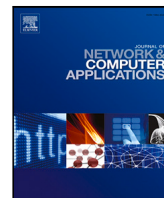


Listas de conteúdos disponíveis em [Science Direct](#)

Journal of Network and Computer Applications

Página inicial do jornal: www.elsevier.com/locate/jnca

Análise

Pesquisa de plataformas de desenvolvimento de blockchain proeminentes

Colin Connors*, Dilip Sarkar

Universidade de Miami, Departamento de Ciência da Computação, Estados Unidos da América

ARTIGO INFO

Palavras-chave:

Blockchain

Dapp

Web3

contrato inteligente

Desenvolvimento de blockchain

ABSTRATO

Muitos desenvolvedores têm ideias para criar aplicativos blockchain, mas não sabem por onde começar. Frequentemente, esses desenvolvedores usam como padrão a primeira plataforma de desenvolvimento de blockchain que descobrem, que pode não ser a melhor plataforma para seu projeto. Mais de 8.000 novos projetos relacionados a blockchain são adicionados ao GitHub por ano. O fluxo quase constante de novos projetos pode tornar difícil para os desenvolvedores pesquisar em projetos e plataformas existentes para encontrar a melhor plataforma para seus projetos. Consideramos 65 plataformas de desenvolvimento de blockchain para este trabalho e fornecemos um resumo breve, porém abrangente, das 23 plataformas mais populares. Nosso objetivo com este trabalho é auxiliar os desenvolvedores na seleção da plataforma mais adequada para seus projetos de blockchain.

1. Introdução

No artigo seminal, Bitcoin: um sistema de caixa eletrônico peer-to-peer (Nakamoto, 2008), Satoshi Nakamoto apresentou o blockchain ao mundo. Embora o sistema descrito neste artigo nunca seja explicitamente chamado de blockchain, ele possui todas as propriedades dos blockchains modernos. Blockchains, como agora os conhecemos, são registros digitais invioláveis e implementados de maneira distribuída (Yaga et al., 2018). Essa tecnologia inicialmente levou ao surgimento de criptomoedas como o Bitcoin.

No entanto, à medida que as blockchains se tornaram mais populares, os desenvolvedores começaram a expandir os casos de uso das blockchains. Essa segunda geração de blockchains permitiu que os desenvolvedores criassem contratos inteligentes, códigos que podem ser executados em um blockchain (Zou et al., 2021). Com contratos inteligentes, os desenvolvedores podem criar aplicativos descentralizados governados por um blockchain. Esses aplicativos descentralizados são chamados dApps, que se tornaram centrais no desenvolvimento Web3. O desenvolvimento Web3 depende de muitas ideias trazidas do desenvolvimento tradicional da Web, às vezes chamado de desenvolvimento Web2. Para criar um aplicativo tradicional no desenvolvimento Web2, os desenvolvedores criam e gerenciam um front-end e uma camada de back-end. A camada front-end é responsável por exibir informações para um usuário. Essa camada geralmente depende de tecnologias como HTML, CSS e JavaScript para que um navegador da Web possa renderizar uma página. A camada de back-end gerencia os dados por meio de bancos de dados, APIs e hospedagem na web. Os desenvolvedores Web2 usam as camadas front-end e back-end para criar aplicativos da Web dinâmicos.

No desenvolvimento Web3, um desenvolvedor ainda possui uma camada de front-end e back-end, além de uma nova camada de contrato inteligente. A camada de contrato inteligente permite que desenvolvedores Web3 interajam com um blockchain para criar dApps. Essa nova camada interage com o front-end e o back-end

camadas. No entanto, a camada de contrato inteligente permite que os desenvolvedores criem aplicativos que não possuem uma entidade centralizada gerenciando os dados.

No desenvolvimento Web2, os desenvolvedores têm muitas opções de plataformas de desenvolvimento para as camadas front-end e back-end, como vários frameworks JavaScript ou linguagens de script back-end. No desenvolvimento Web3, os desenvolvedores têm opções semelhantes para a camada de contrato inteligente. Criticamente, para criar um dApp, um desenvolvedor deve selecionar uma plataforma de desenvolvimento de blockchain apropriada para seu projeto.

No desenvolvimento Web2, uma das escolhas mais desafiadoras de um novo desenvolvedor é selecionar a pilha de tecnologia de front-end e back-end apropriada para um projeto. O mesmo problema existe no Web3 ao selecionar uma plataforma de desenvolvimento blockchain. Existem muitas plataformas de desenvolvimento de blockchain, com novas sendo criadas anualmente. Isso pode tornar difícil para os desenvolvedores selecionar a plataforma que melhor complementa seus projetos.

Nossa motivação para este trabalho é simplificar os processos de seleção de uma plataforma de desenvolvimento blockchain para tornar mais fácil para os desenvolvedores começarem a criar dapps. Para fazer isso, consideramos 65 plataformas de desenvolvimento de blockchain e selecionamos as 23 plataformas mais populares para analisar. Nossa análise visa destacar os diferenciais entre várias plataformas de desenvolvimento de blockchain para que os desenvolvedores possam entender qual plataforma terá melhor sinergia com seus projetos. Pesquisas anteriores, embora extensas, precisam fornecer tudo o que um desenvolvedor em potencial precisa saber. Esses trabalhos cobrem apenas algumas plataformas de desenvolvimento de blockchain (Valenta e Sandner, 2017), cobrem apenas plataformas de desenvolvimento de blockchain para uma indústria específica (Agbo e Mahmoud, 2019), cobrem apenas tipos específicos de plataformas de desenvolvimento de blockchain (Quasim et al., 2020; Polge et al., 2021; Javaid et al., 2021) ou não fornecem detalhes suficientes para selecionar o

* Autor correspondente.

Endereço de email: cdc104@miami.edu (C. Connors).
<https://doi.org/10.1016/j.jnca.2023.103650>

Recebido em 27 de outubro de 2022; Recebido no formulário revisado em 27 de fevereiro de 2023; Aceito em 19 de abril de 2023

Disponível online em 25 de abril de 2023

1084-8045/© 2023 Elsevier Ltd. Todos os direitos reservados.

plataforma mais adequada (Tavares e outros.,2019;Dernayka e Chehab, 2021). Em trabalhos anteriores (Connors e Sarkar,2022b), realizamos um estudo comparativo ao aqui apresentado mas apenas em 15 plataformas. Desde a publicação desse trabalho, expandimos o número de plataformas analisadas, coletamos dados mais recentes e selecionamos mais métricas para análise. Da mesma forma, uma versão pré-impressa deste artigo está disponível (Connors e Sarkar,2022a).

Organizamos o restante do artigo da seguinte maneira. Na seção2, fornecemos algumas informações básicas sobre blockchain, plataformas de desenvolvimento de blockchain e alguns aspectos técnicos que diferenciam blockchains. Na seção3, fornecemos alguns detalhes sobre trabalhos relacionados. Na seção4, apresentamos nossas 23 plataformas selecionadas e nossos métodos para escolher essas plataformas. Na seção5, analisamos as plataformas de desenvolvimento de blockchain selecionadas em várias métricas. Nossa análise é dividida em seis subseções e abrange os dados apresentados em nossas sete tabelas. Na seção6, discutimos como os desenvolvedores podem usar nossa análise para selecionar uma plataforma de desenvolvimento blockchain para seus projetos. Por fim, fornecemos algumas considerações finais na Seção7.

2. Antecedentes

Em sua forma mais básica, um blockchain é um livro digital que é inviolável, inviolável e geralmente implementado de maneira distribuída. Uma blockchain mantém essas propriedades por meio do uso de hashes criptográficos. Uma blockchain é segmentada em "blocos" que contêm dados. Cada bloco contém o hash criptográfico do bloco anterior. Incluir o hash do bloco anterior em um bloco cria uma cadeia de blocos vinculados por hashes. Se os dados mudarem em algum bloco, os hashes devem ser atualizados em todos os blocos subsequentes. Para uma descrição mais detalhada do blockchain, consulteYaga et al.(2018),Guo e Yu(2022), eNofer et ai. (2017).

Este trabalho não analisa blockchains específicos, mas sim plataformas de desenvolvimento de blockchain. Uma plataforma de desenvolvimento blockchain é qualquer plataforma que permite aos usuários desenvolver aplicativos baseados em blockchain. Por exemplo, Ethereum é uma plataforma de desenvolvimento de blockchain, enquanto a rede principal Ethereum ou a rede de testes Ropsten são exemplos de blockchains Ethereum. Embora essa distinção possa parecer sutil, é uma distinção importante a ser feita em relação aos blockchains autorizados. Plataformas como o Hyperledger Fabric permitem que os desenvolvedores criem seus próprios blockchains. Nesse caso, fica claro que a plataforma de desenvolvimento blockchain é diferente da blockchain. Blockchains se diferenciam através do mecanismo de controle Sybil (Iqbal e Matulevičius,2021). Muitos trabalhos anteriores se referem a isso como protocolo de consenso ou algoritmo de consenso. No entanto, esses mecanismos evitam especificamente ataques Sybil; portanto, são propriamente chamados de mecanismos de controle Sybil. Em contraste, o protocolo de consenso é como uma rede concorda quando ocorre uma disputa, como uma bifurcação na blockchain. No Bitcoin, o mecanismo de controle Sybil é Proof of Work (PoW), e o protocolo de consenso é que a rede seleciona a cadeia mais longa quando há uma bifurcação.

Existem muitos mecanismos de controle Sybil diferentes e está fora do escopo deste trabalho discuti-los aqui. No entanto, os leitores são encorajados a verMingxiao et al.(2017),Nguyen e Kim(2018), eVelliangiri e Karthikeyan(2020) para pesquisas detalhadas que abrangem mecanismos populares de controle Sybil. Outro diferencial das blockchains é a estrutura de permissão. Um blockchain pode ser público ou privado, bem como sem permissão ou com permissão. Essas propriedades definem como os usuários podem interagir com uma rede blockchain.

Um blockchain público é um blockchain onde qualquer usuário pode entrar e ler os dados no blockchain. Da mesma forma, um blockchain pode ser sem permissão, onde qualquer usuário pode ingressar no mecanismo de controle Sybil e tentar adicionar novos blocos. Em contraste, um blockchain privado permite apenas que um pequeno grupo de usuários leia os dados no blockchain. Da mesma forma, em um blockchain autorizado, os usuários devem ter permissão para ingressar no mecanismo de controle Sybil e tentar adicionar novos blocos.

