

:~\$ Scalar School

Filosofia do Desenvolvimento Bitcoin



Filosofia do desenvolvimento do Bitcoin

Kalle Rosenbaum, Linnéa Rosenbaum

Table of Contents

Sobre este livro	1
O que esperar?	1
Quem escreveu isto?	1
Como isso está organizado?	2
1. Descentralização	3
1.1. Descentralização dos mineradores	3
1.2. Descentralização de nós completos	6
1.3. Neutralidade	7
1.4. Compreendendo a descentralização	8
1.5. Conclusão	10
2. Confiança Zero	11
2.1. Não confie, verifique	14
2.2. Conclusão	17
3. Privacidade	18
3.1. O que significa privacidade?	18
3.2. Por que a privacidade é importante?	19
3.3. Pseudonimidade	20
3.4. Privacidade na blockchain	21
3.5. Privacidade fora da blockchain	23
3.6. Fungibilidade	24
3.7. Medidas de privacidade	25
3.8. Conclusão	27
4. Emissão finita	28
4.1. Subsídio de bloco e taxas de transação	30
4.2. Conclusão	32
5. Atualizações	33
5.1. Vocabulário	34
5.2. Atualizações históricas	35
5.2.1. Atualização do Segwit	36
5.2.2. Discussão pós-Segwit	38
5.2.3. Atualização do Taproot - Speedy Trial	40
5.2.4. Mecanismos de implantação futuros	43
5.3. Riscos	43
5.3.1. Custos de uma divisão	44
5.4. Conclusão	45
6. Open Source	46
6.1. Manutenção do software	48
6.2. Desenvolvimento sem permissões	49

6.3. Desenvolvimento pseudônimo	50
6.4. Criptografia de seleção	51
6.5. Revisão	52
6.6. Financiamento	54
6.7. Choque cultural	55
6.8. Conclusão	56
7. Escalabilidade	57
7.1. História	58
7.2. Abordagens de escalabilidade	59
7.2.1. Escalabilidade vertical	59
7.2.2. Escalabilidade horizontal	60
7.2.3. Escalabilidade interna	61
7.2.4. Escalabilidade em camadas	64
7.3. Conclusão	66
8. Quando dá tudo errado	67
8.1. Divulgação responsável	67
8.2. Infância traumática	71
8.2.1. 2010-07-28: Gastar moedas de qualquer pessoa (CVE-2010-5141)	71
8.2.2. 2010-08-15 Transbordo de saída combinada (CVE-2010-5139)	74
8.2.3. 2013-03-11 Problema de bloqueios de banco de dados 0.7.2 - 0.8.0 (CVE-2013-3220)	77
8.2.4. BIP66	79
8.3. Conclusão	83
Appendix A: Perguntas para Discussão	84
A.1. Descentralização	84
A.2. Confiança Nula	84
A.3. Privacidade	84
A.4. Oferta Finita	84
A.5. Atualizações	85
A.6. Pensamento Adversarial	85
A.7. Código Aberto	85
A.8. Escalabilidade	85
Appendix B: Feedback e contribuição	87
B.1. Build	87
B.1.1. Manualmente usando <code>asciidoctor</code>	87
B.1.2. Usando Gnu <code>make</code>	88
B.1.3. Construir um PDF	88

Sobre este livro

Bitcoin Development Philosophy é um guia para desenvolvedores de Bitcoin que já entendem os conceitos básicos e processos como Proof-of-Work, construção de blocos e o ciclo de vida das transações, e que desejam avançar adquirindo uma compreensão mais profunda dos trade-offs de design do Bitcoin e sua filosofia. Deve ajudar novos desenvolvedores a absorver as lições mais importantes de mais de uma década de desenvolvimento e debate público do Bitcoin, ao mesmo tempo que fornece um contexto útil para avaliar novas ideias (boas e ruins!).

Visão geral do índice:

Chapter 1, *Descentralização*

Chapter 2, *Confiança Zero*

Chapter 3, *Privacidade*

Chapter 4, *Emissão finita*

Chapter 5, *Atualizações*

[adversarialthinking]

Chapter 6, *Open Source*

Chapter 7, *Escalabilidade*

Chapter 8, *Quando dá tudo errado*

Appendix A, *Perguntas para Discussão*

Appendix B, *Feedback e contribuição*

Seu feedback e contribuições são muito bem-vindos! Instruções para construção e contribuição podem ser encontradas em [Appendix B](#).

O que esperar?

Como afirmado acima, este é um guia prático para desenvolvedores de Bitcoin. No entanto, o Bitcoin é um assunto amplo e complexo e não poderíamos cobrir todos os seus aspectos aqui. Com este livro, esperamos discutir os recursos necessários para iniciar sua atividade de desenvolvimento, bem como permitir que você explore isso por conta própria.

Existem muitas pessoas envolvidas no Bitcoin; como algumas delas têm opiniões opostas, aqui você pode encontrar recursos que expressam ideias contraditórias. No entanto, sempre tentamos nos ater ao domínio dos fatos, onde as opiniões não importam.

Quem escreveu isto?



O autor principal deste livro é Kalle Rosenbaum, e Linnéa Rosenbaum contribuiu como coautora. Este trabalho foi encomendado e financiado por [Chaincode Labs](#), um centro de desenvolvimento

que oferece programas educacionais para desenvolvedores que desejam aprender sobre o desenvolvimento do Bitcoin.



Kalle é o autor de [Grokking Bitcoin](#) (Manning Publications) e é um desenvolvedor de software experiente. Ele trabalha profissionalmente com desenvolvimento relacionado ao Bitcoin desde 2015.



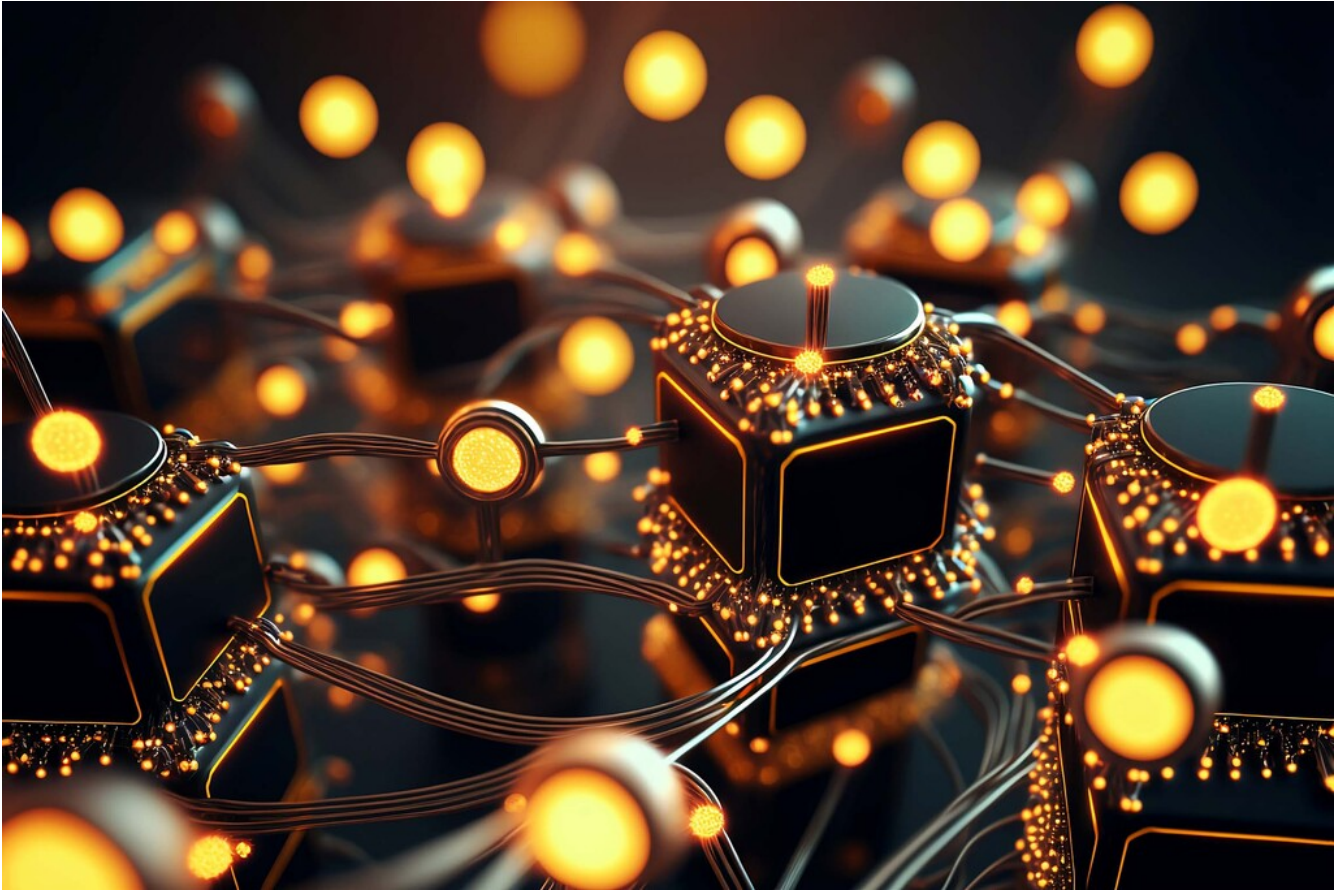
Linnéa tem um Ph.D. em Sistemas Eletrônicos. Ela é a tradutora sueca de [The Little Bitcoin Book](#) e co-tradutora de [The Bitcoin Standard](#). Ela também faz parte do conselho da Associação Sueca de Bitcoin. Seu histórico é em desenvolvimento de firmware, com uma mudança recente para desenvolvimento de software.

Como isso está organizado?

O livro está dividido em capítulos, cada um cobrindo um tópico fundamental dentro do Bitcoin. Cada capítulo guia você pelas informações relevantes relacionadas ao foco do capítulo, com a adição de links e códigos QR que levarão você a artigos ou vídeos que recomendamos ler ou assistir. O material reportado foi escrito por indivíduos que estudam o desenvolvimento do Bitcoin há muito tempo.

Os links referem-se a recursos externos em plataformas que não podemos controlar. Portanto, salvamos os artigos vinculados localmente neste repositório, junto com as informações de onde foram copiados e quando. Os recursos estão coletados em [um documento separado](#) ([sources/sources.adoc](#)) e organizados pelo capítulo ao qual estão vinculados. Os links encontrados nos capítulos referem-se às fontes originais, que devem ser encontradas online, mas caso você não tenha conexão à internet, os links apareçam mortos ou as informações pareçam severamente alteradas, você pode ler o conteúdo localmente em vez disso.

Chapter 1. Descentralização



Este capítulo analisa o que é descentralização e por que ela é essencial para o funcionamento do Bitcoin. Distinguimos entre a descentralização de mineradores e a de nós completos, e discutimos o que eles trazem para a resistência à censura, uma das propriedades mais centrais do Bitcoin. A discussão então muda para entender a neutralidade - ou a ausência de necessidade de permissão para usuários, mineradores e desenvolvedores - que é uma propriedade necessária de qualquer sistema descentralizado. Por fim, abordamos como pode ser difícil compreender um sistema descentralizado como o Bitcoin e apresentamos alguns modelos mentais que podem ajudar a entendê-lo.

Um sistema sem nenhum ponto central de controle é chamado de *descentralizado*. O Bitcoin é projetado para evitar ter um ponto central de controle, ou mais precisamente, um *ponto central de censura*. A descentralização é um meio para alcançar a *resistência à censura*.

Existem dois principais aspectos da descentralização no Bitcoin: descentralização dos mineradores e descentralização dos nós completos. A descentralização dos mineradores refere-se ao fato de que o processamento das transações não é realizado nem coordenado por nenhuma entidade central. A descentralização dos nós completos refere-se ao fato de que a validação dos blocos, ou seja, os dados que os mineradores produzem, é feita na borda da rede, em última instância pelos seus usuários, e não por algumas autoridades confiáveis.

1.1. Descentralização dos mineradores

Houve tentativas de criar moedas digitais antes do Bitcoin, mas a maioria delas falhou devido à

falta de descentralização na governança e resistência à censura.

A descentralização dos mineradores no Bitcoin significa que a *ordenação das transações* não é realizada por nenhuma entidade única ou conjunto fixo de entidades. É realizada coletivamente por todos os atores que desejam participar disso; esse coletivo de mineradores é um conjunto dinâmico de usuários. Qualquer pessoa pode entrar ou sair conforme desejar. Esta propriedade torna o Bitcoin resistente à censura.



