrencies. In: *2015 IEEE Symposium on Security and Privacy*. [S.l.: s.n.], 2015. p. 104–121. ISSN 1081-6011.

BOOGARD, K. *Model-Driven Approach to Smart Contract Development*. Dissertação (Dissertação de Mestrado) — Business Informatics - Utrecht University, Utrecht, Paises Baixos, Junho 2018.

BOSU, A. et al. Understanding the motivations, challenges and needs of blockchain software developers: A survey. 11 2018.

BRANDÃO, A.; MAMEDE, H.; GONÇALVES, R. Systematic review of the literature, research on blockchain technology as support to the trust model proposed applied to smart places. In: ____. [S.l.: s.n.], 2018. p. 1163–1174. ISBN 978-3-319-77702-3.

BRERETON, P. et al. Lessons from applying the systematic literature review process within the software engineering domain. j syst softw. *Journal of Systems and Software*, v. 80, p. 571–583, 04 2007.

CASINO, F.; DASAKLIS, T.; PATSAKIS, C. A systematic literature review of blockchain-based applications: Current status, classification and open issues. *Telematics and Informatics*, 11 2018.

CHRISTIDIS, K.; DEVETSIKIOTIS, M. Blockchains and smart contracts for the internet of things. *IEEE Access*, v. 4, p. 2292–2303, 2016. ISSN 2169-3536.

CONTI, M. et al. A survey on security and privacy issues of bitcoin. *IEEE Communications Surveys Tutorials*, v. 20, n. 4, p. 3416–3452, Fourthquarter 2018. ISSN 1553-877X.

CROSBY, M. et al. Blockchain technology: Beyond bitcoin. *Applied Innovation Review*, Issue No. 2, p. 55–81, 2016. Disponível em: <https://j2-capital.com/wp-content/uploads/2017/11/AIR-2016-Blockchain.pdf>.

DELMOLINO, K. et al. Step by step towards creating a safe smart contract: Lessons and insights from a cryptocurrency lab. In: . [S.l.: s.n.], 2016. v. 9604, p. 79–94. ISBN 978-3-662-53356-7.

ELSEVIER. *Scopus*. 2019. Consultado na INTERNET: https://www.elsevier.com/solutions/scopus, 2019.

FALAZI, G. et al. Modeling and execution of blockchain-aware business processes. v. 34, p. 105–, 06 2019.

FALBO, R. D. A.; SOUZA, E. F. D.; FELIZARDO, K. R. Revisão sistemática da literatura em engenharia de software. In: ____. 1. ed. Rio de Janeiro, Rio de Janeiro, Brasil: Elsevier, 2017. cap. Mapeamento Sistemático, p. 93–112.

FELIZARDO, K. R. et al. *Revisão sistemática da literatura em engenharia de software*. 1. ed. Rio de Janeiro, Rio de Janeiro, Brasil: Elsevier, 2017.

HARMON, P. Bpmn for business—the role of the customer. In: . [S.l.: s.n.], 2010.

HAWLITSCHEK, F.; NOTHEISEN, B.; TEUBNER, T. The limits of trust-free systems: A literature review on blockchain technology and trust in the sharing economy. *Electronic Commerce Research and Applications*, v. 29, p. 50 – 63, 2018. ISSN 1567-4223. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1567422318300292>.

KARAFILOSKI, E.; MISHEV, A. Blockchain solutions for big data challenges: A literature review. In: *IEEE EUROCON 2017 -17th International Conference on Smart Technologies*. [S.l.: s.n.], 2017. p. 763–768.

KHALILOV, M. C. K.; LEVI, A. A survey on anonymity and privacy in bitcoin-like digital cash systems. *IEEE Communications Surveys Tutorials*, v. 20, n. 3, p. 2543–2585, thirdquarter 2018. ISSN 1553-877X.

KHAN, M. S. A. et al. Performance analysis of receiver power sensitivity of advanced modulation formats in wdm based standard mode fibre for next generation data rate. In: *2017 4th International Conference on Advances in Electrical Engineering (ICAEE)*. [S.l.: s.n.], 2017. p. 395–399. ISSN 2378-2692.

KITCHENHAM, B. et al. *Repeatability of systematic literature reviews*. [S.l.], 01 2011. v. 2011, 46-55 p.

KITCHENHAM, B.; CHARTERS, S. *Guidelines for performing Systematic Literature Reviews in Software Engineering*. [S.l.], 2007.

KNIRSCH, F.; UNTERWEGER, A.; ENGEL, D. Implementing a blockchain from scratch: why, how, and what we learned. *EURASIP Journal on Information Security*, v. 2019, 12 2019.

LAMPORT, L.; SHOSTAK, R.; PEASE, M. The byzantine generals problem. *ACM Trans. Program. Lang. Syst.*, ACM, New York, NY, USA, v. 4, n. 3, p. 382–401, jul. 1982. ISSN 0164-0925. Disponível em: <http://doi.acm.org/10.1145/357172.357176>.

LAPES Laboratório de Pesquisa em Engenharia de Software. *StArt - State of the Art through Systematic Review*. 2019. Consultado na INTERNET: http://lapes.dc.ufscar.br/tools/start_tool, 2019.

LEE, J. A. et al. *Blockchain Technology and Legal Implications of 'Crypto 2.0'*. [S.l.], 03 2015. 1-5 p.

LI, X. et al. A survey on the security of blockchain systems. *Future Generation Computer Systems*, 2017. ISSN 0167-739X. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0167739X17318332>.