Por fim, as plataformas de desenvolvimento de blockchain podem ser categorizadas por sua geração (Tavares e outros.,2019). Sendo o primeiro blockchain,

Bitcoin é uma blockchain de geração 1.X. Esta geração de blockchain se concentra em um único caso de uso para o blockchain e permite apenas uma pequena quantidade de expansão. A blockchain do Bitcoin se concentra apenas na gravação de transações \$ XBT e não permite que os desenvolvedores escrevam dapps expansivos.

Querendo expandir a tecnologia blockchain além da criptomoeda, os desenvolvedores criaram plataformas de desenvolvimento de blockchain de geração 2.X, como Ethereum. Essas plataformas permitiram que os desenvolvedores escrevessem contratos inteligentes. Essa geração de blockchain, pela primeira vez, permitiu que os desenvolvedores usassem o blockchain em uma variedade de casos de uso e levou os desenvolvedores a criar dapps e Web3.

Embora as plataformas de desenvolvimento da geração 2.X ainda sejam populares entre os desenvolvedores, as plataformas da geração 3.X visam criar blockchains adequados para aplicativos corporativos. Alguns dos principais recursos das plataformas de desenvolvimento de blockchain da geração 3.X incluem protocolos de camada 2, comunicação máquina a máquina (M2M), compatibilidade móvel, uso de DAGs e uso de protocolos de computação.

Nas seções a seguir, usaremos a terminologia discutida aqui para destacar os recursos de várias plataformas de desenvolvimento de blockchain.

3. Trabalho relacionado

À medida que o blockchain se tornou mais popular, muitos pesquisaram várias plataformas de desenvolvimento de blockchain. Já em 2017 (Valenta e Sandner,2017), os pesquisadores têm comparado as plataformas blockchain existentes umas com as outras. Este trabalho compara Ethereum, Hyperledger Fabric e Corda usando características como Governança, Consenso e moeda nativa.

Outras obras, comoPolge et ai.(2021), analisou ainda mais o desempenho do Hyperledger Fabric, Ethereum, Quorum, Multichain e Corda. Este trabalho não analisa as considerações práticas de desenvolvimento, como a forma como os contratos inteligentes são escritos, mas compara os desempenhos dos blockchains fornecidos. Este trabalho descobriu que o Hyperledger Fabric e o Ethereum tiveram melhor desempenho em seus testes.

Trabalho de comparação de desempenho semelhante, comoDernayka e Chehab (2021) descobriram que o Ethereum teve o melhor desempenho em seus testes. Este trabalho comparou recursos adicionais, como armazenamento e recursos necessários. No entanto, este trabalho analisou apenas Ethereum e EOS, deixando de fora muitos blockchains populares. Da mesma forma, outras comparações analisam apenas plataformas blockchain no contexto de um domínio específico, como saúde (Agbo e Mahmoud,2019). Este trabalho comparou recursos como velocidade, segurança e escalabilidade de várias plataformas de desenvolvimento de blockchain em aplicativos de saúde. Novamente, este trabalho comparou apenas algumas plataformas.

Trabalhos mais extensos comoTavares e outros.(2019) analisa a pesquisa de blockchain como um todo. Este trabalho usa várias métricas para comparar cadeias de blocos, como recursos do Blockchain 3.0, recursos de consenso e documentação. Também usamos algumas dessas métricas em nosso trabalho para comparar as plataformas blockchain. No entanto, este trabalho não distingue entre plataformas de desenvolvimento de blockchain e outros projetos de blockchain, tornando difícil para novos desenvolvedores selecionar uma plataforma apropriada.

Este trabalho anterior deixa de fora muitas plataformas populares de blockchain em suas comparações. Novos desenvolvedores de blockchain devem estar cientes de todas as principais plataformas de desenvolvimento de blockchain para decidir qual plataforma melhor se adapta ao seu projeto. Da mesma forma, essas pesquisas não fornecem os detalhes necessários para que novos desenvolvedores escolham qual plataforma blockchain usar. Embora seja útil conhecer o desempenho de vários blockchains, os novos desenvolvedores precisam considerar mais aspectos, como os recursos fornecidos, os recursos de aprendizado fornecidos ou o ambiente de desenvolvimento. Neste trabalho, procuramos melhorar comparando de forma abrangente as 23 plataformas de desenvolvimento de blockchain mais populares. Mostramos como cada plataforma difere e fornecemos as informações necessárias para que os desenvolvedores selecionem uma plataforma que melhor se adapte aos seus projetos.

A seção a seguir descreve como selecionamos nossas plataformas para análise. Em seguida, fornecemos uma série de métricas comparando as várias plataformas. Por fim, discutimos como os desenvolvedores podem usar nossas métricas para selecionar uma plataforma apropriada.

4. Projetos selecionados

Existem mais de 146.000 projetos blockchain no GitHub. A maioria dos projetos foi criada apenas nos últimos anos. Como existem tantos projetos, pode levar algum tempo para novos desenvolvedores encontrarem uma plataforma de desenvolvimento que melhor se adapte aos objetivos de seus projetos. Este trabalho visa fornecer aos desenvolvedores as informações para selecionar uma plataforma para desenvolver aplicativos blockchain. Como existem tantas plataformas de desenvolvimento de blockchain, com muitas novas sendo criadas a cada ano, este trabalho analisou apenas plataformas populares de código aberto que permitem o desenvolvimento dinâmico de contratos inteligentes.

Consideramos um projeto suficientemente popular se tiver mais de mil (1000) estrelas no GitHub. A restrição de popularidade garante que nossas plataformas de desenvolvimento de blockchain analisadas sejam de alta qualidade. Definimos nosso requisito de popularidade baixo o suficiente para incluir muitas plataformas, mas alto o suficiente para garantir que todas as nossas plataformas discutidas sejam bem conhecidas. No entanto, essa restrição de popularidade favorece os blockchains mais antigos, pois eles tiveram mais tempo para acumular seguidores. Da mesma forma, analisamos apenas plataformas de código aberto para que os desenvolvedores pudessem ver o código subjacente. A transparência adicional de projetos de código aberto permite um código mais confiável.

Por fim, analisamos apenas as plataformas que permitiam o desenvolvimento dinâmico de contratos inteligentes, pois nosso objetivo é auxiliar os desenvolvedores na criação de novos aplicativos distribuídos. Essa restrição limita os projetos analisados apenas a plataformas que permitem aos desenvolvedores usar blockchains de geração 2.X ou 3.X. No total, consideramos 65 plataformas; no entanto, apenas 23 dessas plataformas atendem aos nossos requisitos. Algumas plataformas conhecidas que não atenderam aos nossos requisitos incluem Cardano (David et al., 2023), Avalanche (Sekniqi et al., 2020), IOTA (Popov, 2018) e Lisk (anon, 0000b). A lista completa das 65 plataformas consideradas está listada em nosso Repositório GitHub (Connors, 2023).

Seção 4 mostra os projetos que selecionamos. Marcamos projetos que são bifurcações de projetos mais antigos com um asterisco. Quorum e BNBChain são forks do Ethereum, e Qtum é um fork do Bitcoin. Observe que os dados de todas as nossas tabelas foram coletados em setembro de 2022. Muitos whitepapers que definem as plataformas de desenvolvimento de blockchain não foram submetidos a publicações revisadas por pares. Em vez disso, muitas plataformas publicam seu whitepaper diretamente em seu site ou página do GitHub. Fornecemos nosso próprio repositório GitHub (Connors, 2023) com links para o site de cada projeto, whitepaper e outros links úteis relacionados ao projeto.

Na seção a seguir, analisaremos nossas 23 plataformas selecionadas e forneceremos métricas para ajudar os desenvolvedores a selecionar uma plataforma que melhor complemente seus projetos.

5. Análise de projetos blockchain

5.1. Popularidade

Na primeira etapa de nossa análise, comparamos a popularidade de nossos projetos selecionados usando a métrica de pontuação de popularidade, mostrada em [Tabela 2](#). Devido à nossa restrição de popularidade, todas as 23 plataformas selecionadas têm pelo menos 1.000 estrelas no GitHub. Assim, mesmo as menos populares de nossas plataformas selecionadas são bem conhecidas. Os desenvolvedores podem usar nossas medidas de popularidade para avaliar o tamanho da comunidade em torno de uma plataforma de desenvolvimento blockchain e quantos desenvolvedores estão usando uma determinada plataforma.

Nossa métrica de pontuação de popularidade, Eq. (1), é gerado a partir de várias fontes que medem a popularidade de um projeto. Usamos o GitHub Stars, o GitHub Forks, o número de perguntas nos projetos Stack Exchange e o número de artigos no Google Scholar mencionando cada projeto. Algumas métricas de popularidade, como o número de DApps desenvolvidos, foram deixadas de fora de nossa pontuação de popularidade, pois nem todos os nossos projetos foram listados nos repositórios DApp.

Como todos os nossos projetos têm repositórios GitHub, podemos comparar o número de GitHub Stars. Uma estrela no GitHub permite que um desenvolvedor goste de um projeto e o salve para mais tarde. Assim, esta métrica reflete quantos

os desenvolvedores estão interessados em cada plataforma de desenvolvimento blockchain. Da mesma forma, vimos os forks do GitHub. O número de forks do GitHub nos permite saber quantos desenvolvedores estão interessados em criar projetos baseados em uma plataforma de desenvolvimento blockchain.

Da mesma forma, examinamos o número de problemas fechados do GitHub para ver quantas pessoas estavam fazendo perguntas sobre a plataforma de desenvolvimento blockchain. No entanto, algumas plataformas excluem problemas antigos quando uma nova atualização para a plataforma de desenvolvimento blockchain é lançada. Os problemas excluídos explicam por que plataformas como o Hyperledger Fabric têm menos problemas fechados do GitHub do que o esperado. Devido aos problemas excluídos, não incluímos essa métrica ao calcular as pontuações de popularidade. Também medimos a popularidade dos projetos fora do GitHub usando o Stack Exchange. Stack Exchange é um site onde os usuários podem fazer perguntas e receber respostas de outras pessoas com experiência na plataforma. Para todas as nossas plataformas, pesquisamos o Stack Exchange usando a tag blockchain seguida do nome da plataforma para eliminar questões não relacionadas ao blockchain. Por exemplo, ao pesquisar Waves no Stack Exchange, revivemos muitas respostas relacionadas à física e não ao blockchain; assim, procuramos por "[blockchain] Waves". Embora isso possa ter removido algumas perguntas sobre cada plataforma que não incluía a tag blockchain, garantiu que todas as perguntas retornadas fossem sobre a plataforma de desenvolvimento blockchain relevante.