Se o Bitcoin fosse centralizado, ele seria vulnerável àqueles que desejassem censurá-lo, como governos. Ele teria o mesmo destino que as tentativas anteriores de criar dinheiro digital. Na introdução de [um artigo](#) intitulado “Enabling Blockchain Innovations with Pegged Sidechains”, os autores explicam como as primeiras versões de dinheiro digital não estavam preparadas para um ambiente adversarial (veja também [\[adversarialthinking\]](#)):

David Chaum introduziu o dinheiro digital como um tópico de pesquisa em 1983, em um cenário com um servidor central que é confiável para prevenir o gasto duplo [Cha83]. Para mitigar o risco à privacidade dos indivíduos por parte desse servidor central confiável, e para garantir a fungibilidade, Chaum introduziu a assinatura cega, que ele usou para fornecer um método criptográfico para evitar a vinculação das assinaturas do servidor central (que representam moedas), enquanto ainda permitia que o servidor central realizasse a prevenção do gasto duplo. A exigência de um servidor central tornou-se o calcanhar de Aquiles do dinheiro digital [Gri99]. Embora seja possível distribuir esse ponto único de falha substituindo a assinatura do servidor central por uma assinatura threshold de vários signatários, é importante para a auditabilidade que os signatários sejam distintos e identificáveis. Isso ainda deixa o sistema vulnerável a falhas, pois cada signatário pode falhar, ou ser forçado a falhar, um por um.

— various authors, Enabling Blockchain Innovations with Pegged Sidechains (2014)

Ficou claro que usar um servidor central para ordenar transações não era uma opção viável devido ao alto risco de censura. Mesmo que alguém substituísse o servidor central por uma federação de um conjunto fixo de n servidores, dos quais pelo menos m devem aprovar uma ordenação, ainda haveria dificuldades. O problema, de fato, se deslocaria para um cenário onde os usuários precisam concordar com esse conjunto de n servidores assim como sobre como substituir servidores maliciosos por bons sem depender de uma autoridade central.

Vamos contemplar o que poderia acontecer se o Bitcoin fosse censurável. O censor poderia pressionar os usuários a se identificarem, a declarar de onde vem seu dinheiro ou o que estão comprando com ele antes de permitir que suas transações entrem na blockchain.

Além disso, a falta de resistência à censura permitiria que o censor coagisse os usuários a adotar

novas regras do sistema. Por exemplo, eles poderiam impor uma mudança que permitisse inflacionar a oferta de dinheiro, enriquecendo assim a si próprios. Em tal caso, um usuário que verifica blocos teria três opções para lidar com as novas regras:

- Adotar: Aceitar as mudanças e adotá-las em seu nó completo.
- Rejeitar: Recusar-se a adotar as mudanças; isso deixaria o usuário com um sistema que não processa mais transações, pois os blocos do censor agora seriam considerados inválidos pelo nó completo do usuário.
- Mover: Nomear um novo ponto central de controle; todos os usuários devem descobrir como coordenar e depois concordar com o novo ponto de controle central. Se eles tiverem sucesso, os mesmos problemas provavelmente ressurgirão em algum momento no futuro, considerando que o sistema permaneceu tão censurável quanto era antes.

Nenhuma dessas opções é benéfica para o usuário.



A resistência à censura através da descentralização é o que diferencia o Bitcoin de outros sistemas monetários, mas não é uma coisa fácil de realizar devido ao *problema do gasto duplo*. Este é o problema de garantir que ninguém possa gastar a mesma moeda duas vezes, uma questão que muitas pessoas pensavam ser impossível de resolver de maneira descentralizada. Satoshi Nakamoto escreve em seu [whitepaper do Bitcoin](#) sobre como resolver o problema do gasto duplo:

Neste artigo, propomos uma solução para o problema do gasto duplo usando um servidor de timestamp distribuído peer-to-peer para gerar prova computacional da ordem cronológica das transações.

— Satoshi Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System (2008)



Aqui ele usa a frase peculiar “servidor de timestamp distribuído peer-to-peer”. A palavra-chave aqui é *distribuído*, o que neste contexto significa que não há ponto central de controle. Nakamoto então prossegue explicando como a prova de trabalho é a solução. Ainda assim, ninguém explica isso melhor do que [Gregory Maxwell no Reddit](#), onde ele responde a alguém que propõe limitar o poder de hash dos mineradores para evitar possíveis ataques de 51%:

Um sistema descentralizado como o Bitcoin usa uma eleição pública. Mas você não pode simplesmente ter um voto de 'pessoas' em um sistema descentralizado porque isso exigiria uma parte centralizada para autorizar as pessoas a votar. Em vez disso, o Bitcoin usa um voto de poder computacional porque é possível verificar o poder computacional sem a ajuda de qualquer terceira parte centralizada.

— Gregory Maxwell, r/Bitcoin subreddit (2019)

A postagem explica como a rede descentralizada do Bitcoin pode chegar a um acordo sobre a ordenação de transações através do uso da prova de trabalho. Ele então conclui dizendo que o ataque de 51% não é particularmente preocupante, em comparação com as pessoas não se importarem ou não entenderem as propriedades de descentralização do Bitcoin.

Um risco muito maior para o Bitcoin é que o público que o usa não entenda, não se importe e não proteja as propriedades de descentralização que o tornam valioso em relação às alternativas centralizadas.

— Gregory Maxwell, r/Bitcoin subreddit (2019)

A conclusão é importante. Se as pessoas não protegerem a descentralização do Bitcoin, que é um proxy para sua resistência à censura, o Bitcoin pode cair vítima de poderes centralizadores, até que esteja tão centralizado que a censura se torne uma realidade. Então, a maior parte, senão toda, de sua proposta de valor desaparece. Isso nos leva à próxima seção sobre descentralização de nós completos.

1.2. Descentralização de nós completos

Nos parágrafos acima, falamos principalmente sobre a descentralização dos mineradores e como a centralização dos mineradores pode permitir a censura. Mas há também outro aspecto da descentralização, a saber, a *descentralização de nós completos*.

A importância da descentralização dos nós completos está relacionada à confiança nula (veja [Chapter 2](#)). Suponha que um usuário pare de executar seu próprio nó completo devido, por exemplo, a um aumento proibitivo no custo de operação. Nesse caso, ele terá que interagir com a rede Bitcoin de alguma outra forma, possivelmente usando carteiras web ou carteiras leves, o que exige um certo nível de confiança nos provedores desses serviços. O usuário passa de impor diretamente as regras de consenso da rede para confiar que alguém mais o fará. Agora suponha que a maioria dos usuários delegue a imposição do consenso a uma entidade confiável. Nesse caso, a rede pode rapidamente entrar em uma espiral de centralização, e as regras da rede podem ser alteradas por atores maliciosos conspirando.



Em [um artigo da Bitcoin Magazine](#), Aaron van Wirdum entrevista desenvolvedores de Bitcoin sobre suas opiniões sobre descentralização e os riscos envolvidos no aumento do tamanho máximo do bloco do Bitcoin. Essa discussão foi um tópico quente durante a era de 2014-2017, quando muitas pessoas discutiam sobre o aumento do limite de tamanho do bloco para permitir maior throughput de transações.

Um argumento poderoso contra o aumento do tamanho do bloco é que ele aumenta o custo de verificação (veja [o capítulo de Escalabilidade](#)). Se o custo de verificação aumentar, isso levará alguns usuários a parar de executar seus nós completos. Isso, por sua vez, levará a mais pessoas não conseguirem usar o sistema de forma confiável. Pieter Wuille é citado no artigo, onde ele explica os riscos da centralização dos nós completos.

Se muitas empresas executarem um nó completo, isso significa que todas precisarão ser convencidas a implementar um conjunto de regras diferente. Em outras palavras: a descentralização da validação de blocos é o que dá peso às regras de consenso. Mas se o número de nós completos cair muito, por exemplo porque todos usam as mesmas carteiras web, exchanges e carteiras SPV ou móveis, a regulamentação pode se tornar uma realidade. E se as autoridades puderem regulamentar as regras de consenso, significa que podem mudar qualquer coisa que faz o Bitcoin ser Bitcoin. Até mesmo o limite de 21 milhões de bitcoins.

— Pieter Wuille, *The Decentralist Perspective or Why Bitcoin Might Need Small Blocks* (2015)

Aí está. Os usuários de Bitcoin devem executar seus próprios nós completos para dissuadir reguladores e grandes corporações de tentar mudar as regras de consenso.

1.3. Neutralidade

O Bitcoin é neutro, ou sem necessidade de permissão, como as pessoas gostam de chamar. Isso significa que o Bitcoin não se importa com quem você é ou para que você o usa.

o bitcoin é neutro, o que é uma coisa boa, e a única maneira como ele pode funcionar. se fosse controlado por uma organização, seria apenas mais um tipo de objeto virtual e eu não teria nenhum interesse nele

— wumpus on freenode IRC (pontuação adicionada), #bitcoin-core-dev 2012-04-04T17:34:04 UTC

Desde que você jogue pelas regras, você é livre para usá-lo como quiser, sem pedir permissão a ninguém. Isso inclui *mineração*, *transações* e *construção de protocolos e serviços* em cima do Bitcoin.

- Se a **mineração** fosse um processo com necessidade de permissão, precisaríamos de uma autoridade central para selecionar quem tem permissão para minerar. Isso provavelmente levaria a mineradores tendo que assinar contratos legais nos quais concordariam em censurar transações de acordo com os caprichos da autoridade central, o que anularia o propósito da mineração em primeiro lugar.
- Se as pessoas que **transacionam** em Bitcoin tivessem que fornecer informações pessoais, declarar para que servem suas transações ou de outra forma provar que são dignas de transacionar, também precisaríamos de um ponto central de autoridade para aprovar usuários ou transações. Novamente, isso levaria à censura e exclusão.
- Se os desenvolvedores tivessem que pedir permissão para **construir protocolos** em cima do Bitcoin, apenas os protocolos permitidos pelo comitê central de desenvolvimento seriam desenvolvidos. Isso, devido à intervenção do governo, inevitavelmente excluiria todos os protocolos que preservam a privacidade e todas as tentativas de melhorar a descentralização.

Em todos os níveis, tentar impor restrições sobre quem pode usar o Bitcoin para o quê prejudicará

o Bitcoin a ponto de ele não mais corresponder à sua proposta de valor.



Pieter Wuille [responde uma pergunta no Stack Exchange](#) sobre como a blockchain se relaciona com bancos de dados normais. Ele explica como a ausência de necessidade de permissão é alcançada através do uso da prova de trabalho em combinação com incentivos econômicos. Ele conclui:

Usar algoritmos de consenso sem confiança como PoW realmente adiciona algo que nenhuma outra construção oferece (participação sem necessidade de permissão, ou seja, não há um grupo definido de participantes que pode censurar suas mudanças), mas vem a um custo alto, e suas suposições econômicas fazem com que seja praticamente útil apenas para sistemas que definem sua própria criptomoeda. Provavelmente há espaço no mundo para apenas um ou alguns desses sistemas realmente usados.

— Pieter Wuille, Stack Exchange (2019)

Ele explica que, para alcançar a ausência de necessidade de permissão, o sistema provavelmente precisa de sua própria moeda, limitando assim os casos de uso a praticamente apenas criptomoedas. Isso ocorre porque a participação sem necessidade de permissão, ou mineração, requer incentivos econômicos embutidos no próprio sistema.

1.4. Compreendendo a descentralização



Um aspecto fascinante do Bitcoin é como é difícil compreender que ninguém o controla. Não há comitês ou executivos no Bitcoin. Gregory Maxwell, novamente [no subreddit do Bitcoin](#), compara isso à língua inglesa de uma maneira intrigante:

Muitas pessoas têm dificuldade em entender sistemas autônomos, há muitos em suas vidas coisas como a língua inglesa-- mas as pessoas simplesmente os tomam como garantidos e nem mesmo os consideram sistemas. Elas estão presas em um modo de pensar centralizado onde tudo o que consideram um 'objeto' tem uma autoridade que o controla.

O Bitcoin não foca em nada. Várias pessoas que adotaram o Bitcoin escolheram, por vontade própria, promovê-lo, e como elas escolhem fazer isso é problema delas. Pessoas fixadas em autoridade podem ver essas atividades e acreditar que são alguma operação pela autoridade do bitcoin, mas tal autoridade não existe.

— Gregory Maxwell, r/Bitcoin subreddit (2022)

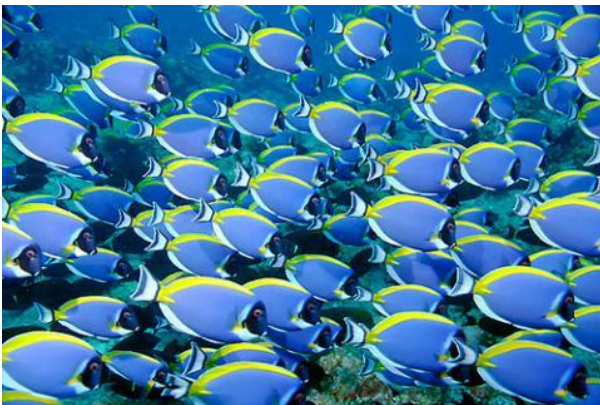


Figure 1. Cardumes de peixes não têm líderes.



A forma como o Bitcoin funciona através da descentralização se assemelha à inteligência coletiva extraordinária encontrada entre muitas espécies na natureza. A cientista da computação Radhika Nagpal fala em um [Ted talk](#) sobre o comportamento coletivo dos cardumes de peixes e como os cientistas estão tentando imitá-lo usando robôs.