MARMSOLER, D.; GIDEY, H. Interactive verification of architectural design patterns in factum. *Formal Aspects of Computing*, 07 2019.

MEDEIROS, H.; VILAIN, P.; PEREIRA JÚNIOR, V. C. Reducing the execution time of unit tests of smart contracts in blockchain platforms. In: *Proceedings of the XV Brazilian Symposium on Information Systems*. New York, NY, USA: ACM, 2019. (SBSI'19), p. 16:1–16:8. ISBN 978-1-4503-7237-4. Disponível em: <http://doi.acm.org/10.1145/3330204.3330225>.

MUKHOPADHYAY, U. et al. A brief survey of cryptocurrency systems. In: *2016 14th Annual Conference on Privacy, Security and Trust (PST)*. [S.l.: s.n.], 2016. p. 745–752.

NAKAMOTO, S. Bitcoin: A peer-to-peer electronic payment system. *Journal for General Philosophy of Science*, v. 39, n. 1, p. 53–67, 2008.

NEISSE, R. et al. Seckit: A model-based security toolkit for the internet of things. *Computers & Security*, v. 54, p. 60 – 76, 2015. ISSN 0167-4048. Secure Information Reuse and Integration & Availability, Reliability and Security 2014. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0167404815000887>.

NEISSE, R.; STERI, G.; NAI-FOVINO, I. A blockchain-based approach for data accountability and provenance tracking. In: *Proceedings of the 12th International Conference on Availability, Reliability and Security*. New York, NY, USA: ACM, 2017. (ARES '17), p. 14:1–14:10. ISBN 978-1-4503-5257-4. Disponível em: <http://doi.acm.org/10.1145/3098954.3098958>.

PERBOLI, G.; MUSSO, S.; ROSANO, M. Blockchain in logistics and supply chain: A lean approach for designing real-world use cases. *IEEE Access*, v. 6, p. 62018–62028, 2018.

PETERSEN, K. et al. Systematic mapping studies in software engineering. BCS Learning & Development Ltd., Swindon, UK, p. 68–77, 2008. Disponível em: <http: //dl.acm.org/citation.cfm?id=2227115.2227123>.

PETERSEN, K.; VAKKALANKA, S.; KUZNIARZ, L. Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology*, v. 64, 08 2015.

PORRU, S. et al. Blockchain-oriented software engineering: Challenges and new directions. 02 2017.

PUSTIŠEK, M.; KOS, A. Approaches to front-end iot application development for the ethereum blockchain. *Procedia Computer Science*, v. 129, p. 410–419, 01 2018.

REGNATH, E.; STEINHORST, S. Leapchain: efficient blockchain verification for embedded iot. In: . [S.l.: s.n.], 2018. p. 1–8.

SANKAR, L. S.; SINDHU, M.; SETHUMADHAVAN, M. Survey of consensus protocols on blockchain applications. In: *2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS)*. [S.l.: s.n.], 2017. p. 1–5.

SCHOLLMEIER, R. A definition of peer-to-peer networking for the classification of peer-to-peer architectures and applications. p. 101–102, Aug 2001.

SEEBACHER, S.; SCHÜRITZ, R. Blockchain technology as an enabler of service systems: A structured literature review. In: . [S.l.: s.n.], 2017. p. 12–23. ISBN 978-3-319-56924-6.

SHULL, F.; SINGER, J.; SJØBERG, D. I. *Guide to Advanced Empirical Software Engineering*. Berlin, Heidelberg: Springer-Verlag, 2007. ISBN 184800043X.

SILVA, I. da et al. Agile software product lines: A systematic mapping study. *Softw., Pract. Exper.*, v. 41, p. 899–920, 07 2011.

SOMMERVILLE, I. *Software engineering*. 9. ed. São Paulo, São Paulo, Brasil: Pearson Education, 2011.

SPRINGERLINK. *SpringerLink*. 2019. Consultado na INTERNET: https://media.springernature.com/full/springer-cms/rest/v1/content/15749236/data/v3, 2019.

SZABO, N. *The Idea of Smart Contracts*. 1997. Consultado na INTERNET: http://web.archive.org/web/20140406003401/szabo.best.vwh.net/idea.html, 2019.

TAMA, B. A. et al. A critical review of blockchain and its current applications. In: *2017 International Conference on Electrical Engineering and Computer Science (ICECOS)*. [S.l.: s.n.], 2017. p. 109–113.

TENORIO-FORNÉS, A.; HASSAN, S.; PAVÓN, J. Open peer-to-peer systems over blockchain and ipfs: an agent oriented framework. In: . [S.l.: s.n.], 2018. p. 19–24.

WOHLIN, C. et al. On the reliability of mapping studies in software engineering. *Journal of Systems and Software*, v. 86, n. 10, p. 2594–2610, 2013.

WRIGHT, A.; FILIPPI, P. D. Decentralized blockchain technology and the rise of lex cryptographia. *SSRN Electronic Journal*, 01 2015.

YLI-HUUMO, J. et al. Where is current research on blockchain technology? — a systematic review. *PLOS ONE*, v. 11, n. 10, p. 1–27, 2016. Disponível em: <https://doi.org/10.1371/journal.pone.0163477>.

ZHANG, J. Walks trajectory tracking of shared information based on consortium blockchain. *Revista de la Facultad de Ingenieria*, v. 31, p. 8–17, 01 2016.

ZHAO, J. L.; FAN, S.; YAN, J. Overview of business innovations and research opportunities in blockchain and introduction to the special issue. *Financial Innovation*, v. 2, 12 2016.