Da mesma forma, usando o Google Scholar, medimos o número de trabalhos acadêmicos relacionados a cada plataforma de desenvolvimento de blockchain. Novamente, para todas as nossas plataformas selecionadas, precisamos restringir os resultados apenas a artigos contendo a plataforma seguida da palavra blockchain. Por exemplo, ao pesquisar Optimism, Blockchain retorna muitos artigos onde os autores declaram seu otimismo sobre a tecnologia blockchain; assim, em vez disso, buscamos apenas artigos que incluíam Optimism, seguido de blockchain, que retornou apenas artigos sobre a plataforma. Novamente, isso provavelmente deixou cair alguns papéis relacionados à plataforma, mas garantiu que todos os papéis encontrados fossem relevantes para a plataforma.

A BNB Chain não teve nenhum resultado do Google Scholar. A falta de papéis é porque o blockchain foi formalmente chamado de Binance Smart Chain (BSC). No entanto, a Binance fundiu o BSC com outras tecnologias de blockchain que eles hospedavam e o renomeou como BNB Chain. Não procuramos artigos relacionados ao Binance Smart Chain porque existem diferenças significativas entre o BNB Chain e seu antecessor.

Por fim, analisamos quantos dapps cada plataforma de desenvolvimento blockchain postou no DappRadar. DappRadar é um site que agrega e classifica dApps conhecidos. Naturalmente, nenhuma das plataformas que criam blockchains privados possui entradas no DappRadar, pois os aplicativos criados nessas plataformas não são públicos. No entanto, algumas plataformas públicas de blockchain, como Stellar, não possuem entradas no DappRadar. Como o DappRadar não coleta informações sobre todos os nossos projetos selecionados, não incluímos o número de dApps ao calcular a pontuação de popularidade.

Para calcular a pontuação de popularidade mostrada na Eq. (1), pegamos as somas normalizadas de bifurcações do GitHub, estrelas do GitHub, perguntas do StackExchange e documentos do Google Scholar. A plataforma mais popular pode ter uma pontuação de 100.

$$= \frac{\text{GitHub Stars} + \text{GitHub Forks} + \text{Stack Exchange} + \text{Google Scholar}}{\text{Máximo Valor}} * 100 \quad (1)$$

Ethereum é a plataforma mais popular, com uma pontuação de 100. Hyperledger Fabric é a plataforma privada de desenvolvimento de blockchain mais popular e a segunda plataforma mais popular, com pontuação de 32,93. Enquanto nosso projeto menos popular Zilliqa tem apenas uma pontuação de 1,76. A pontuação de popularidade apenas nos permite comparar o projeto selecionado entre si. Todas as plataformas mostradas têm pelo menos 1000 estrelas no GitHub, o que significa que são todas bem conhecidas.

Uma plataforma popular não é a melhor plataforma para todos os projetos. Os desenvolvedores devem considerar a popularidade de um projeto apenas como um meio de ver quanto suporte está disponível. Esperamos que ao usar esta tabela em conjunto com nossas outras tabelas, os desenvolvedores possam escolher a plataforma que melhor suporte seus projetos.

5.2. Propriedades das plataformas de desenvolvimento blockchain

Nossa tabela subsequente, [Tabela 3](#), descreve os diferenciais fundamentais entre blockchains vinculados às plataformas de desenvolvimento de blockchain. A tabela inclui a geração do blockchain, se o blockchain é permissionado ou não, qual mecanismo de controle Sybil o blockchain usa se os desenvolvedores precisam aprender uma linguagem de programação específica da estrutura para desenvolver contratos inteligentes e se uma criptomoeda governa o blockchain. Algumas de nossas plataformas selecionadas estão vinculadas a uma rede principal bem estabelecida como a Ethereum. Em contraste, outras plataformas farão com que os desenvolvedores criem seus próprios blockchains, como o Hyperledger Fabric.

A maioria de nossas plataformas selecionadas está vinculada a blockchains de geração 2.X. Dos blockchains da geração 3.X, tanto o Optimism quanto o Harmony estão vinculados aos blockchains da camada 2 (L2). Uma blockchain L2 realiza transações em uma cadeia lateral menor, mas usa uma blockchain mais extensa para finalidade. Tanto o Optimism quanto o Harmony finalizam suas transações na blockchain da rede principal Ethereum. Os desenvolvedores devem observar que, embora os L2s pretendam aumentar a velocidade da transação, eles abrem mão de alguma segurança usando uma rede menor. Essa troca de segurança por velocidade é essencial em aplicativos que exigem uma confiança de rede excepcionalmente alta. No entanto, a velocidade obtida com o uso de um L2 pode compensar a segurança reduzida para muitos aplicativos.

Da mesma forma, NEO e IoTeX permitem a comunicação M2M, outra propriedade das blockchains da geração 3.X. Os desenvolvedores que desejam criar aplicativos IoT devem considerar o uso dessas plataformas de desenvolvimento blockchain.

Nossos projetos selecionados contêm uma mistura de blockchains com e sem permissão. Frequentemente, aplicativos corporativos ou outros aplicativos privados exigirão um blockchain autorizado. No entanto, os desenvolvedores devem estar cientes de que, ao criar um blockchain autorizado, eles também podem precisar criar e operar nós de blockchain, criar uma autoridade para conceder permissão à rede blockchain ou até mesmo escolher o algoritmo de consenso que a rede usará. Por exemplo, no Hyperledger Fabric, um desenvolvedor precisa criar sua própria rede privada antes de começar a trabalhar no desenvolvimento de seu dApp. Embora o Hyperledger Fabric ofereça um conjunto de ferramentas para tornar essa etapa menos complexa, ele ainda complica o processo de desenvolvimento do dApp.

Por outro lado, os aplicativos voltados para o público geralmente exigem que os desenvolvedores usem blockchains sem permissão. Embora os desenvolvedores não precisem de suas próprias redes, eles geralmente precisam pagar uma taxa para usar a rede principal pública. Da mesma forma, os desenvolvedores terão pouco controle sobre a rede; portanto, um blockchain sem permissão pode não ser adequado para aplicativos que lidam com informações confidenciais. Os desenvolvedores devem considerar qual estrutura de permissão é melhor para seu aplicativo. Algumas plataformas selecionadas, como EOS, NEO e NEAR, permitem que os desenvolvedores usem uma rede sem permissão ou criem uma rede privada. O acesso a dois tipos de redes permite que os desenvolvedores alternem o tipo de rede que estão usando sem trocar de plataforma. Por exemplo, um desenvolvedor pode começar usando uma rede principal pública. No entanto,

Nossos projetos selecionados usam vários mecanismos de controle Sybil, geralmente chamados de algoritmos de consenso. Embora o blockchain original, Bitcoin, use Proof of Work (PoW) como seu mecanismo de controle Sybil, nenhum de nossos projetos selecionados usa PoW. PoW demonstrou ter muitas desvantagens negativas em comparação com algoritmos modernos, como Proof of Stake (PoS), incluindo alto consumo de energia ([Zhang e Chan, 2020](#)) e desempenho lento ([Nair e Dorai, 2021](#)). Assim, PoS e algoritmos semelhantes a PoS são os mais populares em nossa lista.

Os desenvolvedores devem entender os prós e contras do mecanismo de controle Sybil. Sugerimos que os desenvolvedores leiam o whitepaper do projeto selecionado para entender como uma plataforma específica de desenvolvimento de blockchain implementa o mecanismo de controle Sybil em seus blockchains.

Alguns blockchains, como o Hyperledger Fabric, permitem que o usuário escolha qual mecanismo a rede usará quando o usuário criar o

rede. Essa camada de mecanismo de controle Sybil modular pode ser desejável para desenvolvedores que precisam de mecanismos de controle específicos para um determinado aplicativo.

Antes de iniciar um projeto, os desenvolvedores devem saber se um blockchain requer uma linguagem de programação específica para o desenvolvimento de contratos inteligentes. Solidity é a linguagem de programação específica de framework mais conhecida, criada inicialmente para desenvolver dapps Ethereum. [Tabela 6](#) mostra todos os idiomas disponíveis para cada projeto. Como os contratos inteligentes podem ser responsáveis por lidar com transações financeiras, os desenvolvedores devem se certificar de que entendem as peculiaridades de qualquer idioma que escolherem. Um pequeno erro no código pode facilmente levar a uma perda financeira significativa para os desenvolvedores e usuários do dApp.

Por fim, blockchains sem permissão são regidos por criptomoedas. Os desenvolvedores devem pagar uma taxa para executar seus aplicativos em um blockchain sem permissão. Algumas plataformas têm várias criptomoedas. O NEO tem \$NEO para participar do PoS e \$GAS para pagar as taxas de criação do bloco. Como os preços das criptomoedas são voláteis, não os relatamos aqui; no entanto, os desenvolvedores devem pesquisar o custo de criação de dApps em sua plataforma selecionada.

Os desenvolvedores devem entender os fundamentos de cada plataforma de desenvolvimento blockchain antes de iniciar um projeto. Os desenvolvedores devem garantir que a plataforma esteja alinhada com os objetivos do projeto. Por exemplo, um desenvolvedor deve selecionar um blockchain sem permissão se planeja criar um aplicativo que qualquer pessoa possa acessar. Compreender os fundamentos ajudará os desenvolvedores a escolher a melhor plataforma para seus projetos.

5.3. Aplicativos existentes

[Tabela 4](#) analisa os vários aplicativos criados usando cada plataforma de desenvolvimento blockchain. Antes de iniciar um aplicativo, os desenvolvedores devem estar cientes dos tipos de projetos para os quais cada plataforma foi projetada. Embora muitas plataformas sejam projetadas para aplicações gerais, algumas têm casos de uso específicos.