Em segundo lugar, e a coisa que ainda acho mais notável, é que sabemos que não há líderes supervisionando esse cardume de peixes. Em vez disso, esse comportamento de mente coletiva incrível está emergindo puramente das interações de um peixe com outro. De alguma forma, existem essas interações ou regras de engajamento entre os peixes vizinhos que fazem tudo funcionar.

— Radhika Nagpal, What intelligent machines can learn from a school of fish (2017)

Ela aponta que muitos sistemas, sejam naturais ou artificiais, podem e funcionam sem líderes, e eles são poderosos e resilientes. Cada indivíduo apenas interage com seu ambiente imediato, mas juntos formam algo tremendo.

Não importa o que você pense sobre o Bitcoin, sua natureza descentralizada torna difícil controlá-lo. O Bitcoin existe, e não há nada que você possa fazer a respeito. É algo a ser estudado, não debatido.

1.5. Conclusão

Distinguimos entre descentralização de nós completos e descentralização de mineradores. A descentralização dos mineradores é um meio para alcançar a resistência à censura, enquanto a descentralização dos nós completos é o que mantém as regras de consenso da rede difíceis de mudar sem amplo apoio entre os usuários.

A natureza descentralizada do Bitcoin permite a neutralidade em relação a desenvolvedores, usuários e mineradores. Qualquer pessoa é livre para participar sem pedir permissão.

Sistemas descentralizados podem ser difíceis de entender, mas há alguns modelos mentais que podem ajudar, como a língua inglesa ou cardumes de peixes.

Chapter 2. Confiança Zero



Este capítulo diseca o conceito de confiança zero (trustlessness), o que ele significa do ponto de vista da ciência da computação e por que o Bitcoin precisa ser trustless para manter sua proposta de valor. Em seguida, falamos sobre o que significa usar o Bitcoin de uma maneira sem confiança e que tipo de garantias um full node pode e não pode oferecer. Na última seção, examinamos a interação do Bitcoin no mundo real com softwares ou usuários reais, e a necessidade de fazer trade-offs entre conveniência e confiança zero para conseguir realizar qualquer coisa.



As pessoas frequentemente dizem coisas como “Bitcoin é ótimo porque não exige confiança”. O que elas querem dizer com confiança zero? Pieter Wuille explica esse termo amplamente usado no [Stack Exchange](#):

A confiança de que estamos falando em "confiança zero" é um termo técnico abstrato. Um sistema distribuído é chamado de sem confiança quando não requer nenhuma parte confiável para funcionar corretamente.

— Pieter Wuille, Bitcoin Stack Exchange (2016)

Em resumo, a palavra *trustless* refere-se a uma propriedade do protocolo Bitcoin, segundo a qual ele pode funcionar logicamente sem “nenhuma parte confiável”. Isso é diferente da confiança que você inevitavelmente tem que depositar no software ou hardware que você utiliza. Mais sobre esse

aspecto da confiança será discutido mais adiante neste capítulo.



Em sistemas centralizados, confiamos na reputação de um ator central para garantir que ele cuidará da segurança ou reverterá em caso de problemas, bem como no sistema legal para sancionar quaisquer violações. Esses requisitos de confiança são problemáticos em sistemas descentralizados pseudônimos - não há possibilidade de recurso, então realmente não pode haver confiança. Na introdução ao [whitepaper do Bitcoin](#), Satoshi Nakamoto descreve esse problema:

O comércio na Internet passou a depender quase exclusivamente de instituições financeiras que servem como terceiros confiáveis para processar pagamentos eletrônicos. Embora o sistema funcione bem o suficiente para a maioria das transações, ele ainda sofre com as fraquezas inerentes ao modelo baseado em confiança. Transações completamente não reversíveis não são realmente possíveis, pois as instituições financeiras não podem evitar a mediação de disputas. O custo da mediação aumenta os custos das transações, limitando o tamanho mínimo prático da transação e eliminando a possibilidade de pequenas transações casuais, e há um custo mais amplo na perda da capacidade de fazer pagamentos não reversíveis para serviços não reversíveis. Com a possibilidade de reversão, a necessidade de confiança se espalha. Os comerciantes devem ficar atentos aos seus clientes, incomodando-os por mais informações do que normalmente precisariam. Uma certa porcentagem de fraude é aceita como inevitável. Esses custos e incertezas de pagamento podem ser evitados pessoalmente usando moeda física, mas não existe nenhum mecanismo para fazer pagamentos através de um canal de comunicação sem uma parte confiável.

— Satoshi Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System (2008)

Parece que não podemos ter um sistema descentralizado baseado em confiança, e é por isso que a confiança zero é importante no Bitcoin.

Para usar o Bitcoin de forma sem confiança, você precisa executar um node Bitcoin de validação completa. Só então você será capaz de verificar se os blocos que você recebe de outros estão seguindo as regras de consenso; por exemplo, se o cronograma de emissão de moedas está sendo mantido e se não ocorrem double-spends na blockchain. Se você não executar um full node, você terceiriza a verificação dos blocos do Bitcoin para outra pessoa e confia que eles estão lhe dizendo a verdade, o que significa que você não está usando o Bitcoin de forma sem confiança.



David Harding é autor de [um artigo no site bitcoin.org](#) explicando como executar um full node - ou usar o Bitcoin sem confiança - realmente ajuda você.

A moeda bitcoin só funciona quando as pessoas aceitam bitcoins em troca de outras coisas valiosas. Isso significa que são as pessoas que aceitam bitcoins que dão valor a ele e que decidem como o Bitcoin deve funcionar.

Quando você aceita bitcoins, você tem o poder de impor as regras do Bitcoin, como evitar a confiscação de bitcoins de qualquer pessoa sem acesso às chaves privadas dessa pessoa.

Infelizmente, **muitos usuários terceirizam seu poder de imposição**. Isso deixa a descentralização do Bitcoin em um estado enfraquecido, onde um punhado de mineradores pode conspirar com um punhado de bancos e serviços gratuitos para mudar as regras do Bitcoin para todos aqueles usuários não verificadores que terceirizaram seu poder.

Ao contrário de outras carteiras, **o Bitcoin Core impõe as regras** - então, se os mineradores e bancos mudarem as regras para seus usuários não verificadores, esses usuários não conseguirão pagar usuários do Bitcoin Core de validação completa como você.

— David Harding, Validação Completa no [bitcoin.org](#) (2015)

Ele diz que executar um full node ajudará você a verificar todos os aspectos da blockchain sem confiar em ninguém, de modo a garantir que as moedas que você recebe de outros são genuínas. Isso é ótimo, mas há uma coisa importante que um full node não pode ajudar: ele não pode impedir o double-spending através de reescritas da cadeia:

Observe que, embora todos os programas - incluindo o Bitcoin Core - sejam vulneráveis a reescritas da cadeia, o Bitcoin oferece um mecanismo de defesa: quanto mais confirmações suas transações tiverem, mais seguro você estará. Não há defesa descentralizada conhecida melhor do que essa.

— David Harding, Validação Completa no [bitcoin.org](#) (2015)

Não importa quão avançado seja o seu software, você ainda precisa confiar que os blocos contendo suas moedas não serão reescritos. No entanto, como apontado por Harding, você pode esperar um número de confirmações, após as quais você considera que a probabilidade de uma reescrita da cadeia é pequena o suficiente para ser aceitável.

Os incentivos para usar o Bitcoin de forma sem confiança estão alinhados com a necessidade do sistema de [descentralização dos full nodes](#). Quanto mais pessoas usam seus próprios full nodes, maior é a descentralização dos full nodes, e assim o Bitcoin se torna mais resistente a mudanças maliciosas no protocolo. Mas, infelizmente, como explicado na seção de descentralização dos full

nodes, os usuários muitas vezes optam por serviços confiáveis como consequência do trade-off inevitável entre confiança zero e conveniência.



A confiança zero no Bitcoin é absolutamente imperativa do ponto de vista do sistema. Em 2018, Matt Corallo, [falou sobre confiança zero](https://youtu.be/66ZoGUAnY9s?t=4019) na conferência Baltic Honeybadger em Riga. // Vídeo: <https://youtu.be/66ZoGUAnY9s?t=4019> O cerne dessa palestra é que você não pode construir sistemas sem confiança em cima de um sistema confiável, mas pode construir sistemas confiáveis - por exemplo, uma carteira custodial - em cima de um sistema sem confiança.

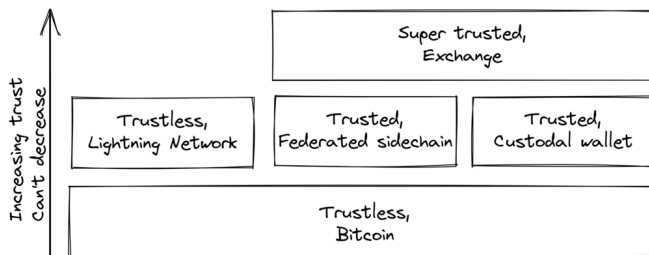


Figure 2. Uma camada base sem confiança permite vários trade-offs em níveis superiores.

Esse modelo de segurança permite que o designer do sistema selecione trade-offs que fazem sentido para eles, sem forçar esses trade-offs a outros.

2.1. Não confie, verifique

O Bitcoin funciona sem confiança, mas você ainda precisa confiar no seu software e hardware em algum grau. Isso porque o seu software ou hardware pode não estar programado para fazer o que está especificado. Por exemplo:

- A CPU pode ser projetada maliciosamente para detectar operações criptográficas com chave privada e vazá-las.
- O gerador de números aleatórios do sistema operacional pode não ser tão aleatório quanto afirma.
- O Bitcoin Core pode ter introduzido código que enviará suas chaves privadas para algum ator mal-intencionado.



Então, além de executar um full node, você também precisa garantir que está executando o que pretende. O usuário do Reddit brianddk [escreveu um artigo](#) sobre os vários níveis de confiança que você pode escolher, ao verificar seu software. Na seção “Confiando nos construtores”, ele fala sobre *builds reprodutíveis*:

Builds reproduzíveis são uma maneira de projetar software para que muitos desenvolvedores da comunidade possam construir o software e garantir que o instalador final construído seja idêntico ao que outros desenvolvedores produzem. Com um projeto muito público e reproduzível como o bitcoin, nenhum desenvolvedor individual precisa ser completamente confiável. Muitos desenvolvedores podem todos realizar a construção e atestar que produziram o mesmo arquivo que o arquivo que o construtor original assinou digitalmente.

— brianddk no Reddit, Bitcoin v22.0 e Guix; Defesa mais forte contra o "Ataque de Confiança Confiável" (2022)

O artigo define 5 níveis de confiança: confiar no site, nos construtores, no compilador, no kernel e no hardware.



Para aprofundar ainda mais o tópico de builds reproduzíveis, Carl Dong [fez uma apresentação sobre Guix](#) explicando por que confiar no sistema operacional, bibliotecas e compiladores pode ser problemático e como corrigir isso com um sistema chamado Guix, que é usado pelo Bitcoin Core hoje.

Então, o que podemos fazer sobre o fato de que nossa cadeia de ferramentas pode ter um monte de binários confiáveis que podem ser maliciosamente reproduzíveis? Precisamos ser mais do que reproduzíveis. Precisamos ser inicializáveis. Não podemos ter tantas ferramentas binárias que precisamos baixar e confiar de servidores externos controlados por outras organizações. Devemos saber como essas ferramentas são construídas e exatamente como podemos passar pelo processo de construí-las novamente, de preferência a partir de um conjunto muito menor de binários confiáveis. Precisamos minimizar nosso conjunto confiável de binários o máximo possível e ter um caminho facilmente auditável dessas cadeias de ferramentas até o que usamos para construir o bitcoin. Isso nos permite maximizar a verificação e minimizar a confiança.

— Carl Dong sobre Guix, Conferência Breaking Bitcoin (2019)

Ele então explica como o Guix nos permite confiar apenas em um binário mínimo de 357 bytes que pode ser verificado e completamente entendido se você souber como interpretar as instruções. Isso é bastante notável: você verifica que o binário de 357 bytes faz o que deveria, depois o usa para construir todo o sistema de build a partir do código-fonte e acaba com um binário do Bitcoin Core que deve ser uma cópia exata do build de qualquer outra pessoa.

Há um mantra ao qual muitos bitcoiners aderem, que captura bem grande parte do que foi dito

acima:

Não confie, verifique.

— Bitcoiners em todo lugar

Isso alude à frase "[confie, mas verifique](#)" que o ex-presidente dos EUA Ronald Reagan usou no contexto do desarmamento nuclear. [Bitcoiners inverteram isso para destacar a rejeição da confiança e a importância de executar um full node.](#)

Cabe aos usuários decidir em que grau desejam verificar o software que usam e os dados da blockchain que recebem. Como em tantas outras coisas no Bitcoin, há um trade-off entre conveniência e confiança zero. Quase sempre é mais conveniente usar uma carteira custodial em comparação a executar o Bitcoin Core no seu próprio hardware. No entanto, à medida que o software do Bitcoin amadurece e as interfaces do usuário melhoram, com o tempo deve se tornar melhor em apoiar usuários dispostos a trabalhar em direção à confiança zero. Além disso, à medida que os usuários ganham mais conhecimento ao longo do tempo, eles devem ser capazes de remover gradualmente a confiança da equação.