A área de aplicação mais comum do blockchain está na FinTech; no entanto, blockchain tem muitos casos de uso em outras áreas. Muitas plataformas privadas de blockchain, como o Hyperledger Fabric, se concentram em permitir que os usuários criem aplicativos corporativos seguros. Da mesma forma, algumas plataformas de desenvolvimento de blockchain, como a Exonum, se concentraram em permitir que os usuários criem aplicativos governamentais. A Stellar está focada em transferências interbancárias, um caso de uso específico da FinTech. Os desenvolvedores devem saber quais plataformas são projetadas para apoiar seu setor de projetos.

Da mesma forma, algumas plataformas de desenvolvimento de blockchain focam na conectividade com blockchains existentes. Optimismo e Harmonia são blockchains L2; assim, ambos se concentram na compatibilidade com o blockchain da rede principal Ethereum. Da mesma forma, Stacks e Qtum permitem que os desenvolvedores criem dApps que interagem com o blockchain do Bitcoin. Os desenvolvedores que criam aplicativos que precisam interagir com esses blockchains devem considerar qual suporte está disponível na plataforma de desenvolvimento escolhida.

Para ajudar os desenvolvedores a entender melhor quais tipos de aplicativos podem criar em cada plataforma, listamos alguns casos de uso comuns e alguns dApps populares para cada projeto. Alguns projetos exclusivos incluem a equipe de F1 Red Bull Racing usando Tezos para gerenciamento de equipe e o país da Geórgia usando Exonum para registro de terras. Embora esteja fora do escopo deste trabalho discutir cada um desses dApps, os desenvolvedores são incentivados a pesquisar os dApps existentes. Em particular, o código dApp está disponível para qualquer desenvolvedor ler em blockchains públicos, o que pode ajudar novos desenvolvedores a entender os paradigmas de codificação comuns.

Os desenvolvedores devem se familiarizar com alguns aplicativos existentes em cada plataforma. Isso ajudará os desenvolvedores a entender quais aplicativos são possíveis em cada plataforma. Por exemplo, se um desenvolvedor deseja criar um aplicativo IoT, ele deve considerar plataformas projetadas para IoT. A sinergia entre a plataforma e o projeto pode tornar as aplicações mais fáceis de construir e mais acessíveis aos usuários.

5.4. Considerações de desenvolvimento

Tabela 5 destaca as principais áreas que os desenvolvedores devem entender antes de iniciar um projeto. A tabela cobre os recursos de privacidade de cada plataforma, a documentação disponível para desenvolvedores, se há uma rede de teste para desenvolvedores, se há ferramentas disponíveis para desenvolvedores e quanto controle os desenvolvedores têm sobre o mecanismo de controle Sybil.

Além deste artigo, criamos um repositório GitHub ([Connors, 2023](#)), onde postamos links para os projetos. Nosso repositório GitHub inclui links para a documentação do projeto, recursos de aprendizado, ferramentas como estruturas de teste e recursos da comunidade.

Atribuímos uma pontuação de privacidade a cada plataforma. A pontuação de privacidade permite que os desenvolvedores conheçam os recursos de privacidade disponíveis em cada plataforma. Aplicativos de negócios podem exigir mais privacidade do que aplicativos gerais, como jogos. Assim, os desenvolvedores devem selecionar uma plataforma que permita o nível adequado de privacidade. Um projeto recebia 1 ponto se permitisse transações privadas, 2 pontos para Zero-Knowledge Proofs (ZKP) e 4 pontos se permitisse que desenvolvedores criassem seu próprio blockchain privado. Um ZKP é uma maneira de dois atores interagirem entre si sem abrir mão de suas identidades. ZCash ([Ben Sasson et al., 2014](#)) é uma criptomoeda popular que implementa ZKP para permitir transações privadas de blockchain. Para obter informações detalhadas sobre o ZKP, consulte [Goldreich e Oren \(1994\)](#) e [Sol e outros. \(2021\)](#).

Como atribuímos pontos como potências de dois, uma pontuação de privacidade pode ser facilmente decomposta em suas partes convertendo a pontuação em binário. Por exemplo, Tron marcou 5 pontos, que na base 2 é bin(101). Lendo o binário da esquerda para a direita, vemos que ele permite a criação de blockchains privados (bin(100)), não tem suporte para ZKP (bin(00)) e suporta transações privadas (bin(1)).

Usando um sistema de pontuação semelhante, atribuímos a cada projeto uma pontuação de documentação. Essa métrica visa mostrar aos desenvolvedores quanto suporte está disponível na plataforma. Um projeto recebeu 1 ponto por ter documentação escrita oficial, 2 pontos por ter um tutorial simples sobre como criar um aplicativo, 4 pontos se a documentação mostrasse aos desenvolvedores como criar um aplicativo de exemplo completo, 8 pontos se a documentação incluísse vídeos e 16 pontos se a plataforma forneceu documentação extensa. Alguns exemplos de documentação extensa incluem Ethereum e Tezos com desenvolvedores de jogos que podem jogar para aprender o básico da plataforma ou EOS com um curso de treinamento oficial que os desenvolvedores podem fazer.

A próxima coluna indica se a plataforma possui uma rede de teste. Como no desenvolvimento Web2, os desenvolvedores muitas vezes só desejam implantar seu código na produção depois de concluírem o teste. Uma rede de teste é um blockchain onde os desenvolvedores podem testar seus dApps. Como todos os outros blockchains, um testnet pode ser público ou privado. Uma rede de teste privada exigirá que os desenvolvedores criem seu próprio ambiente para testes.

Freqüentemente, uma torneira acompanha um teste público. Uma faucet dará aos usuários criptomoedas para pagar taxas de criação de blocos em uma rede de teste pública. A criptomoeda fornecida pela faucet é diferente da criptomoeda da rede principal. Da mesma forma, as torneiras dependem do apoio da comunidade e podem secar; isso pode levar a problemas para os desenvolvedores que desejam testar em redes de teste públicas.

A seguir, observamos quais plataformas possuem ferramentas de desenvolvimento oficiais. As ferramentas de desenvolvimento são criadas por uma plataforma de desenvolvimento blockchain para auxiliar os desenvolvedores na criação de dApps. Por exemplo, Ethereum tem o Remix IDE, um IDE online e muitos frameworks para testar dApps. Os desenvolvedores devem estar cientes das ferramentas de desenvolvimento disponíveis que podem auxiliá-los na criação de dApps.

Por fim, cobrimos a modularidade da camada Sybil Control de cada projeto, geralmente chamada de camada de consenso. Os desenvolvedores podem precisar otimizar o mecanismo de controle Sybil para melhor atender a sua aplicação. Poucas plataformas permitem aos desenvolvedores a flexibilidade de alterar a camada de controle Sybil. Hyperledger Fabric e Sawtooth permitem que os desenvolvedores selecionem um conjunto de mecanismos de controle Sybil (mostrado em [Tabela 3](#)). Da mesma forma, os desenvolvedores podem modificar parte do mecanismo de controle Sybil se criarem um

tabela 1

Consideramos 65 plataformas e selecionamos as 23 plataformas mais populares para nossa análise. As datas marcadas com um * representam pacotes que são bifurcações de pacotes mais antigos. Uma lista completa das 65 plataformas consideradas está disponível em nosso repositório GitHub.

Plataforma	Nome do pacote Github	Confirmação inicial data	papel branco
Ethereum	ethereum/go-ethereum	dezembro de 2013	Buterin(2014)
Hyperledger	hyperledger/tecido	maio de 2016	Androulaki et al. (2018) , Hyperledger Fundação (2018)
EOS	EOSIO/eos	abril de 2017	Larimer et ai. (2018)
solana	solana-labs/solana	fevereiro de 2018	Yakovenko (2018)
hortelã	Tendermint/Tendermint	abril de 2014	Kwon(2014)
Quorum	ConsenSys/quórum	dezembro de 2013*	ConsenSys (2018)
corda	corda / corda	novembro de 2015	Marrom(2018) , ouvir e Marrom(2019)
neo	neo-projeto/neo	maio de 2015	anon(0000a)
Tron	tronprotocol/java-tron	dezembro de 2017	TRON(2018)
Estelar	estelar/estelar-core	novembro de 2014	Mazieres(2016)
Pilhas	stacks-network/stacks-blockchain	janeiro de 2014	Todos(2020)
Cadeia BNB	bnb-chain/bsc	dezembro de 2013*	Bnb-Chain (2020)
Hyperledger dente de serra	hyperledger/núcleo dente de serra	março de 2016	Olson e outros. (2018) , Hyperledger Fundação (2018)
APROXIMAR	NEAR/nearcore	outubro de 2018	APROXIMAR Fundação (2022)
Tezos	tezos/tezos	Set 2016	Bom homem(2014)
Otimismo	ethereum-otimismo/otimismo	Set 2020	Equipe Otimismo (2023)
IoTeX	iotexproject/iotex-core	abril de 2018	A Equipe Iotex (2018)
Harmonia	harmonia-um/harmonia	maio de 2018	a harmonia Equipe(2019)
Ondas	plataforma de ondas/Ondas	janeiro de 2015	Waves Enterprise (2022)
Algorand	algorand/go-algorand	junho de 2019	Chen e Micali (2016)
QtumName	qtumproject/qtum	agosto de 2009*	Qtum OrgQtumName (2020)
Exonum	exon/exonum	abril de 2016	Yanovitch et ai. (2018)
Zilliqa	Zilliqa/Zilliqa	dezembro de 2017	anon(2017)

Rede EOS. Observe que nenhuma outra plataforma, incluindo as plataformas para blockchains privados, permite que os desenvolvedores modifiquem a camada Sybil Control, tornando as plataformas Hyperledger e EOS únicas.

Os desenvolvedores devem considerar os recursos de privacidade e a flexibilidade das camadas de controle Sybil para garantir que a plataforma selecionada suporte o projeto. Da mesma forma, os desenvolvedores mais novos podem considerar plataformas com bastante documentação, uma rede de teste fácil de usar e ferramentas de desenvolvimento. Os desenvolvedores podem selecionar melhor uma plataforma que se adapte ao seu nível de habilidade e objetivos do projeto usando nossa tabela.

5.5. Recursos de contrato inteligente

As tabelas a seguir destacam os recursos dos contratos inteligentes fornecidos por cada plataforma. [Tabela 6](#) mostra quais linguagens os desenvolvedores podem usar para escrever contratos inteligentes. Os idiomas marcados com um * são idiomas projetados para o desenvolvimento de contratos inteligentes.

mesa 2

Esta tabela mostra várias métricas de popularidade para nossas plataformas selecionadas. A pontuação de popularidade é a soma normalizada dos Forks, Stars, Stack Exchange Questions e Google Scholar Articles.

Projeto	Estrelas do Github	Garfos do Github	Fechado Problemas do Github	Pilha intercâmbio	Google estudioso	Dapps	Popularidade pontuação
Ethereum	39163	14840	6391	6317	18500	3452	100,00
Hyperledger Tecido	13950	8190	162	1502	2310	–	32,93
EOS	11346	3700	4751	81	275	579	19.54
solana	9372	2329	3491	383	78	105	15.43
hortelã	5142	1786	2800	68	138	–	9.05
Quorum	4234	1184	757	206	379	–	7.62
corda	3878	1069	579	500	225	–	7.20
neo	3381	1000	1020	32	194	1	5.84
Tron	3132	1139	1030	121	43	1397	5.63
Estelar	3000	992	1157	33	190	–	5.35
Pilhas	2688	545	1749	219	20	–	4,40
Cadeia BNB	1685	788	652	86	0	4067	3.25
Hyperledger dente de serra	1393	762	4	113	135	–	3.05
APROXIMAR	1822	364	2064	170	13	43	3.01
Tezos	1501	209	0	46	212	80	2,50
Otimismo	1397	450	434	4	4	25	2.35
iotex	1431	291	987	1	18	45	2.21
Harmonia	1447	268	971	12	12	117	2.21
Ondas	1164	419	343	67	72	16	2.18
Algorand	1132	346	5	21	112	12	2.04
Quantum	1162	389	292	11	28	–	2.02
Exonum	1176	245	340	5	57	–	1,88
Zilliqa	1110	263	264	1	15	–	1,76

Nossa lista de plataformas contém uma variedade de linguagens, sendo as linguagens mais comuns Javascript e Solidity. Poucas plataformas permitem contratos inteligentes móveis. Apenas Stellar e Hyperledger Sawtooth permitem que os desenvolvedores usem linguagens móveis como Swift.

Linguagens específicas de framework, como Solidity, oferecem o benefício de serem projetadas para o desenvolvimento de blockchain ao custo de os desenvolvedores aprenderem uma nova linguagem de programação. Ao selecionar uma plataforma, os desenvolvedores devem reconhecer os prós e os contras de usar uma linguagem específica do framework. Por exemplo, considere um desenvolvedor que deseja criar um aplicativo financeiro. Um pequeno erro no código pode levar a uma perda financeira significativa. Por um lado, o desenvolvedor pode querer programar em uma linguagem que já conhece, pois já entende as nuances da linguagem, mas pode ter um bug relacionado ao blockchain em seu código. Por outro lado, eles podem escolher uma linguagem específica do framework, pois ajudará o desenvolvedor a evitar erros de blockchain. Ainda assim, o desenvolvedor deve aprender a linguagem e entender suas nuances.

Tabela 7 mostra quais plataformas podem ser atualizadas diretamente após a implantação. Como os contratos inteligentes são escritos em um blockchain, que é imutável, os desenvolvedores nunca podem alterar o código original após a publicação. No entanto, algumas plataformas possuem sistemas para garantir que os desenvolvedores possam atualizar seu código e o código antigo é ignorado. Embora os desenvolvedores tenham conseguido replicar esses sistemas na maioria das plataformas, apenas algumas de nossas plataformas selecionadas permitem que os desenvolvedores atualizem seu código diretamente após a implantação.

Algumas de nossas plataformas selecionadas permitem que os desenvolvedores criem contratos ricardianos. Embora os contratos inteligentes sejam frequentemente comparados a acordos legais, eles não são contratos legais. Um Contrato Ricardiano é um contrato legal que máquinas e pessoas podem ler (Grigg, 2004). Os desenvolvedores que desejam criar aplicativos que exigem contratos legais devem considerar uma plataforma que permita contratos ricardianos.

Por fim, destacamos a implementação de contratos inteligentes em cada plataforma. O modelo de execução mostra quando as transações são solicitadas e quando o contrato inteligente é executado. Por exemplo, no modelo Order-Execute, as transações são primeiro ordenadas e, em seguida, o código é executado. Em contraste, no modelo Execute-Order-Validate, as transações são executadas e a saída dessas transações é ordenada em um bloco.

O artigo técnico do Hyperledger Fabric (Androulaki et al., 2018) detalha cada tipo de modelo de execução.

Da mesma forma, observamos o modelo de sistema operacional por trás de cada plataforma de desenvolvimento blockchain. A maioria de nossas plataformas é executada em uma máquina virtual distribuída, como a Ethereum Virtual Machine (EVM). O whitepaper de cada projeto contém informações sobre como sua máquina virtual é implementada.

Por fim, mostramos a linguagem compilada que cada modelo usa. Embora os desenvolvedores não precisem escrever código na linguagem compilada, pode ser útil entender todo o modelo durante a depuração. Em aplicativos tradicionais, os desenvolvedores não precisam saber escrever em assembly para criar aplicativos. No entanto, entender que o sistema compilará seu programa para montagem pode facilitar a depuração para os desenvolvedores. A mesma ideia se aplica ao desenvolvimento Web3; os desenvolvedores devem entender como seus aplicativos são compilados para que possam depurar seu código com mais facilidade.

Compreendendo os idiomas permitidos por cada plataforma, os desenvolvedores podem selecionar melhor uma plataforma com a qual se sintam confortáveis. Da mesma forma, os desenvolvedores devem entender quais recursos os contratos inteligentes fornecem para garantir que selecionem uma plataforma que tenha sinergia com seu projeto. Usando esta tabela, os desenvolvedores podem entender como os contratos inteligentes funcionam em cada plataforma para selecionar uma plataforma que atenda às necessidades de seus projetos.

5.6. Engajamento da comunidade

Tabela 8 resume a comunidade em torno de cada plataforma de desenvolvimento blockchain. Os desenvolvedores devem estar cientes do tamanho e do envolvimento de uma comunidade para que saibam a quem recorrer quando precisarem de assistência técnica. Da mesma forma, como o blockchain se concentra na descentralização, a comunidade costuma tomar decisões para mudanças substanciais na plataforma.

Primeiro registramos quem são os apoiadores da plataforma de desenvolvimento blockchain. As plataformas de desenvolvimento Blockchain suportadas por empresas terão objetivos finais diferentes das plataformas suportadas por indivíduos ou fundações. Os desenvolvedores devem ter em mente os objetivos dos patrocinadores da plataforma de desenvolvimento.

Tabela 3

Esta tabela mostra as principais propriedades de nossas plataformas de desenvolvimento de blockchain selecionadas. O Prog. Lang. coluna representa se o projeto requer uma linguagem específica do framework, como Solidity.

Plataforma	Geração	sem permissão	Consenso	Prog. Lang.	Cripto-moeda
Ethereum	2.0	Sim	PoS	Sim	\$ ETH (Éter)
Hyperledger Tecido	2.x	Não	Kafka, Solo, BFT-SMaRt	Não	–
EOS	2.x	Sim	aBFT + DPoS	Não	\$EOS (EÓsIO)
solana	2.x	Sim	PoH (Semelhante ao PoS)	Não	\$ SOL (Solana)
hortelã	2.x	Não	pBFT	Não	–
Quorum	2.x	Não	PoA	Sim	–
corda	2.x	Não	Nodos Notariais	Não	–
neo	3.x	Sim	dBFT (Semelhante ao PoS)	Não	\$ NEO (Neo) \$GAS (NeoGás)
Tron	2.x	Sim	DPoS	Sim	\$ TRX (TRON)
Estelar	2.x	Sim	FBA (Estelar)	Não	\$XML (Lúmen)
Pilhas	2.x	Sim	PoX (Prova de Transferir)	Sim	\$ STX (Pilhas)
Cadeia BNB	2.x	Sim	pBFT & PoSA	Sim	\$ BNB (BNB)
Hyperledger dente de serra	2.x	Não	PoET, Jangada	Não	–
APPROXIMAR	2.x	Sim	Nightshade	Não	\$ PERTO (APPROXIMAR)
Tezos	2.x	Sim	PoS	Não	\$XTZ (Tezos)
Otimismo	3.x	Sim	DPoS	Sim	\$ OP (Otimismo)
iotex	3.x	Sim	DPoS	Sim	\$IOTX (IoTeX)
Harmonia	3.x	Sim	pBFT	Sim	\$ UM (Harmonia)
Ondas	2.x	Sim	LPoS	Sim	\$ONDAS (Ondas)
Algorand	2.x	Sim	PPoS	Não	\$ALGO (Algorand)
QtumName	2.x	Não	MPoS	Não	\$QTUM (Qtum)
Exonum	2.x	Não	pBFT	Não	–
Zilliqa	2.x	Sim	pBFT	Sim	\$ZIL (Zilliqa)

Em seguida, registramos quando cada plataforma fez um commit no GitHub pela última vez. Registramos nossas datas em setembro de 2022. Além disso, registramos o número total de commits. Essas duas métricas mostram o quão ativa é uma comunidade de plataforma de desenvolvimento blockchain. Quando registramos nossos dados, apenas o EOS ainda não havia feito um commit no último ano. Da mesma forma, registramos se a plataforma possui um roteiro oficial. O roteiro descreve as mudanças futuras que a plataforma planeja fazer e quando pretende fazê-las. Os desenvolvedores devem saber o quão ativa é uma plataforma e quais atualizações estão planejadas. Alguns aplicativos não exigirão modificações na plataforma após a implantação, enquanto outros se beneficiarão de atualizações contínuas.

Medimos o quão ativa é a comunidade de cada plataforma nas redes sociais. Observamos quais plataformas possuem um canal oficial do Telegram ou Discord. Esses são aplicativos populares de mensagens em que os desenvolvedores podem ingressar em um grupo e fazer perguntas a outros membros da comunidade sobre a plataforma. Da mesma forma, registramos quantos vídeos do YouTube foram postados pela plataforma. Por fim, registramos o número de membros no subreddit de cada plataforma. Um subreddit é um fórum na plataforma de mídia social Reddit dedicado a um tópico específico. As plataformas vinculadas a uma criptomoeda têm significativamente mais membros do subreddit do que plataformas sem criptomoeda. Essa disparidade é tão drástica que

a plataforma menos popular Zilliqa, vinculada à criptomoeda \$ ZIL, tem mais membros do subreddit do que nosso segundo projeto mais popular, Hyperledger Fabric, que não possui criptomoeda.