Alguns usuários pensam de forma adversarial (veja [\[adversarialthinking\]](#)) e verificam a maioria dos aspectos do software que executam. Como consequência, eles reduzem a necessidade de confiança ao mínimo, já que precisam confiar apenas no hardware e no sistema operacional de seu computador. Ao fazer isso, eles também ajudam pessoas que não verificam seu hardware tão minuciosamente, levantando suas vozes em público para alertar sobre qualquer problema que possam encontrar. Um bom exemplo disso é um [evento que ocorreu em 2018](#), quando alguém descobriu um bug que permitiria aos mineradores gastar uma saída duas vezes na mesma transação:

CVE-2018-17144, uma correção para a qual foi lançada em 18 de setembro nas versões 0.16.3 e 0.17.0rc4 do Bitcoin Core, inclui tanto um componente de Denial of Service quanto uma vulnerabilidade crítica de inflação. Foi originalmente relatado a vários desenvolvedores trabalhando no Bitcoin Core, bem como em projetos que suportam outras criptomoedas, incluindo ABC e Unlimited em 17 de setembro apenas como um bug de Denial of Service, no entanto, determinamos rapidamente que o problema também era uma vulnerabilidade de inflação com a mesma causa raiz e correção.

— Divulgação Completa do CVE-2018-17144 no [bitcoincore.org](#) (2018)

Aqui, uma pessoa anônima relatou um problema que acabou sendo muito pior do que o relator imaginava. Isso destaca o fato de que as pessoas que verificam o código frequentemente relatam falhas de segurança em vez de explorá-las. Isso é benéfico para aqueles que não conseguem verificar tudo por si mesmos. No entanto, os usuários não devem confiar nos outros para mantê-los seguros, mas devem verificar por si mesmos sempre que e o que puderem; é assim que se

permanece o mais soberano possível e como o Bitcoin prospera. Quanto mais olhos no software, menos provável é que código malicioso e falhas de segurança passem despercebidos.

2.2. Conclusão

O protocolo Bitcoin é sem necessidade de confiança porque permite que os usuários interajam com ele sem confiar em uma terceira parte. Na prática, entretanto, a maioria das pessoas não é capaz de verificar toda a pilha de software e hardware em que executa o Bitcoin. Pessoas habilidosas que verificam software ou hardware são capazes de alertar outras, menos habilidosas, quando encontram código malicioso ou bugs.

Sem confiança zero, não podemos ter descentralização, porque a confiança inevitavelmente envolve algum ponto central de autoridade. Você pode construir um sistema confiável em cima de um sistema sem confiança, mas não pode construir um sistema sem confiança em cima de um sistema confiável.

Chapter 3. Privacidade



Este capítulo trata de como manter suas informações financeiras privadas. Ele explica o que significa privacidade no contexto do Bitcoin, por que é importante e o que significa dizer que o Bitcoin é pseudônimo. Também examina como os dados privados podem vazar, tanto na blockchain quanto fora dela. Em seguida, aborda o fato de que os bitcoins devem ser fungíveis, ou seja, intercambiáveis com outros bitcoins, e como a fungibilidade e a privacidade andam de mãos dadas. Por fim, o capítulo apresenta algumas medidas que você pode tomar para melhorar sua privacidade e a de outros.

O Bitcoin pode ser descrito como um sistema pseudônimo (veja [Section 3.3](#) para mais detalhes sobre isso), onde os usuários têm vários pseudônimos na forma de chaves públicas. À primeira vista, isso parece uma boa maneira de proteger os usuários de serem identificados, mas na verdade é muito fácil vazar informações financeiras privadas de forma não intencional.

3.1. O que significa privacidade?

Privacidade pode significar coisas diferentes em diferentes contextos. No Bitcoin, geralmente significa que os usuários não precisam revelar suas informações financeiras para os outros, a menos que o façam voluntariamente.

Existem muitas maneiras pelas quais você pode vazar suas informações privadas para outros, com ou sem saber. Os dados podem vazar da blockchain pública ou por outros meios, por exemplo, quando atores mal-intencionados interceptam suas comunicações na internet.

3.2. Por que a privacidade é importante?



Pode parecer óbvio por que a privacidade é importante no Bitcoin, mas há alguns aspectos disso que talvez não sejam imediatamente percebidos. [No fórum Bitcoin Talk](#), Gregory Maxwell nos guia por várias boas razões pelas quais ele acredita que a privacidade é importante. Entre elas estão o mercado livre, a segurança e a dignidade humana:

A privacidade financeira é um critério essencial para o funcionamento eficiente de um mercado livre: se você administra um negócio, não pode definir preços de forma eficaz se seus fornecedores e clientes podem ver todas as suas transações contra sua vontade. Você não pode competir de forma eficaz se seus concorrentes estiverem acompanhando suas vendas. Individualmente, sua alavancagem informacional é perdida em seus negócios privados se você não tiver privacidade sobre suas contas: se você pagar seu senhorio em Bitcoin sem a privacidade adequada, seu senhorio verá quando você recebeu um aumento de salário e poderá pedir mais aluguel.

A privacidade financeira é essencial para a segurança pessoal: se ladrões podem ver seus gastos, renda e posses, eles podem usar essas informações para direcioná-lo e explorá-lo. Sem privacidade, partes mal-intencionadas têm mais capacidade de roubar sua identidade, roubar suas grandes compras na sua porta ou se passar por empresas com as quais você transaciona... eles podem saber exatamente quanto tentar te enganar.

A privacidade financeira é essencial para a dignidade humana: ninguém quer que o barista intrometido na cafeteria ou seus vizinhos curiosos comentem sobre sua renda ou hábitos de consumo. Ninguém quer que seus sogros obcecados por bebês perguntem por que estão comprando contraceptivos (ou brinquedos sexuais). Seu empregador não tem nada a ver com saber para qual igreja você doa. Apenas em um mundo perfeitamente iluminado, livre de discriminação, onde ninguém tem autoridade indevida sobre ninguém, poderíamos manter nossa dignidade e fazer nossas transações legais livremente, sem auto-censura, se não tivérmos privacidade.

— Gregory Maxwell, Bitcoin Talk forum (2013)

Maxwell também toca na fungibilidade, que será discutida [mais adiante neste capítulo](#), bem como

na ideia de que privacidade e aplicação da lei não são contraditórias.

3.3. Pseudonimidade

Mencionamos acima que o Bitcoin é pseudônimo, e que os pseudônimos são chaves públicas. Na mídia, muitas vezes se ouve que o Bitcoin é anônimo, o que não é correto. Há uma distinção entre anonimato e pseudonimidade.



Andrew Poelstra [explica em um post no Bitcoin Stack Exchange](#) como seria o anonimato em transações:

O anonimato total, no sentido de que, quando você gasta dinheiro, não há vestígios de onde ele veio ou para onde está indo, é teoricamente possível usando a técnica criptográfica de provas de conhecimento zero.

— Andrew Poelstra sobre anonimato, Bitcoin Stack Exchange (2016)

A diferença parece ser que, em uma forma de dinheiro pseudônimo, você pode rastrear pagamentos entre pseudônimos, enquanto em uma forma anônima de dinheiro, você não pode. Como os pagamentos em bitcoin são rastreáveis entre pseudônimos, não é um sistema anônimo.



Também dissemos que os pseudônimos são chaves públicas, mas, na verdade, são endereços derivados de chaves públicas. Por que usamos endereços como pseudônimos e não outra coisa, por exemplo, alguns nomes descritivos, como “watchme1984”? Isso foi [bem explicado](#) pelo usuário Tim S., também no Bitcoin Stack Exchange:

Para que a ideia do Bitcoin funcione, você deve ter moedas que só possam ser gastas pelo proprietário de uma determinada chave privada. Isso significa que o que quer que você envie deve estar vinculado, de alguma forma, a uma chave pública.

Usar pseudônimos arbitrários (por exemplo, nomes de usuário) significaria que você teria que, de alguma forma, vincular o pseudônimo a uma chave pública para permitir a criptografia de chave pública/privada. Isso removeria a capacidade de criar endereços/pseudônimos com segurança offline (por exemplo, antes que alguém pudesse enviar dinheiro para o nome de usuário "tdumidu", você teria que anunciar na blockchain que "tdumidu" é de propriedade da chave pública "a1c...", e incluir uma taxa para que outros tenham um motivo para anunciá-lo), reduziria o anonimato (incentivando você a reutilizar pseudônimos) e inchava desnecessariamente o tamanho da blockchain. Também criaria uma falsa sensação de segurança de que você está enviando para quem acha que está (se eu pegar o nome "Linus Torvalds" antes dele, então ele é meu, e as pessoas podem enviar dinheiro pensando que estão pagando o criador do Linux, não eu).

— Tim S. sobre pseudônimos, Bitcoin Stack Exchange (2014)

Ao usar endereços ou chaves públicas, alcançamos objetivos importantes, como eliminar a necessidade de registrar um pseudônimo com antecedência, reduzir os incentivos para reutilização de pseudônimos, evitar o inchaço da blockchain e dificultar a personificação de outras pessoas.

3.4. Privacidade na blockchain

A privacidade na blockchain refere-se às informações que você divulga ao transacionar na blockchain. Isso se aplica a todas as transações, tanto as que você envia quanto as que você recebe.



Satoshi Nakamoto reflete sobre a privacidade na blockchain na seção 7 de seu [whitepaper do Bitcoin](#):

Como uma barreira adicional, um novo par de chaves deve ser usado para cada transação para evitar que sejam vinculadas a um proprietário comum. Alguns vínculos ainda são inevitáveis em transações com várias entradas, que necessariamente revelam que suas entradas eram de propriedade do mesmo proprietário. O risco é que, se o proprietário de uma chave for revelado, o vínculo pode revelar outras transações que pertenciam ao mesmo proprietário.

— Satoshi Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System (2008)

O documento resume os principais problemas de privacidade na blockchain, ou seja, a reutilização de endereços e o agrupamento de endereços. O primeiro é autoexplicativo, o segundo refere-se à capacidade de determinar, com algum nível de certeza, que um conjunto de endereços diferentes pertence ao mesmo usuário.

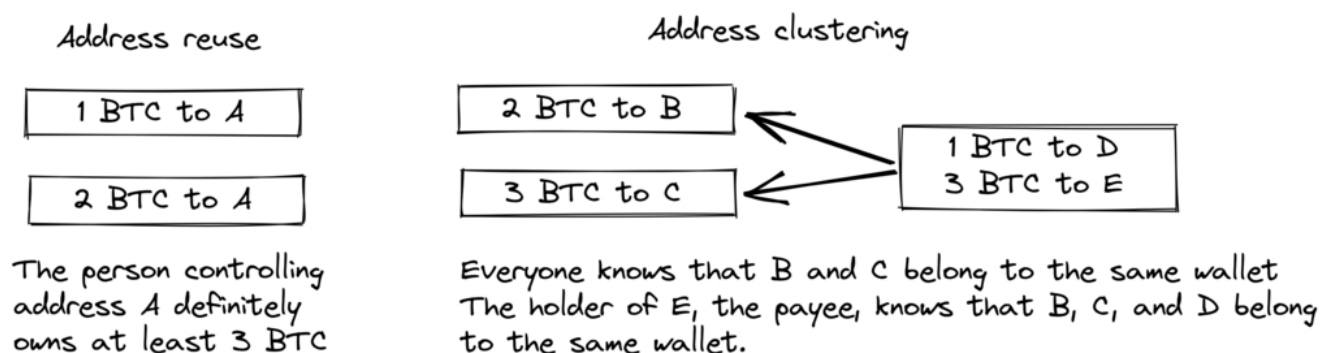


Figure 3. Típicos vazamentos de privacidade na blockchain.



Chris Belcher [escreveu em detalhes](#) sobre os diferentes tipos de vazamentos de privacidade que podem ocorrer na blockchain do Bitcoin. Recomendamos que você leia pelo menos as primeiras subseções em “Ataques à privacidade na blockchain.”

A conclusão é que a privacidade no Bitcoin não é perfeita. É necessário um trabalho significativo para transacionar de forma privada. A maioria das pessoas não está disposta a ir tão longe pela privacidade. Parece haver um claro trade-off entre privacidade e usabilidade.



Outro aspecto importante da privacidade é que as medidas que você toma para proteger sua própria privacidade afetam outros usuários também. Se você for negligente com sua própria privacidade, outras pessoas podem experimentar uma privacidade reduzida também. Gregory Maxwell explica isso de forma muito clara na mesma discussão do Bitcoin Talk [que vinculamos acima](#) e conclui com um exemplo:

Isso realmente funciona na prática, também... Um hacker whitehat amigável no IRC estava brincando com cracking de brainwallet e encontrou uma frase com ~250 BTC nela. Conseguimos identificar o proprietário apenas pelo endereço, porque eles haviam sido pagos por um serviço de Bitcoin que reutilizou endereços, e ele conseguiu convencê-los a fornecer as informações de contato do usuário. Ele realmente conseguiu falar com o usuário ao telefone, eles ficaram chocados e confusos—mas gratos por não perderem suas moedas. Um final feliz lá. (Este não é o único exemplo disso, de longe ... mas é um dos mais divertidos).