Os desenvolvedores devem usar nossa tabela de comunidade para avaliar o quão ativa é a comunidade de uma plataforma para entender quanto suporte estará disponível. Projetos com uma comunidade maior terão mais pessoas para auxiliar um novo desenvolvedor. Da mesma forma, os desenvolvedores devem entender o quão ativa é a comunidade e quem são as partes interessadas, para que saibam para onde o projeto está indo e quando podem esperar que os novos recursos sejam implementados.

6. Discussão

Além de nossas tabelas, também criamos um repositório GitHub contendo links para o site de cada projeto, white paper, ferramentas, recursos de aprendizado, comunidade e exemplos de dApps. Esta página do GitHub pode ser encontrada aqui ([Connors,2023](#))

Nossa primeira mesa, [tabela 1](#), liste nossos projetos selecionados. Selecionamos apenas projetos que atendem a certos requisitos de popularidade para garantir que todos os nossos projetos selecionados sejam conhecidos o suficiente para que novos desenvolvedores

Tabela 4

Esta tabela mostra exemplos de aplicativos existentes para nossas plataformas de desenvolvimento de blockchain selecionadas. Links para os aplicativos de exemplo podem ser encontrados em nosso Repositório GitHub.

Projeto	Area de aplicação	Exemplos de casos de uso	Exemplos de DApps
Ethereum	FinTech, dApps em geral	Câmbio descentralizado, NFTs, jogos	Uniswap, Gatinhos criptográficos
Hyperledger Tecido	GovTech, Aplicações Corporativas, Saúde	votação, Cadeias de suprimentos B2B	Cadeia BRUIN, cadeia de saúde
EOS	dApps gerais, Aplicações Enterprise	Aplicativos de negócios, Jogos, Negociação de ativos digitais	Defibox, planalto
solana	dApps gerais, FinTech	jogos, DAO, Pagamento DeFi	Éden mágico, gameta
hortelã	FinTech	DEX, StableCoin, Comércio de Carbono	Terra, Rede Regan
Quorum	FinTech	Transferências Interbancárias, Mercado de Empréstimos	Projeto Ubin, Skeps
corda	FinTech, saúde, construção	Mercados capitais, Gestão de Reivindicações, Cadeia de mantimentos	HSBLOX
neo	FinTech, IoT	Loteria, Rede Social	Efeito, Nartivo
Tron	FinTech, dApps em geral	Câmbio descentralizado, NFTs, jogos	SunSwap, BSG
Estelar	FinTech	Transferências Interbancárias	IBM BWW
Pilhas	FinTech, Bitcoin dApps	Troca descentralizada, registrador de domínio	ALEX, Gama
Cadeia BNB	FinTech, dApps em geral	Câmbio descentralizado, NFTs, jogos	PancakeSwap, Era7
Hyperledger dente de serra	Saúde, Empresa Formulários	EHR, Gestão Empresarial	PokitDok, Sextante
APROXIMAR	FinTech, dApps em geral	Câmbio descentralizado, NFTs, jogos	Paras, Toca
Tezos	FinTech, GovTech, dApps em geral	Ativos estáveis, gerenciamento de equipe	lugh, Red Bull Racing
Otimismo	Fintech, dApps em geral	DeFi, mercados, Trocas	Quixotesco, Entre
IoTeX	IoT	Gerenciamento de IoT, Compartilhamento de dados IoT	StarCrazy, VITA
Harmonia	Fintech, dApps gerais	DeFi, NFT, Trocas descentralizadas	Sushi, Eterno
Ondas	dApps gerais	NFTs, jogos, DeFi	Viers.finance, Patos das Ondas
Algorand	FinTech, dApps em geral	Câmbio descentralizado, NFTs, jogos	Algodex, Algofi
Quantum	FinTech, Bitcoin dApps	Criando fichas, viagens	fichas QRC20, Travala
Exonum	GovTech, dApps	Gestão da propriedade, eAuctions	Registo Predial na Geórgia, Leilão eletrônico da Ucrânia plataforma
Zilliqa	FinTech, dApps	ePayments, Publicidade	Xfers, Aqiliz

poderia facilmente encontrar recursos para ajudá-los a aprender a plataforma. Consideramos 65 plataformas e nossa restrição de popularidade restringiu nossa análise a apenas 23 plataformas. Uma lista completa das 65 plataformas consideradas pode ser encontrada em nosso repositório GitHub.

Em seguida, analisamos ainda mais nossas plataformas para determinar a plataforma mais popular. Atribuímos a cada plataforma uma pontuação de popularidade, Eq.(1), com base no envolvimento do GitHub, do StackExchange e da popularidade acadêmica. A popularidade de cada projeto é mostrada em [Tabela 2](#). A popularidade e o apoio da comunidade mostrados em [Tabela 8](#) dar aos desenvolvedores uma ideia de quanto suporte está disponível. Novos desenvolvedores de blockchain geralmente precisam de suporte guiado para começar. Um projeto popular com uma grande comunidade oferece aos novos desenvolvedores recursos da comunidade para ajudá-los a aprender sobre uma plataforma. Sugerimos desenvolvedores que aprendem melhor

por meio de recursos da comunidade, como tutoriais do Youtube ou aplicativos de exemplo, aprenda plataformas populares com grandes comunidades.

Em seguida, examinamos as propriedades das plataformas de desenvolvimento de blockchain selecionadas. [Tabela 3](#) mostra essas propriedades fundamentais das plataformas de desenvolvimento blockchain. Novos desenvolvedores devem observar quais plataformas exigem o aprendizado de um novo idioma para escrever código de contrato inteligente. Fornecemos detalhes detalhados da linguagem de desenvolvimento em [Tabela 6](#). Da mesma forma, os desenvolvedores devem estar cientes da estrutura de permissão de suas plataformas de desenvolvimento. Os desenvolvedores que desejam criar projetos públicos, como um jogo baseado em blockchain, devem garantir o uso de um blockchain sem permissão.

Para dar aos desenvolvedores uma ideia de qual plataforma pode atender melhor seus projetos, fornecemos exemplos de aplicativos existentes em [Tabela 4](#). links

Tabela 5

Esta tabela destaca os recursos de desenvolvimento de nossas plataformas de desenvolvimento de blockchain selecionadas. Os links para a documentação de cada projeto, redes de teste e ferramentas de desenvolvimento estão em nosso repositório GitHub.

Plataforma	Pontuação de privacidade	Documentação	rede de teste	Desenvolvimento	modularidade Sybil
		pontuação		ferramentas	
Ethereum	1	31	Público	Sim	Nenhum
Hyperledger	7	7	Privado	Sim	Alto
Tecido					
EOS	5	31	Público	Sim	Baixo
solana	0	7	Público	Sim	Nenhum
hortelã	5	7	Privado	Sim	Nenhum
Quorum	7	7	Não	Sim	Nenhum
corda	7	15	Público	Não	Nenhum
neo	4	3	Público Privado	Sim	Nenhum
Tron	5	3	Público	Sim	Nenhum
Estelar	0	15	Público	Sim	Nenhum
Pilhas	0	3	Público	Sim	Nenhum
Cadeia BNB	2	7	Público	Sim	Nenhum
Hyperledger	5	3	Público	Sim	Alto
dente de serra					
APROXIMAR	7	7	Público Privado	Não	Nenhum
Tezos	3	19	Não	Sim	Nenhum
Otimismo	4	3	Público Privado	Sim	Nenhum
IoTeX	0	23	Privado	Não	Nenhum
Harmonia	3	23	Público	Sim	Nenhum
Ondas	0	3	Não	Não	Nenhum
Algorand	0	23	Privado	Sim	Nenhum
Quantum	0	7	Público	Sim	Nenhum
Exonum	5	3	Privado	Não	Nenhum
Zilliqa	0	3	Público	Sim	Nenhum

Tabela 6

Esta tabela mostra quais linguagens de programação podem ser usadas para desenvolver contratos inteligentes para cada plataforma. Os idiomas marcados com um * são idiomas específicos da blockchain.

Plataforma	Linguagem de desenvolvimento de contrato inteligente
Ethereum	Solidity*, Viper*
Tecido Hyperledger	Ir, Java, Javascript, Typescript C++
EOS	
solana	Ferrugem, C++, C
hortelã	Go, Python, Cosmos CLI*
Quorum	Solidity*
corda	Kotlin, Java
neo	Python, C#, Go, Java, Javascript, Typescript
Tron	Solidity*
Estelar	Javascript, Java, Go, Python, C#.NET, Ruby, IOS, Scala, Qt/C++, Flutter
Pilhas	Clarity*
Cadeia BNB	Solidez*
APROXIMAR	Javascript, Rust, NEAR API
Hyperledger Sawtooth	Python, Go, Javascript, Rust, Java, C++, Swift
Otimismo	Solidity*
Tezos	Python, OCaml, Javascript, Pascal, Reason, Indigo, Archtype, Michelson*
IoTeX	Solidity*, Javascript, Java, Go, Swift, C
Harmonia	Solidez*
Ondas	Andar de*
Quantum	Javascript, Solidez*
Algorand	Python, alcance
Exonum	Ferrugem, Java
Zilliqa	Scilla*

para os aplicativos de exemplo podem ser encontrados em nosso repositório GitHub. Os desenvolvedores devem usar esta tabela para determinar se uma plataforma cria sinergia com seu projeto. Por exemplo, os desenvolvedores que desejam criar um aplicativo fintech devem considerar plataformas projetadas para oferecer suporte a fintech.

Em seguida, revisamos várias considerações de desenvolvimento em [Tabela 5](#).

Recomendamos que os desenvolvedores revisem a documentação e o suporte oficial de cada plataforma, que fornecemos em nossa página do GitHub ([Connors, 2023](#)). Os desenvolvedores devem escolher uma plataforma que entendam e tenham certeza de que podem trabalhar. Algumas plataformas possuem um conjunto de ferramentas para auxiliar os desenvolvedores na criação de aplicativos. Os desenvolvedores que gostam de usar esses IDEs e estruturas de teste devem procurar usar uma plataforma com um conjunto de ferramentas diversificado. Links para as ferramentas oficiais fornecidas por cada plataforma podem ser encontrados em nossa página do GitHub.