— Gregory Maxwell, Bitcoin Talk forum (2013)

Nesse caso, tudo acabou bem graças ao hacker de bom coração, mas não conte com isso na próxima vez.

3.5. Privacidade fora da blockchain



Embora a blockchain seja uma notória fonte de vazamentos de privacidade, há muitos outros vazamentos que não usam a blockchain, alguns mais sorrateiros que outros. Esses variam de key-loggers a análise de tráfego de rede. Para ler sobre alguns desses métodos, consulte novamente o [artigo de Chris Belcher](#), especificamente a seção “Ataques fora da blockchain à privacidade.”

Entre uma infinidade de ataques, Belcher menciona a possibilidade de alguém espionar sua conexão de internet, por exemplo, seu ISP:

Se o adversário vir uma transação ou bloco saindo do seu nó que não entrou anteriormente, ele pode saber com quase certeza que a transação foi feita por você ou o bloco foi minerado por você. Como conexões de internet estão envolvidas, o adversário poderá vincular o endereço IP com as informações de bitcoin descobertas.

— Chris Belcher, Bitcoin wiki

No entanto, entre os vazamentos de privacidade mais óbvios estão as exchanges. Devido a leis, geralmente referidas como KYC (Know Your Customer) e AML (Anti-Money Laundering), que são válidas nas jurisdições em que operam, exchanges e empresas relacionadas geralmente precisam coletar dados pessoais sobre seus usuários, criando grandes bancos de dados sobre quais usuários possuem quais bitcoins. Esses bancos de dados são grandes alvos para governos malignos e criminosos que estão sempre à procura de novas vítimas. Existem mercados reais para esse tipo de dados, onde hackers vendem dados para o maior lance. Para piorar as coisas, as empresas que gerenciam esses bancos de dados geralmente têm pouca experiência em proteger dados financeiros, na verdade, muitas delas são start-ups jovens, e sabemos de fato que já ocorreram

vários vazamentos. Alguns exemplos são [MobiQwik, com sede na Índia](#) e [HubSpot](#).

Novamente, proteger dados contra essa ampla gama de ataques é difícil, e é provável que você não consiga fazer isso totalmente. Você terá que optar pelo trade-off entre conveniência e privacidade que funciona melhor para você.

3.6. Fungibilidade



Fungibilidade, no contexto de moedas, significa que uma moeda é intercambiável por qualquer outra moeda da mesma denominação. Essa palavra curiosa foi brevemente mencionada em [Section 3.2](#). No artigo discutido lá, Gregory Maxwell [afirmou](#):

A privacidade financeira é um elemento essencial para a fungibilidade no Bitcoin: se você pode distinguir significativamente uma moeda de outra, então sua fungibilidade é fraca. Se nossa fungibilidade for muito fraca na prática, então não podemos ser descentralizados: se alguém importante anunciar uma lista de moedas roubadas que eles não aceitarão moedas derivadas, você deve verificar cuidadosamente as moedas que aceita contra essa lista e devolver as que falharem. Todos acabam verificando listas negras emitidas por várias autoridades porque, nesse mundo, ninguém gostaria de ficar com moedas ruins. Isso adiciona fricção e custos transacionais e torna o Bitcoin menos valioso como dinheiro.

— Gregory Maxwell, Bitcoin Talk forum (2013)

Aqui, ele fala sobre os perigos decorrentes da falta de fungibilidade. Suponha que você tenha um UTXO. O histórico desse UTXO normalmente pode ser rastreado por vários saltos, se espalhando para múltiplos outputs anteriores. Se algum desses outputs esteve envolvido em qualquer atividade ilegal, indesejada ou suspeita, alguns potenciais destinatários de sua moeda podem rejeitá-la. Se você achar que seus pagadores verificarão suas moedas contra algum serviço centralizado de listas brancas ou negras, você pode começar a verificar as moedas que recebe também, apenas para garantir. O resultado é que uma fungibilidade ruim vai fomentar uma fungibilidade ainda pior.



Adam Back e Matt Corallo [fizeram uma apresentação sobre fungibilidade](#) na Scaling Bitcoin em Milão, em 2016. Eles estavam pensando da mesma forma:

Você precisa de fungibilidade para o bitcoin funcionar. Se você recebe moedas e não pode gastá-las, então começa a duvidar se poderá gastá-las. Se houver dúvidas sobre as moedas que você recebe, então as pessoas vão procurar serviços de checagem de contaminação e verificar se "essas moedas são abençoadas" e, em seguida, as pessoas vão se recusar a negociar. O que isso faz é transformar o bitcoin de um sistema descentralizado sem permissão em um sistema centralizado com permissão, onde você tem um "IOU" dos provedores de listas negras.

— Matt Corallo and Adam Back, Fungibility Overview (2016)

Parece que a privacidade e a fungibilidade andam de mãos dadas. A fungibilidade enfraquecerá se a privacidade for fraca, por exemplo, à medida que as moedas de pessoas indesejadas podem se tornar incluídas em listas negras. Da mesma forma, a privacidade enfraquecerá se a fungibilidade for fraca: se houver uma lista negra, você terá que perguntar aos provedores da lista negra sobre quais moedas aceitar, possivelmente revelando assim seu endereço IP, endereço de e-mail e outras informações confidenciais. Essas duas características são tão entrelaçadas que é difícil falar de uma delas isoladamente.

3.7. Medidas de privacidade



Várias técnicas foram desenvolvidas para ajudar as pessoas a se protegerem contra vazamentos de privacidade. Entre as mais óbvias, como mencionado por Nakamoto em [Section 3.4](#), está o uso de endereços exclusivos para cada transação, mas várias outras existem. Não vamos te ensinar como se tornar um ninja da privacidade. No entanto, Bitcoin Q+A tem um [resumo rápido das tecnologias que melhoram a privacidade](#), ordenado de acordo com a dificuldade de implementação. Quando você lê-lo, perceberá que a privacidade no Bitcoin muitas vezes tem a ver com coisas fora do Bitcoin. Por exemplo, você não deve se gabar de seus bitcoins e deve usar Tor e VPN. O post também lista algumas medidas diretamente relacionadas ao Bitcoin:

Full node

Se você não usa seu próprio full node, você vazará muitas informações sobre sua carteira para servidores na internet. Executar um full node é um ótimo primeiro passo.

Lightning Network

Vários protocolos existem sobre o Bitcoin, por exemplo, a Lightning Network e a sidechain Liquid da Blockstream.

CoinJoin

Uma maneira para várias pessoas fundirem suas transações em uma só, dificultando a análise de cadeia.



Em [uma palestra](#) na conferência Breaking Bitcoin, Chris Belcher deu um exemplo prático interessante de como a privacidade foi melhorada.

Eles eram um cassino de bitcoin. Jogos de azar online não são permitidos nos EUA. Qualquer cliente da Coinbase que depositasse diretamente na Bustabit teria sua conta encerrada porque a Coinbase estava monitorando isso. A Bustabit fez algumas coisas. Eles fizeram algo chamado de evitação de troco, onde você verifica se pode construir uma transação que não tenha saída de troco. Isso economiza taxas de minerador e também dificulta a análise. Além disso, eles importaram seus endereços de depósito muito usados e reutilizados no joinmarket. A partir desse ponto, os clientes da coinbase.com nunca foram banidos. Parece que o serviço de vigilância da Coinbase não conseguiu fazer a análise depois disso, então é possível quebrar esses algoritmos.

— Chris Belcher em "Breaking Bitcoin Privacy", Breaking Bitcoin conference (2019)



Ele também mencionou este exemplo, entre outros, na [página de privacidade](#) no wiki do Bitcoin.

Observe como a privacidade pode ser melhorada construindo sistemas em cima do Bitcoin, como é o caso da Lightning Network:

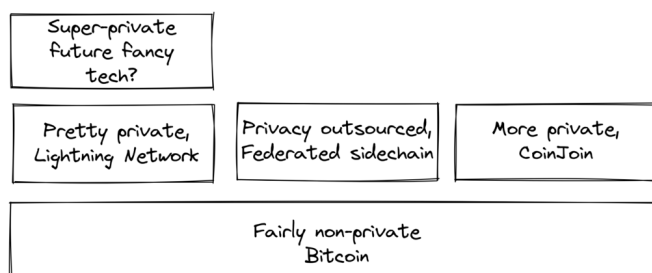


Figure 4. Camadas em cima do Bitcoin podem adicionar privacidade.

Notamos em [Chapter 2](#) que a necessidade de confiança só pode aumentar com camadas em cima, mas isso não parece ser o caso da privacidade, que pode ser melhorada ou piorada arbitrariamente em camadas adicionais. Por que isso? Qualquer camada em cima do Bitcoin, como explicado em [Section 7.2.4](#), deve usar transações na cadeia ocasionalmente, caso contrário, não seria “em cima do Bitcoin”. As camadas que melhoram a privacidade geralmente tentam usar a camada base o mínimo possível para minimizar a quantidade de informações reveladas.

As acima são formas um tanto técnicas de melhorar sua privacidade. Mas há outras maneiras. No início deste capítulo, dissemos que o Bitcoin é um sistema pseudônimo. Isso significa que os usuários no Bitcoin não são conhecidos por seus nomes reais ou outros dados pessoais, mas por

suas chaves públicas. Uma chave pública é um pseudônimo para um usuário, e um usuário pode ter vários pseudônimos. Em um mundo ideal, sua identidade em pessoa é desacoplada de seus pseudônimos de Bitcoin. Infelizmente, devido aos problemas de privacidade descritos neste capítulo, esse desacoplamento geralmente se degrada com o tempo.



Para mitigar os riscos de ter seus dados pessoais revelados, não forneça-os em primeiro lugar nem os entregue a serviços centralizados, que constroem grandes bancos de dados que podem vazar (veja [Section 3.5](#)). Um artigo de Bitcoin Q+A [explica KYC](#) e os perigos decorrentes disso. Também sugere alguns passos que você pode tomar para melhorar sua situação.

Felizmente, existem algumas opções para comprar Bitcoin por meio de fontes sem KYC. Estas são todas exchanges P2P (peer-to-peer) onde você negocia diretamente com outro indivíduo e não com uma terceira parte centralizada. Infelizmente, alguns vendem outras moedas além de bitcoin, então recomendamos que você tome cuidado.

— Bitcoin Q+A, noKYC only, Avoid the creep, [bitcoiner.guide](#)

O artigo sugere que você evite usar exchanges que exigem KYC/AML e, em vez disso, negocie em privado, ou use exchanges descentralizadas como [bisq](#).



Para leitura mais aprofundada sobre contramedidas, consulte o artigo mencionado anteriormente [artigo wiki sobre privacidade](#), começando em “Métodos para melhorar a privacidade (fora da blockchain)”.

3.8. Conclusão

A privacidade é muito importante, mas difícil de alcançar. Não há uma bala de prata para a privacidade. Para obter uma privacidade decente no Bitcoin, você precisa tomar medidas ativas, algumas das quais são caras e demoradas.

Chapter 4. Emissão finita



Este capítulo explora o limite de fornecimento de bitcoin de 21 milhões de BTC, ou quanto realmente é? Falamos sobre como esse limite é imposto e o que se pode fazer para verificar se está sendo respeitado. Além disso, damos uma espiada na bola de cristal e discutimos as dinâmicas que entrarão em jogo quando a recompensa de bloco passar de baseada em subsídio para baseada em taxas.

O bem conhecido limite finito de 21 milhões de BTC é considerado uma propriedade fundamental do Bitcoin. Mas será que realmente está gravado em pedra?



Vamos começar olhando o que as regras de consenso atuais dizem sobre o fornecimento de bitcoin e quanto dele será realmente utilizável. Pieter Wuille escreveu um artigo sobre isso [no Stack Exchange](#), no qual ele calculou quantos bitcoins haveriam uma vez que todas as moedas fossem mineradas:

Se você somar todos esses números, você obtém **20999999.9769 BTC**.

— Pieter Wuille, Stack Exchange (2015)

Mas devido a uma série de razões - como problemas iniciais com transações de coinbase, mineradores que acidentalmente reivindicam menos do que o permitido, e perda de chaves privadas - esse limite superior nunca será alcançado. Wuille conclui:

Isso nos deixa com **20999817.31308491** BTC (levando tudo em conta até o bloco 528333)

... No entanto, várias carteiras foram perdidas ou roubadas, transações foram enviadas para o endereço errado, as pessoas esqueceram que possuíam bitcoin. O total dessas perdas pode muito bem ser de milhões. As pessoas tentaram tally known losses up [aqui](#).

Isso nos deixa com: ??? BTC.