Por fim, os desenvolvedores devem considerar os recursos de seus contratos inteligentes. Compreender como os contratos inteligentes são executados e implantados

é fundamental para a criação de código seguro. Em particular, os desenvolvedores que criam aplicativos fintech devem prestar muita atenção em como os contratos inteligentes são executados em uma plataforma blockchain. Um pequeno erro em sua implementação pode levar a grandes perdas financeiras.

Além de nossa tabela, colocamos algumas questões que os desenvolvedores devem se perguntar antes de escolher uma plataforma blockchain. Essas perguntas podem ajudar novos desenvolvedores a escolher a melhor plataforma de desenvolvimento de blockchain.

Primeiro, os desenvolvedores devem se perguntar se precisam de um blockchain sem permissão ou com permissão. Se um desenvolvedor planeja ter apenas alguns usuários selecionados, por exemplo, em um aplicativo de negócios, uma plataforma autorizada pode ser mais apropriada. Uma plataforma sem permissão será mais adequada ao projeto se um desenvolvedor planeja permitir que qualquer pessoa interaja com seu projeto. Os desenvolvedores devem se fazer perguntas como

Tabela 7

Esta tabela destaca os recursos de contratos inteligentes em nossas plataformas de desenvolvimento de blockchain selecionadas.

Plataforma	Diretamente atualizável	ricardiano contratos	Modelo de execução	modelo do sistema operacional	linguagem compilada
Ethereum	Não	Não	Ordem-Execução (OE)	Ethereum Máquina virtual	Ethereum Bytecode
Hyperledger Tecido	Sim	Sim	Executar-Ordem-Validar (EOV)	Hyperledger OS	–
EOS	Sim	Sim	OE	EÓCIO Máquina virtual	WebAssembly
solana	Sim	Não	OE: Execução Paralela	solana Máquina virtual	Solana Bytecode
hortelã	Não	Não	OE	Protocolo ABCI	ABCI
Quorum	Não	Não	OE	Ethereum Máquina virtual	Ethereum Bytecode
corda	Sim	Sim	Se Então Senão (EU TE)	Corde API	API REST
neo	Sim	Não	OE	Máquina Virtual Neo	Neo Bytecode
Tron	Não	Não	OE	Máquina Virtual Tron	Ethereum Bytecode
Estelar	Não	Não	EU TE	API estelar	API REST
Pilhas	Não	Não	EOV	Clarity Intérprete	–
Cadeia BNB	Não	Não	OE	Ethereum Máquina virtual	Ethereum Bytecode
Hyperledger dente de serra	Sim	Não	OE	dente de serra Máquinas de estado	WebAssembly
APROXIMAR	Não	Não	OE	PERTO da Máquina Virtual	WebAssembly
Tezos	Não	Não	OE: Funcional Programação	Michelson Funcional Programação	Michelson
Otimismo	Não	Não	OE	Otimismo Máquina virtual	Ethereum Bytecode
IoTeX	Não	Não	OE	Ethereum Máquina virtual	Ethereum Bytecode
Harmonia	Não	Não	OE	Ethereum Máquina virtual	Ethereum Bytecode
Ondas	Não	Não	EU TE	API de ondas	API REST
Algorand	Sim	Não	OE	Algorand Máquina virtual	CERCETA
QuantName	Não	Não	OE	máquina virtual x86	Montagem x86
Exonum	Sim	Não	OE	Máquina Virtual JAVA	Java Bytecode
Zilliqa	Não	Não	EOV	Scilla Intérprete	–

"Alguém deve ter permissão para interagir com o contrato inteligente?" ou "O código do contrato inteligente precisa ser confidencial?"

Da mesma forma, os desenvolvedores devem considerar se precisam de uma rede principal pública. Os desenvolvedores devem entender as taxas e a velocidade da transação usando uma rede pública. Por outro lado, se os desenvolvedores optarem por criar sua própria rede principal privada, eles devem entender o custo de criar e manter essa rede. Se uma empresa estiver rastreando remessas, provavelmente desejará manter esses dados ocultos dos concorrentes; assim, o desenvolvedor deve criar uma rede privada. Por outro lado, se um desenvolvedor não estiver armazenando informações confidenciais no blockchain, uma rede pública pode ser mais acessível. Perguntas como "Quero usar uma infraestrutura de blockchain existente?", "Qual é o máximo que qualquer execução do meu contrato inteligente pode custar?" são essenciais para tomar essa decisão.

Em seguida, os desenvolvedores devem garantir que a plataforma permita que eles criem seus projetos. Por exemplo, um desenvolvedor que deseja criar um aplicativo bancário deve usar uma plataforma que outros já usaram para criar aplicativos semelhantes. Os desenvolvedores devem considerar algumas plataformas para qualquer projeto, pois uma plataforma pode ser melhor otimizada para o projeto do desenvolvedor. Por exemplo, um desenvolvedor pode desejar usar o Ethereum para um aplicativo bancário. No entanto, eles também devem considerar plataformas como Stellar, pois são mais otimizadas para esse tipo de projeto. "Ter

aplicativos semelhantes foram criados usando esta plataforma de desenvolvimento blockchain?" e "Qual plataforma de desenvolvimento blockchain é especializada para o meu caso de uso?" podem ajudar os desenvolvedores a escolher a melhor plataforma.

Os desenvolvedores devem garantir que se sintam confortáveis com a linguagem em que os contratos inteligentes são desenvolvidos. Conhecer as nuances da linguagem é particularmente importante na hora de criar aplicações financeiras, pois um erro no código pode causar prejuízos financeiros. Para começar rapidamente, os desenvolvedores podem querer escrever em idiomas que já conhecem. No entanto, se os desenvolvedores gastarem tempo aprendendo uma linguagem específica de estrutura projetada para o desenvolvimento de contratos inteligentes, isso pode facilitar a programação de tarefas de blockchain mais complicadas. Os desenvolvedores devem considerar: "Já conheço a linguagem para o desenvolvimento de contratos inteligentes?" "Estou disposto a aprender uma nova linguagem específica de blockchain?"

Da mesma forma, os desenvolvedores precisam entender quais tipos de ferramentas de desenvolvimento estão disponíveis. As ferramentas de desenvolvimento podem ajudar os desenvolvedores a criar, testar e manter contratos inteligentes. Os desenvolvedores devem entender "Quais ferramentas existem para ajudar os desenvolvedores?" e "Que tipo de suporte existe para novos desenvolvedores?" Uma grande comunidade ativa pode ajudar os desenvolvedores com problemas técnicos ao criar seus dapps. Os desenvolvedores devem

Tabela 8

Esta tabela destaca as comunidades que cercam nossas plataformas de desenvolvimento de blockchain selecionadas. Os últimos dados de confirmação foram coletados em setembro de 2022.

Plataforma	Apoiadores	Durar comprometer-se	Total comete	Roteiro	Tele Oficial grama/Discord	YouTube vídeos	Subreddit
Ethereum	Vitalik Buterin, Fundação Ethereum	Set 2022	13637	Sim	Sim	1094	1444724
Hyperledger Tecido	Fundação Linux, IBM, Intel, SAP Arábia	Set 2022	14130	Não	Sim	1169	3550
EOS	Bloco um, Fundação EOSIO	Julho 2021	20554	Sim	Sim	103	97768
solana	Anatoly Yakovenko, Fundação Solana	Set 2022	20203	Sim	Sim	314	152831
hortelã	Jae Kwon, Cosmos/Ignite	Set 2022	8956	Sim	Sim	359	274
Quorum	ConsenSys, JP Morgan	Set 2022	14311	Não	Não	83	2131
corda	R3	Set 2022	9732	Sim	Não	257	469
neo	Da Hongfei, Erik Zhang, Onchain	Set 2022	1388	Sim	Não	148	117601
Tron	Justin Sun, TRON DAO	Set 2022	17087	Não	Sim	559	124952
Estelar	Jed McCaleb, Joyce Kim, Stellar Development Foundation	Set 2022	7990	Sim	Sim	143	212769
Pilhas	Princeton, Hiro, Stacks Foundation	Set 2022	16993	Não	Sim	494	6652
Cadeia BNB	Binance	Set 2022	13506	Sim	Sim	135	17192
Hyperledger dente de serra	Fundação Linux, IBM, Intel, SAP Arábia	julho de 2022	4590	Não	Sim	1169	3550
APPROXIMAR	Illia Polosukhin, Alexander Skidanov, Fundação NEAR	Set 2022	8251	Sim	Sim	378	12254
Tezos	Soluções de contabilidade dinâmica, Tezos Foundation	Set 2022	3779	Não	Sim	171	71141
Otimismo	Fundação de Otimismo	Set 2022	19477	Sim	Sim	17	5
IoTeX	IoTeX	Set 2022	2831	Sim	Sim	360	18044
Harmonia	Stephen Tse, Fundação Harmonia	Set 2022	7590	Sim	Sim	406	53477
Ondas	Tecnologia de Ondas	Set 2022	12971	Não	Sim	86	1484
Algorand	Algorand	Set 2022	34919	Não	Sim	183	75645
Quantum	Quantum	junho de 2022	3270	Sim	Sim	98	17679
Exonum	Bitfury	Set 2022	5652	Sim	Sim	7	46
Zilliqa	Universidade Nacional de Singapura, Fundação Zilliqa	Set 2022	9383	Sim	Sim	132	44459

considere questões como "Quão popular é a plataforma de desenvolvimento blockchain?" ou "A comunidade é grande o suficiente para ajudar novos desenvolvedores?"

7. Conclusão

Neste trabalho, primeiro descrevemos os fundamentos do blockchain para que os desenvolvedores possam entender o que diferencia várias plataformas de desenvolvimento de blockchain. Em seguida, discutimos a seleção de 23 plataformas de desenvolvimento de blockchain de nossa extensa lista. Em seguida, analisamos sete tabelas destacando as principais métricas que os desenvolvedores devem conhecer para 23 plataformas selecionadas. Terminamos com uma breve discussão sobre como os desenvolvedores podem usar efetivamente nossas métricas ao criar novos projetos. Esperamos que os desenvolvedores usem nossas métricas para selecionar a melhor plataforma de projeto.