— Pieter Wuille, Stack Exchange (2015)



Podemos, portanto, ter certeza de que o fornecimento de bitcoin será de no máximo 20999817.31308491 BTC. Qualquer moeda perdida ou queimada de forma não verificável reduzirá esse número, mas não sabemos em quanto. O interessante é que isso realmente não importa, ou melhor ainda, importa de forma positiva para os detentores de bitcoin, [como explicado](#) por Satoshi Nakamoto:

Moedas perdidas só tornam as moedas de todos os outros um pouco mais valiosas. Pense nisso como uma doação para todos.

— Satoshi Nakamoto on lost bitcoins, Bitcointalk forum (2010)

O fornecimento finito vai diminuir e isso deve, pelo menos em teoria, causar deflação de preços.



Mais importante do que o número exato de moedas em circulação é a maneira como o limite de fornecimento é imposto sem qualquer autoridade central. Alias chytrik coloca isso bem em [Stack Exchange](#).

Então, a resposta é que você não precisa confiar em alguém para não aumentar o fornecimento. Você só precisa rodar algum código que verificará que isso não aconteceu.

— chytrik, Stack Exchange (2021)

Mesmo se alguns nós completos virarem para o lado negro e decidirem aceitar blocos com transações de coinbase de valor maior, todos os outros nós completos simplesmente os negligenciarão e continuarão operando normalmente. Alguns nós completos podem, intencionalmente ou não (veja [Section 8.2.2](#)), rodar softwares mal-intencionados, mas o coletivo protegerá a blockchain de forma robusta. Em conclusão, você pode escolher confiar no sistema sem precisar confiar em ninguém.

4.1. Subsídio de bloco e taxas de transação

Uma recompensa de bloco é composta pelo subsídio de bloco mais as taxas de transação. A recompensa de bloco precisa cobrir os custos de segurança do Bitcoin. Podemos dizer com certeza que, nas condições atuais, em relação ao subsídio de bloco, taxas de transação, preço do bitcoin, tamanho do mempool, poder de hash, grau de descentralização etc., os incentivos para que cada participante siga as regras são altos o suficiente para preservar um sistema monetário seguro.



O que acontece quando o subsídio de bloco se aproxima de zero? Para simplificar, vamos assumir que ele realmente é igual a zero. Nesse ponto, o custo de segurança do sistema é coberto apenas pelas taxas de transação. O que o futuro nos reserva quando isso acontecer, não podemos saber. Os fatores de incerteza são numerosos e ficamos com especulações. Por exemplo, a contribuição de Paul Sztorc sobre o assunto [em seu blog Truthcoin](#) é principalmente especulativa, mas ele tem pelo menos um ponto sólido (observe que M2, conforme referido por Sztorc, é uma medida de oferta monetária fiduciária):

Embora os dois estejam misturados no mesmo "orçamento de segurança", o subsídio de bloco e as taxas de transação são completamente diferentes. Eles são tão diferentes um do outro quanto "os lucros totais da VISA em 2017" são diferentes do "aumento total de M2 em 2017".

— Paul Sztorc, Security Budget in the Long Run, Truthcoin blog (2019)

Hoje, são os detentores que pagam pela segurança (via inflação monetária). Amanhã, será a vez dos gastadores de alguma forma arcar com esse fardo, como ilustrado abaixo.

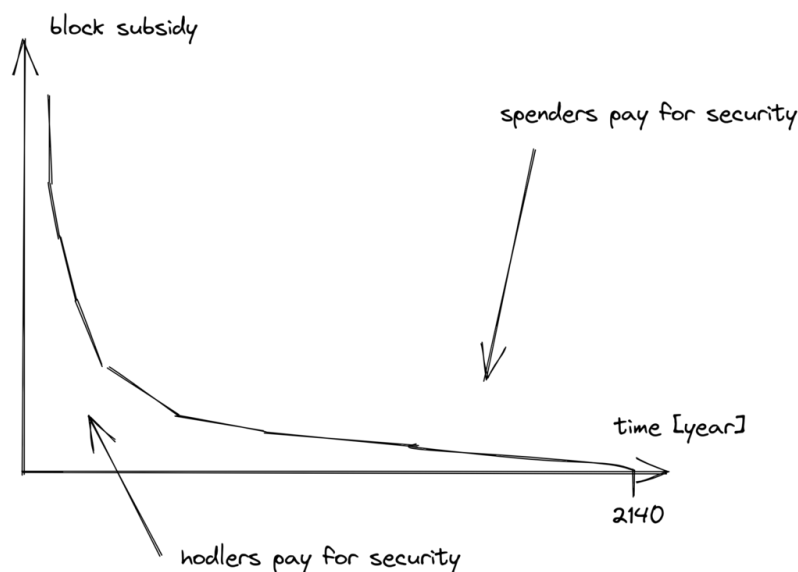


Figure 5. Com o passar do tempo, o ônus dos custos de segurança passará dos detentores para os gastadores.



Quando as taxas de transação forem a principal motivação para a mineração, os incentivos mudarão. Mais notavelmente, se o mempool de um minerador não contiver taxas de transação suficientes, pode se tornar mais lucrativo para esse minerador reescrever a história do Bitcoin em vez de estendê-la. O Bitcoin Optech tem uma [seção específica sobre esse comportamento](#), chamada *fee sniping*, escrita por David Harding:

Fee sniping é um problema que pode ocorrer à medida que o subsídio do Bitcoin continua a diminuir e as taxas de transação começam a dominar as recompensas do bloco do Bitcoin. Se as taxas de transação forem tudo o que importa, então um minerador com x por cento do poder de hash tem uma chance de x por cento de minerar o próximo bloco, então o valor esperado de minerar honestamente para ele é x por cento do [melhor conjunto de taxas de transação](#) em seu mempool.

Alternativamente, um minerador poderia tentar desonestamente re-minerar o bloco anterior, além de um bloco totalmente novo para estender a cadeia. Esse comportamento é conhecido como fee sniping, e a chance do minerador desonesto ter sucesso, se todos os outros mineradores forem honestos, é $(x/(1-x))^2$. Embora fee sniping tenha uma probabilidade geral menor de sucesso do que a mineração honesta, tentar a mineração desonesta pode ser a escolha mais lucrativa se as transações no bloco anterior pagarem taxas significativamente mais altas do que as transações atualmente no mempool — uma pequena chance de ganhar uma grande quantia pode valer mais do que uma grande chance de ganhar uma quantia pequena.

— David Harding, fee sniping, Bitcoin Optech website

Jogando um balde de água fria sobre nossas esperanças para o futuro está o fato de que, se os mineradores começarem a realizar fee sniping, isso incentivará outros a fazerem o mesmo, deixando ainda menos mineradores honestos. Isso poderia prejudicar gravemente a segurança geral do Bitcoin. Harding continua a listar algumas contramedidas que podem ser adotadas, como confiar em bloqueios de tempo de transações para restringir onde na blockchain a transação pode aparecer.



Então, dado que o consenso sobre o fornecimento finito permanece, o subsídio de bloco - graças ao [BIP42](#), que corrigiu um bug de inflação de muito longo prazo - chegará a zero por volta do ano 2140. As taxas de transação serão suficientes para garantir a rede a partir de então? É impossível dizer, mas sabemos algumas coisas:

- Um século é um *longo* tempo do ponto de vista do Bitcoin. Se ele ainda existir, provavelmente terá evoluído enormemente.

- Se uma esmagadora maioria econômica achar necessário mudar as regras e introduzir, por exemplo, uma inflação monetária anual perpétua de 0,1% ou 1%, o fornecimento de bitcoin não será mais finito.
- Com subsídio de bloco zero e um mempool vazio ou quase vazio, as coisas podem se tornar instáveis devido ao fee sniping.



Como a transição para uma recompensa de bloco baseada apenas em taxas está tão longe no futuro, pode ser sensato não tirar conclusões precipitadas e tentar resolver os possíveis problemas enquanto podemos. Por exemplo, Peter Todd acha que existe um risco real de que o orçamento de segurança do Bitcoin não seja suficiente no futuro e, conseqüentemente, argumenta a favor de uma pequena inflação perpétua no Bitcoin. No entanto, ele também acha que não é uma boa ideia discutir essa questão agora, como [ele disse no podcast What Bitcoin Did](#):

Mas, esse é um risco para daqui a 10, 20 anos no futuro. Isso é um tempo muito longo. E, até lá, quem diabos sabe quais serão os riscos?

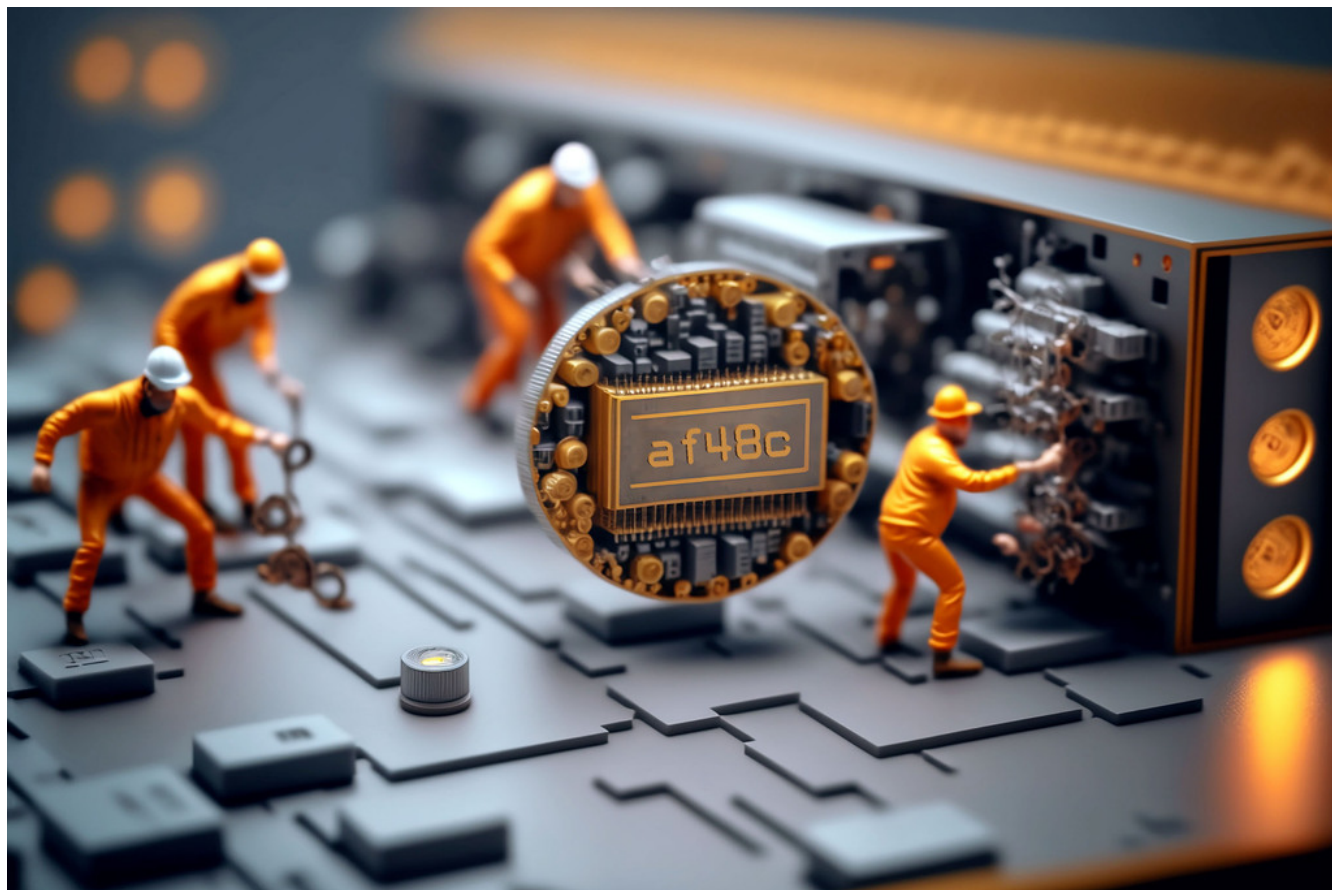
— Peter Todd on security budget, What Bitcoin Did podcast (2019)

Talvez possamos pensar no Bitcoin como algo orgânico. Imagine uma pequena planta de carvalho crescendo lentamente. Imagine também que você nunca viu uma árvore totalmente crescida em sua vida. Não seria prudente então refrear suas questões de controle em vez de definir com antecedência todas as regras sobre como essa planta deve evoluir e crescer?

4.2. Conclusão

Se o fornecimento de bitcoin ultrapassará 21 milhões, não podemos dizer hoje, e isso provavelmente não é tão ruim. Garantir que o orçamento de segurança permaneça alto o suficiente é crucial, mas não urgente. Vamos ter essa discussão daqui a 10-50 anos, quando soubermos mais. Se ainda for relevante.

Chapter 5. Atualizações



Atualizar o Bitcoin de maneira segura pode ser extremamente difícil. Algumas mudanças levam vários anos para serem implementadas. Neste capítulo, aprendemos sobre o vocabulário comum em torno da atualização do Bitcoin e exploramos alguns exemplos de atualizações históricas em seu protocolo, bem como as lições que tiramos delas. Por fim, falamos sobre divisões de cadeia e os riscos e custos relacionados a elas.



Para se sintonizar com este capítulo, você deve ler [O artigo de David Harding sobre harmonia e discord](#).

Especialistas em Bitcoin falam frequentemente sobre consenso, cujo significado é abstrato e difícil de definir. Mas a palavra consenso evoluiu da palavra latina *concentus*, "um cantar juntos, harmonia,"[1] então vamos falar não de consenso do Bitcoin, mas de harmonia do Bitcoin.

A harmonia é o que faz o Bitcoin funcionar. Milhares de full nodes trabalham independentemente para verificar se as transações que recebem são válidas, produzindo um acordo harmonioso sobre o estado do livro-razão do Bitcoin sem que nenhum operador de node precise confiar em mais ninguém. É semelhante a um coro em que cada membro canta a mesma música ao mesmo tempo para produzir algo muito mais bonito do que qualquer um deles poderia produzir sozinho.

O resultado da harmonia do Bitcoin é um sistema onde os bitcoins estão seguros não apenas de ladrões comuns (desde que você mantenha suas chaves seguras), mas também da inflação infinita, confisco em massa ou direcionado, ou simplesmente da confusão burocrática que é o sistema financeiro legado.

— David Harding, *Harmonia e Discórdia*

Este capítulo discute como o Bitcoin pode ser atualizado sem causar discórdia. Manter a harmonia, ou seja, manter o consenso, é de fato um dos maiores desafios no desenvolvimento do Bitcoin. Existem muitas nuances nos mecanismos de atualização, que podem ser melhor compreendidas estudando casos reais de atualizações anteriores. Por esse motivo, o capítulo foca muito em exemplos históricos e começa estabelecendo o cenário com algum vocabulário útil.

5.1. Vocabulário



De acordo com a Wikipedia, [compatibilidade com versões futuras](#) refere-se à condição em que um software antigo pode processar dados criados por softwares mais novos, ignorando as partes que ele não entende.

Um padrão suporta compatibilidade com versões futuras se um produto que está em conformidade com versões anteriores pode "processar graciosamente" a entrada projetada para versões mais recentes do padrão, ignorando novas partes que ele não entende.

— Compatibilidade com versões futuras, Wikipedia



Vice-versa, [compatibilidade com versões anteriores](#) refere-se a quando dados de um software antigo são utilizáveis em softwares mais novos. Uma mudança é dita totalmente compatível se for compatível tanto com versões futuras quanto com versões anteriores.

Uma mudança nas regras de consenso do Bitcoin é considerada um **soft fork** se for totalmente compatível. Esta é a maneira mais comum de atualizar o Bitcoin, por várias razões que discutiremos mais adiante neste capítulo. Se uma mudança nas regras de consenso do Bitcoin for compatível com versões anteriores mas não com versões futuras, é chamada de **hard fork**.



Para uma visão técnica sobre soft forks e hard forks, leia o [capítulo 11 do Grokking Bitcoin](#). Ele explica esses termos e também aprofunda-se nos mecanismos de atualização. É recomendado, embora não seja estritamente necessário, compreender isso antes de continuar a leitura.

5.2. Atualizações históricas



O Bitcoin não é o mesmo hoje que era quando o bloco gênese foi criado. Várias atualizações foram feitas ao longo dos anos. Em 2017, Eric Lombrozo [falou na conferência Breaking Bitcoin](#) sobre os diferentes mecanismos de atualização do Bitcoin, apontando como eles evoluíram ao longo do tempo. Ele até explicou como Satoshi Nakamoto uma vez atualizou o Bitcoin por meio de um hard fork.

Houve na verdade um hard fork no Bitcoin que Satoshi fez que nós nunca faríamos dessa forma - é uma maneira bastante ruim de fazer isso. Se você olhar para a descrição do commit no git aqui [\[757f076\]](#), ele diz algo sobre reverter makefile.unix versão wx-config 0.3.6. Certo. Isso é tudo o que diz. Não há nenhuma indicação de que seja uma mudança significativa. Ele basicamente estava escondendo isso lá. Ele também [postou no bitcointalk](#) e disse, por favor, atualizem para 0.3.6 o mais rápido possível. Corrigimos um bug de implementação onde é possível que transações falsas possam ser exibidas como aceitas. Não aceitem pagamentos em bitcoin até que atualizem para 0.3.6. Se não puderem atualizar imediatamente, seria melhor desligar o seu node do Bitcoin até que o façam. E além disso, não sei por que ele decidiu fazer isso também, ele decidiu adicionar algumas otimizações no mesmo código. Corrigir um bug e adicionar algumas otimizações.

— Eric Lombrozo, Mudando Regras de Consenso Sem Quebrar o Bitcoin na conferência Breaking Bitcoin (2017)



Ele aponta que, intencionalmente ou não, esse hard fork criou oportunidades para futuros soft forks, nomeadamente os operadores de Script (opcodes) OP_NOP1-OP_NOP10. Vamos olhar mais de perto essa mudança de código em [Section 8.2.1](#). Esses opcodes foram utilizados para dois soft forks até agora: [BIP65](#) (OP_CHECKLOCKTIMEVERIFY), e [BIP113](#) (OP_SEQUENCEVERIFY).

Lombrozo também fornece uma visão geral de como os mecanismos de atualização evoluíram ao longo dos anos, até 2017. Desde então, apenas uma outra grande atualização, Taproot (analisada em [Section 5.2.3](#)), foi implantada. O longo e um tanto caótico processo que levou à sua ativação nos ajudou a ganhar mais insights sobre os mecanismos de atualização no Bitcoin.

5.2.1. Atualização do Segwit

Enquanto todas as atualizações anteriores ao Segwit foram mais ou menos indolores, esta foi diferente. Quando o código de ativação do Segwit foi lançado, em outubro de 2016, parecia haver um apoio esmagador para ele entre os usuários de Bitcoin, mas por algum motivo os mineradores não sinalizaram apoio para essa atualização, o que estagnou a ativação sem uma resolução à vista.



Aaron van Wirdum descreve essa estrada sinuosa em seu artigo na Bitcoin Magazine [A Longa Estrada para o Segwit](#). Ele começa explicando o que é o Segwit e como isso se relaciona com o

debate sobre o tamanho dos blocos. Van Wierdum então delineia a sucessão de eventos que levou à sua ativação final. No centro desse processo estava um mecanismo de atualização chamado *soft fork ativado pelo usuário*, ou UASF em resumo, que foi proposto pelo usuário Shaolinfry.

Shaolinfry propôs uma alternativa: um soft fork ativado pelo usuário (UASF). Em vez de ativação pela potência de hash, um soft fork ativado pelo usuário teria uma "ativação por data marcada", onde os nodes começariam a aplicação em um momento predeterminado no futuro. Desde que tal UASF seja aplicado por uma maioria econômica, isso deve compelir uma maioria de mineradores a seguir (ou ativar) o soft fork.

— Aaron van Wierdum, *The Long Road To Segwit* na Bitcoin Magazine (2017)



Entre outras coisas, ele cita o e-mail de Shaolinfry para a lista de discussão Bitcoin-dev. Naquela ocasião, Shaolinfry [argumentou contra soft forks ativados por mineradores](#), listando uma série de problemas com eles.

Em primeiro lugar, requer confiar que a potência de hash validará após a ativação. O soft fork do BIP66 foi um caso em que 95% do hashrate estava sinalizando prontidão, mas na realidade cerca de metade não estava realmente validando as regras atualizadas e mineraram um bloco inválido por engano[1].

Em segundo lugar, a sinalização dos mineradores tem um veto natural, que permite a uma pequena porcentagem do hashrate vetar a ativação do node para a atualização para todos. Até agora, os soft forks têm se aproveitado do panorama relativamente centralizado da mineração, onde há relativamente poucos pools de mineração construindo blocos válidos; à medida que avançamos para uma maior descentralização do hashrate, é provável que soframos cada vez mais com a "inércia da atualização", que vetará a maioria das atualizações.

— Shaolinfry, lista de discussão Bitcoin-dev (2017)

Shaolinfry também chamou a atenção para uma interpretação comum equivocada da sinalização dos mineradores: as pessoas geralmente pensavam que era um meio pelo qual os mineradores podiam decidir sobre as atualizações do protocolo, em vez de uma ação que ajudava a coordenar as atualizações. Devido a esse mal-entendido, os mineradores podem ter se sentido obrigados a proclamar publicamente suas opiniões sobre um determinado soft fork, como se isso desse peso à proposta.

A proposta de UASF é, em poucas palavras, uma “data marcada” na qual os nodes começam a

aplicar novas regras específicas. Dessa forma, os mineradores não precisam fazer um esforço coletivo para coordenar a atualização, mas *podem* desencadear a ativação antes da data marcada se um número suficiente de blocos sinalizar apoio.

Minha sugestão é ter o melhor dos dois mundos. Como um soft fork ativado pelo usuário precisa de um tempo de preparação relativamente longo antes da ativação, podemos combinar com o BIP9 para dar a opção de uma ativação coordenada pela potência de hash mais rápida ou ativação por data marcada, o que ocorrer primeiro. Em ambos os casos, podemos aproveitar os sistemas de alerta no BIP9. A mudança é relativamente simples, adicionando um parâmetro de tempo de ativação que transicionará o estado BIP9 para LOCKED_IN antes do fim do tempo limite de implantação do BIP9.

— Shaolinfry, lista de discussão Bitcoin-dev (2017)



Essa ideia gerou muito interesse, mas não parecia alcançar um apoio quase unânime, o que causou preocupação com uma possível divisão da cadeia. O artigo de Aaron van Wirdum explica como isso foi finalmente resolvido graças ao [BIP91](#), autoria de James Hilliard.

Hilliard propôs uma solução ligeiramente complexa, mas inteligente, que tornaria tudo compatível: ativação do Segregated Witness conforme proposto pela equipe de desenvolvimento do Bitcoin Core, o UASF BIP148 e o mecanismo de ativação do Acordo de Nova York. Seu BIP91 poderia manter o Bitcoin inteiro – pelo menos durante a ativação do SegWit.

— Aaron van Wirdum, The Long Road To Segwit na Bitcoin Magazine (2017)

Havia alguns fatores complicadores adicionais envolvidos (por exemplo, o chamado "Acordo de Nova York"), que este BIP teve que levar em consideração. Recomendamos que você leia o artigo de Van Wirdum na íntegra para conhecer os muitos detalhes interessantes dessa história.

5.2.2. Discussão pós-Segwit



Após a implantação do Segwit, uma discussão sobre mecanismos de implantação emergiu. Como observado por Eric Lombrozo em [sua palestra na conferência Breaking Bitcoin](#) e por Shaolinfry (veja [Section 5.2.1](#) acima), um soft fork ativado por mineradores não é o mecanismo de atualização ideal.

Em algum momento, provavelmente vamos querer adicionar mais recursos ao protocolo do bitcoin. Esta é uma grande questão filosófica que estamos nos perguntando. Fazemos um UASF para o próximo? E quanto a uma abordagem híbrida? Ativação por mineradores sozinha foi descartada. bip9 não vamos usar novamente.

— Eric Lombrozo, Mudando Regras de Consenso Sem Quebrar o Bitcoin na conferência Breaking Bitcoin (2017)



Em janeiro de 2020, Matt Corallo [enviou um e-mail](#) para a lista de discussão Bitcoin-dev que iniciou uma discussão sobre mecanismos futuros de implantação de soft fork. Ele listou cinco objetivos que ele considerava essenciais em uma atualização. David Harding [os resume em um boletim da Bitcoin Optech](#) como:

1. A capacidade de abortar se uma objeção séria às mudanças propostas nas regras de consenso for encontrada
2. A alocação de tempo suficiente após o lançamento do software atualizado para garantir que a maioria dos nodes econômicos sejam atualizados para aplicar essas regras
3. A expectativa de que a taxa de hash da rede será aproximadamente a mesma antes e depois da mudança, bem como durante qualquer transição
4. A prevenção, tanto quanto possível, da criação de blocos que sejam inválidos sob as novas regras, o que poderia levar a falsas confirmações em nodes não atualizados e clientes SPV
5. A garantia de que os mecanismos de abortar não possam ser usados de forma inadequada por provocadores ou partidários para impedir uma atualização amplamente desejada sem problemas conhecidos

— David Harding, Bitcoin Optech newsletter #80 (2020)

O que Corallo propõe é uma combinação de um soft fork ativado por mineradores e um soft fork ativado por usuários:

Assim, como algo um pouco mais concreto, acho que um método de ativação que define o precedente correto e considera adequadamente os objetivos acima, seria:

- 1) uma implantação padrão do BIP 9 com um horizonte de tempo de um ano para ativação com 95% de prontidão dos mineradores,
- 2) no caso de nenhuma ativação ocorrer dentro de um ano, um período de tranquilização de seis meses durante o qual a comunidade pode analisar e discutir as razões para a falta de ativação e,
- 3) no caso de fazer sentido, um simples comando/bitcoin.conf parâmetro que foi suportado desde o lançamento original da implantação permitiria que os usuários optassem por uma implantação BIP 8 com um horizonte de tempo de 24 meses para ativação por data marcada (bem como um novo lançamento do Bitcoin Core ativando a data marcada universalmente).