Em trabalhos futuros, pretendemos criar o mesmo projeto de amostra em todas as nossas plataformas selecionadas e publicar esses projetos no GitHub. Esta página do GitHub permitirá que os desenvolvedores analisem o código para entender melhor as nuances de cada projeto. Esperamos que a adição de código ajude novos desenvolvedores a selecionar uma plataforma e iniciar rapidamente os projetos.

Declaração de interesse concorrente

Os autores declaram que não têm interesses financeiros concorrentes conhecidos ou relacionamentos pessoais que possam parecer influenciar o trabalho relatado neste artigo.

Disponibilidade de dados

Nenhum dado foi utilizado para a pesquisa descrita no artigo.

Referências

- Agbo, Corneilus, Mahmoud, Qusay, 2019. Comparação de estruturas blockchain para aplicações de saúde. *Tecnologia da Internet*. Deixe 2 (5), e122.
- Ali, Muneeb, 2020. Aplicativos Stacks 2.0 e contratos inteligentes para Bitcoin. *Gaia.Blockstack.Org*.
- Androulaki, Elli, Barger, Artem, Bortnikov, Vita, Cachin, Christian, Christidis, Konstantinos, De Caro, Angelo, Enyeart, David, Ferris, Christopher, Laventman, Gennady, Manevich, Yacov et al., 2018. Hyperledger fabric. In: *Actas da Décima Terceira Conferência EuroSys*.
- Ano, 0000a. Neo Livro Branco. Site Neo. Anônimo, 0000b.
- Site da Lisk. Livro Branco da Lisk. Anon, 2017. *O documento técnico ZILLIQA*. Zilliqa.

- Ben Sasson, Eli, Chiesa, Alessandro, Garman, Christina, Green, Matthew, Miers, Ian, Tromer, Eran, Virza, Madars, 2014. Zerocash: Pagamentos anônimos descentralizados de Bitcoin. In: 2014 IEEE Symposium on Security and Privacy.
- BNB-Chain, 2020. Documento técnico da BNB-chain. GitHub.
- Brown, Richard Gendal, 2018. A plataforma Corda: uma introdução.
- Buterin, Vitalik, 2014. Ethereum Whitepaper.
- Chen, Jing, Micali, Silvio, 2016. Algorand. <http://dx.doi.org/10.48550/ARXIV.1607.01341>, arXiv.
- Connors, Colin, 2023. <https://github.com/CollinConnors/Survey-Of-Prominent-Blockchain-Desenvolvimento-Plataformas>.
- Connors, Collin, Sarkar, Dilip, 2022a. Estudo comparativo do desenvolvimento de blockchain plataformas: Recursos e aplicações.
- Connors, Collin, Sarkar, Dilip, 2022b. Revisão das plataformas de código aberto mais populares para o desenvolvimento de blockchains. Em: 2022 Quarta Conferência Internacional sobre Blockchain Computing and Applications. BCCA.
- ConsenSys, 2018. Artigo técnico do quórum.
- David, Bernardo, Gazi, Peter, Kiayias, Aggelos, Russell, Alexander, 2023. Ouroboros Praos: Um blockchain proof-of-stake adaptavelmente seguro e semi-síncrono. Dernaika, Iman, Chehab, Ali, 2021. Plataformas de desenvolvimento Blockchain: Desempenho comparação. In: 2021 11ª Conferência Internacional do IFIP sobre Novas Tecnologias, Mobilidade e Segurança. NTMS, pp. 1–6.
- Goldreich, Oded, Oren, Yair, 1994. Definições e propriedades da prova de conhecimento zero sistemas. J. Cryptol. 7 (1), 1–32.
- Goodman, LM, 2014. Tezos - um livro branco de livro-caixa de criptografia que se autocorrige.
- Grigg, I., 2004. O contrato ricardiano. Em: Processos. Primeiro IEEE Internacional Workshop sobre Contratação Eletrônica, 2004, pp. 25–31.
- Guo, Huaqun, Yu, Xingjie, 2022. Uma pesquisa sobre a tecnologia Blockchain e sua segurança. Blockchain: Res. Appl. 3 (2), 100067.
- Hearn, Mike, Brown, Richard Gendal, 2019. Corda: Um livro-razão distribuído.
- Fundação Hyperledger, 2018. Introdução ao Hyperledger, Hyperledger.
- Iqbal, Mubashar, Matulevičius, Raimundas, 2021. Explorando sybil e gastos duplos riscos em sistemas blockchain. Acesso IEEE 9, 76153–76177.
- Javaid, Mohd, Haleem, Abid, Pratap Singh, Ravi, Khan, Shahbaz, Suman, Rajiv, 2021. Aplicações da tecnologia Blockchain para a Indústria 4.0: Uma revisão baseada na literatura. Blockchain: Res. Appl. 2 (4), 100027.
- Kwon, Jae, 2014. Tendermint: Consenso sem mineração. Tendermint. Com.
- Larimer, Daniel, et al., 2018. eos.io white paper técnico v2.
- Mazieres, David, 2016. O protocolo de consenso estelar: um modelo federado para Consenso no nível da Internet.
- Mingxiao, Du, Xiaofeng, Ma, Zhe, Zhang, Xiangwei, Wang, Qijun, Chen, 2017. Uma revisão no algoritmo de consenso do blockchain. In: Conferência Internacional IEEE 2017 sobre Sistemas, Homem e Cibernética. SMC, pp. 2567–2572.
- Nair, P. Rajitha, Dorai, D. Ramya, 2021. Avaliação de desempenho e segurança de prova de trabalho e prova de aposta usando blockchain. In: 2021 Terceira Conferência Internacional sobre Tecnologias Inteligentes de Comunicação e Redes Móveis Virtuais. ICICV, pp. 279–283.
- Nakamoto, Satoshi, 2008. Bitcoin: Um sistema de dinheiro eletrônico peer-to-peer. SSRN Eletrônico J..
- NEAR Foundation, 2022. O white paper NEAR. Protocolo NEAR.
- Nguyen, Giang-Truong, Kim, Kyungbaek, 2018. Uma pesquisa sobre algoritmos de consenso usado em blockchain. Koreascience.Or.Kr.
- Nofer, Michael, Gomber, Peter, Hinz, Oliver, Schiereck, Dirk, 2017. Blockchain. Ônibus. Informar. Sist. Eng. 59 (3), 183–187.
- Olson, Kelly, Bowman, Mic, Mitchell, James, Amundson, Shawn, Middleton, Dan, Montgomery, Cian, 2018. Sawtooth: An Introduction, Hyperledge Org. Equipe Otimismo, 2023. Como funciona o otimismo. Otimismo Documentos.
- Polge, Julien, Robert, Jérémy, Le Traon, Yves, 2021. Blockchain permitido frameworks na indústria: Uma comparação. ICT Express 7 (2), 229–233. Popov, Serguei, 2018. O Emaranhado.
- Qtum Org, 2020. Qtum Blockchain New Whitepaper.
- Quasim, Mohammad Tabrez, Khan, Mohammad Ayoub, Algarni, Fahad, Alharthy, Abdullah, Alshmrani, Goram Mufareh M., 2020. Estruturas Blockchain. In: Khan, Mohammad Ayoub, Quasim, Mohammad Tabrez, Algarni, Fahad, Alharthy, Abdullah (Eds.), Decentralized Internet of Things: A Blockchain Perspective. Springer International Publishing, Cham, pp. 75–89.
- Sekniqi, Kevin, Laine, Daniel, Gun Sirer, Emin, 2020. White paper da plataforma Avalanche, Assets.Website-Files.Com.
- Sun, Xiaoqiang, Yu, F. Richard, Zhang, Peng, Sun, Zhiwei, Xie, Weixin, Peng, Xiang, 2021. Uma pesquisa sobre prova de conhecimento zero em blockchain. Rede IEEE 35 (4), 198–205.
- Tavares, Bruno, Correia, Filipe, Restivo, André, 2019. A Survey on Blockchain Tecnologias e Pesquisa. Vol. 14, pp. 118–128.
- The Harmony Team, 2019. Papel técnico do Harmony. Harmony.One.
- The Iotex Team, 2018. IoTeX Uma rede descentralizada para inter de coisas alimentada por um blockchain centrado na privacidade. O banco de dados do whitepaper.
- TRON, 2018. Artigo técnico Tron. Whitepaper.Io - Pesquise e encontre todos os whitepapers em Whitepaper.Io.
- Valenta, Martin, Sandner, Philipp, 2017. Comparação de ethereum, tecido hyperledger e corda. Smallake.Kr.
- Velliangiri, S., Karthikeyan, P., 2020. Tecnologia Blockchain: Desafios e Segurança questões no algoritmo Consensus. In: Conferência Internacional de Comunicação e Informática de Computadores 2020. ICCCI, pp. 1–8.
- Waves Enterprise, 2022. Plataforma Blockchain híbrida de nível empresarial.
- Yaga, Dylan, Mell, Peter, Roby, Nik, Scarfone, Karen, 2018. Tecnologia Blockchain visão geral. <http://dx.doi.org/10.6028/nist.ir.8202>.
- Yakovenko, Anatoly, 2018. Solana: Uma nova arquitetura para um alto desempenho blockchain v0.8.13. Solana. Com.
- Yanovich, Yury, Ivaschenko, Ivan, Ostrovsky, Alex, Shevchenko, Aleksandr, Sidorov, Aleksei, 2018. Exonum: protocolo tolerante a falhas bizantino para blockchains.
- Zhang, Rong, Chan, Wai Kin, 2020. Avaliação do consumo de energia em blockchains com prova de trabalho e prova de participação. J. Phys. conf. Ser. 1584 (1), 012023.
- Zou, Weiqin, Lo, David, Kochhar, Pavneet Singh, Le, Xuan-Bach Dinh, Xia, Xin, Feng, Yang, Chen, Zhenyu, Xu, Baowen, 2021. Desenvolvimento de contratos inteligentes: desafios e oportunidades. IEEE Trans. Softw. Eng. 47 (10), 2084–2106.

Colin Connors é um Ph.D. candidato em Ciência da Computação na Universidade de Miami. Sua pesquisa está focada na segurança de tecnologias emergentes, incluindo blockchain e aplicações de inteligência artificial em segurança cibernética.

Dilip Sarkar é professor de Ciência da Computação na Universidade de Miami. Suas áreas de pesquisa incluem comunicação sem fio, transporte multimídia pela Internet, redes de sensores sem fio, computação distribuída e computação paralela.