Isso fornece um horizonte de tempo muito longo para uma ativação mais padrão, enquanto ainda garante que os objetivos do item #5 sejam atendidos, mesmo que, nesses casos, o horizonte de tempo precise ser significativamente estendido para atender aos objetivos do item #3. Desenvolver o Bitcoin não é uma corrida. Se for necessário, esperar 42 meses garante que não estamos estabelecendo um precedente negativo que vamos nos arrepender enquanto o Bitcoin continua a crescer.

— Matt Corallo, Modern Soft Fork Activation na lista de discussão Bitcoin-dev (2020)

5.2.3. Atualização do Taproot - Speedy Trial

Quando o Taproot estava pronto para implantação em outubro de 2020, ou seja, todos os detalhes técnicos em torno de suas regras de consenso haviam sido implementados e haviam alcançado ampla aprovação dentro da comunidade, as discussões sobre como realmente implantar começaram a esquentar. Essas discussões haviam sido bastante discretas até aquele momento.



Muitas propostas para mecanismos de ativação começaram a circular, e David Harding [as resumiu na Wiki do Bitcoin](#). Em seu artigo, ele explicou algumas propriedades do BIP8, que na época tinha algumas mudanças recentes feitas para torná-lo mais flexível.

No momento em que este documento está sendo escrito, [BIP8](#) foi redigido com base nas lições aprendidas em 2017. Uma mudança notável após os BIPs 9+148 é que a ativação forçada agora é baseada na altura do bloco em vez de mediana do tempo passado; uma segunda mudança notável é que a ativação forçada é um parâmetro booleano escolhido quando os parâmetros de ativação de um soft fork são definidos, seja para a implantação inicial ou atualizado em uma implantação posterior.

BIP8 sem ativação forçada é muito semelhante ao [BIP9](#) versão bits com tempo limite e atraso, com a única diferença significativa sendo o uso de alturas de bloco pelo BIP8 em comparação com o uso de mediana do tempo passado pelo BIP9. Esta configuração permite que a tentativa falhe (mas pode ser tentada novamente mais tarde).

O BIP8 com ativação forçada conclui com um período de sinalização obrigatória onde todos os blocos produzidos em conformidade com suas regras devem sinalizar prontidão para o soft fork de uma forma que desencadeie ativação em uma implantação anterior do mesmo soft fork com ativação não obrigatória. Em outras palavras, se a versão do node x for lançada sem ativação forçada e, mais tarde, a versão y for lançada que for bem-sucedida em forçar os mineradores a começar a sinalizar prontidão dentro do mesmo período de tempo, ambas as versões começarão a aplicar as novas regras de consenso ao mesmo tempo.

Essa flexibilidade da proposta revisada do BIP8 torna possível expressar algumas outras ideias em termos de como seriam usando o BIP8. Isso fornece um fator comum para categorizar muitas propostas diferentes.

— David Harding, Taproot Activation Proposals na Wiki do Bitcoin (2020)

A partir deste ponto, as discussões se tornaram muito acaloradas, especialmente sobre se `lockinontimeout` deveria ser `true` (como em um soft fork ativado por usuários, referido como “BIP8 com ativação forçada” por Harding) ou `false` (como em um soft fork ativado por mineradores, referido como “BIP8 sem ativação forçada” por Harding).

Entre as propostas listadas, uma delas foi intitulada “Vamos ver o que acontece”. Por algum motivo, essa proposta não ganhou muita tração até sete meses depois.



Durante esses sete meses, a discussão continuou e parecia não haver como alcançar um consenso amplo sobre qual mecanismo de implantação usar. Havia principalmente dois grupos: um que preferia `lockinontimeout=true` (o grupo UASF) e o outro que preferia `lockinontimeout=false` (o grupo

“tente e se falhar repense”). Como não havia um apoio esmagador para nenhuma dessas opções, o debate girava em círculos sem uma forma aparente de avançar. Algumas dessas discussões foram realizadas no IRC, em um canal chamado `##taproot-activation`, mas [em 5 de março de 2021](#), algo mudou:

```
06:42 <harding> roconnor: alguém está propondo BIP8(3m, false)? Eu mencionei
isso outro dia, mas não vi nenhuma resposta.
[...]
06:43 <willcl_ark_> Curiosamente, eu estava pensando comigo mesmo que, em
comparação, a ativação do SegWit foi realmente bastante simples: simplesmente um
LOT=false e, se falhar, um UASF.
06:43 <maybehuman> é engraçado, "vamos ver o que acontece" (ou seja, false, 3m)
era uma escolha popular bem no começo deste canal, se bem me lembro
06:44 <roconnor> harding: Acho que sou eu. Não sei o quanto isso vale.
Principalmente acho que seria uma configuração amplamente aceitável com base na
minha compreensão das preocupações de todos.
06:44 <willcl_ark_> maybehuman: porque todo mundo realmente quer isso, até os
mineradores disseram que poderiam atualizar em cerca de duas semanas (ou pelo
menos o f2pool disse isso)
06:44 <roconnor> harding: BIP8(3m,false) com um período de lockin estendido.
06:45 <harding> roconnor: oh, que bom. Tem sido minha opção favorita desde que
resumi as opções na wiki, tipo, sete meses atrás.
06:45 <@michaelfolkson> UASF não lançaria (true,3m), mas sim Core poderia lançar
(false, 3m)
06:45 <willcl_ark_> harding: Certamente parece uma boa abordagem para mim. _se_
isso falhar, então você pode tentar entender o porquê, sem desperdiçar muito
tempo
```

— log do IRC `##taproot-activation`



A abordagem “vamos ver o que acontece” finalmente pareceu fazer sentido na mente das pessoas. Esse processo seria posteriormente rotulado como “Speedy Trial” devido ao seu curto período de sinalização. David Harding explica essa ideia para a comunidade mais ampla em um [e-mail para a lista de discussão Bitcoin-dev](#).

A versão anterior desta proposta foi documentada há mais de 200 dias[3] e o código subjacente do taproot foi incorporado ao Bitcoin Core há mais de 140 dias atrás.[4] Se tivéssemos começado o Speedy Trial na época em que o taproot foi incorporado (o que é um pouco irrealista), estaríamos a menos de dois meses de distância de ter o taproot ou já teríamos passado para a próxima tentativa de ativação há mais de um mês.

Em vez disso, debatemos exaustivamente e não parece que estejamos mais próximos de uma solução amplamente aceitável do que quando a lista de discussão começou a discutir esquemas de ativação pós-segwit há mais de um ano.[5] Eu acho que o Speedy Trial é uma maneira de gerar progresso rápido que ou terminará o debate (por enquanto, se a ativação for bem-sucedida) ou nos dará alguns dados reais sobre os quais basear propostas futuras de ativação do taproot.

— David Harding na lista de discussão Bitcoin-dev

Esse mecanismo de implantação foi refinado ao longo de dois meses e depois lançado na [versão 0.21.1 do Bitcoin Core](#). Os mineradores rapidamente começaram a sinalizar para essa atualização, movendo o estado de implantação para `LOCKED_IN`, e após um período de carência as regras do Taproot foram ativadas em meados de novembro de 2021 no bloco [709632](#).

5.2.4. Mecanismos de implantação futuros

Dado os problemas com os soft forks recentes, Segwit e Taproot, não está claro como a próxima atualização será implantada. Speedy Trial foi usado para implantar o Taproot, mas foi usado para preencher o abismo entre os grupos UASF e MASF, não porque tenha surgido como o melhor mecanismo de implantação conhecido.

5.3. Riscos

Durante a ativação de qualquer fork, seja ele hard ou soft, ativado por mineradores ou por usuários, existe o risco de uma divisão de cadeia prolongada. Uma divisão que se prolonga por mais de alguns blocos pode causar danos severos ao sentimento em torno do Bitcoin, bem como ao seu preço. Mas acima de tudo, causaria grande confusão sobre o que é o Bitcoin. O Bitcoin é esta cadeia ou aquela cadeia?

O risco com um soft fork ativado por usuários é que as novas regras sejam ativadas mesmo que a maioria da potência de hash não as suporte. Este cenário resultaria em uma divisão de cadeia prolongada, que persistiria até que a maioria da potência de hash adotasse as novas regras. Poderia ser especialmente difícil incentivar os mineradores a mudar para a nova cadeia se eles já tivessem minerado blocos após a divisão na cadeia antiga, porque ao mudar de ramo estariam abandonando suas próprias recompensas de bloco. No entanto, vale a pena mencionar um episódio notável: em março de 2013 ocorreu uma divisão prolongada, explicada em [Section 8.2.3](#), devido a um hard fork não intencional e, ao contrário desse incentivo, dois grandes pools de mineração tomaram a decisão

de abandonar seu ramo da divisão para restaurar o consenso.

Por outro lado, o risco com um soft fork ativado por mineradores é consequência do fato de que os mineradores podem se envolver em sinalização falsa, o que significa que a participação real da potência de hash que apoia a mudança pode ser menor do que parece. Se o suporte real não compreender a maioria da potência de hash, provavelmente veremos uma divisão de cadeia prolongada semelhante à descrita no parágrafo anterior. Isso, ou pelo menos um problema semelhante, aconteceu na realidade quando o BIP66 foi implantado (veja [Section 8.2.4](#)), mas foi resolvido em cerca de 6 blocos ou algo assim.

5.3.1. Custos de uma divisão



Jimmy Song [falou sobre os custos associados a hard forks](#) na conferência Breaking Bitcoin em Paris, mas muito do que ele disse se aplica a uma divisão de cadeia devido a um soft fork falho também. Ele falou sobre *externalidades negativas* e as definiu como o preço que outra pessoa tem que pagar por suas próprias ações.

O exemplo clássico de uma externalidade negativa é uma fábrica. Talvez eles estejam produzindo - talvez seja uma refinaria de petróleo e eles produzam um bem que é bom para a economia, mas também produzem algo que é uma externalidade negativa, como poluição. Não é apenas algo que todos têm que pagar para limpar ou sofrer. Mas também são efeitos de segunda e terceira ordem, como mais tráfego indo em direção à fábrica como resultado de mais trabalhadores que precisam ir para lá. Você pode também ter - você pode colocar em perigo alguma vida selvagem ao redor. Não é que todos tenham que pagar pelas externalidades negativas, pode ser pessoas específicas, como as pessoas que estavam usando aquela estrada anteriormente ou animais que estavam perto daquela fábrica, e eles também estão pagando pelo custo daquela fábrica.

— Jimmy Song, Custos Socializados de Hard Forks na conferência Breaking Bitcoin (2017)

No contexto do Bitcoin, ele exemplifica externalidades negativas usando o Bitcoin Cash (bcash), que é um hard fork do Bitcoin criado pouco antes daquela conferência em 2017. Ele categoriza as externalidades negativas de um hard fork em custos únicos e custos permanentes.

Entre os muitos exemplos de custos únicos, ele menciona os incorridos pelas exchanges.

Então, temos um monte de exchanges e elas tiveram muitos custos únicos que tiveram que pagar. A primeira coisa que aconteceu foi que os depósitos e retiradas tiveram que ser interrompidos por um ou dois dias para essas exchanges porque eles não sabiam o que aconteceria. Muitas dessas exchanges tiveram que recorrer a cold storage porque seus usuários estavam exigindo bcash. É parte do seu dever fiduciário, eles têm que fazer isso. Você também tem que auditar o novo software. Isso é algo que tivemos que fazer na itbit. Queremos gastar bcash - como fazemos isso? Temos que baixar o electron cash? Ele tem malware? Tivemos que auditá-lo. Tivemos como 10 dias para descobrir se isso estava ok ou não. E então você tem que decidir, vamos permitir uma retirada única ou vamos listar essa nova moeda? Para uma exchange listar uma nova moeda, não é fácil - há todo tipo de novos procedimentos para cold storage, assinaturas, depósitos, retiradas. Ou você poderia simplesmente ter este evento único onde você dá o bcash para seus usuários em algum momento e nunca mais pensa nisso. Mas isso tem seus problemas também. E, finalmente, e de qualquer maneira que você faça, retiradas ou listagem - você precisará de nova infraestrutura para trabalhar com este token de alguma forma, mesmo que seja uma retirada única. Você precisa de alguma forma de dar esses tokens aos seus usuários. De novo, com pouco aviso. Certo? Sem tempo para fazer isso, tem que ser feito rapidamente.

— Jimmy Song, Custos Socializados de Hard Forks na conferência Breaking Bitcoin (2017)

Ele também lista os custos únicos incorridos por comerciantes, processadores de pagamento, carteiras, mineradores e usuários, bem como alguns dos custos permanentes, por exemplo, perda de privacidade e um maior risco de reorganizações (reorgs).

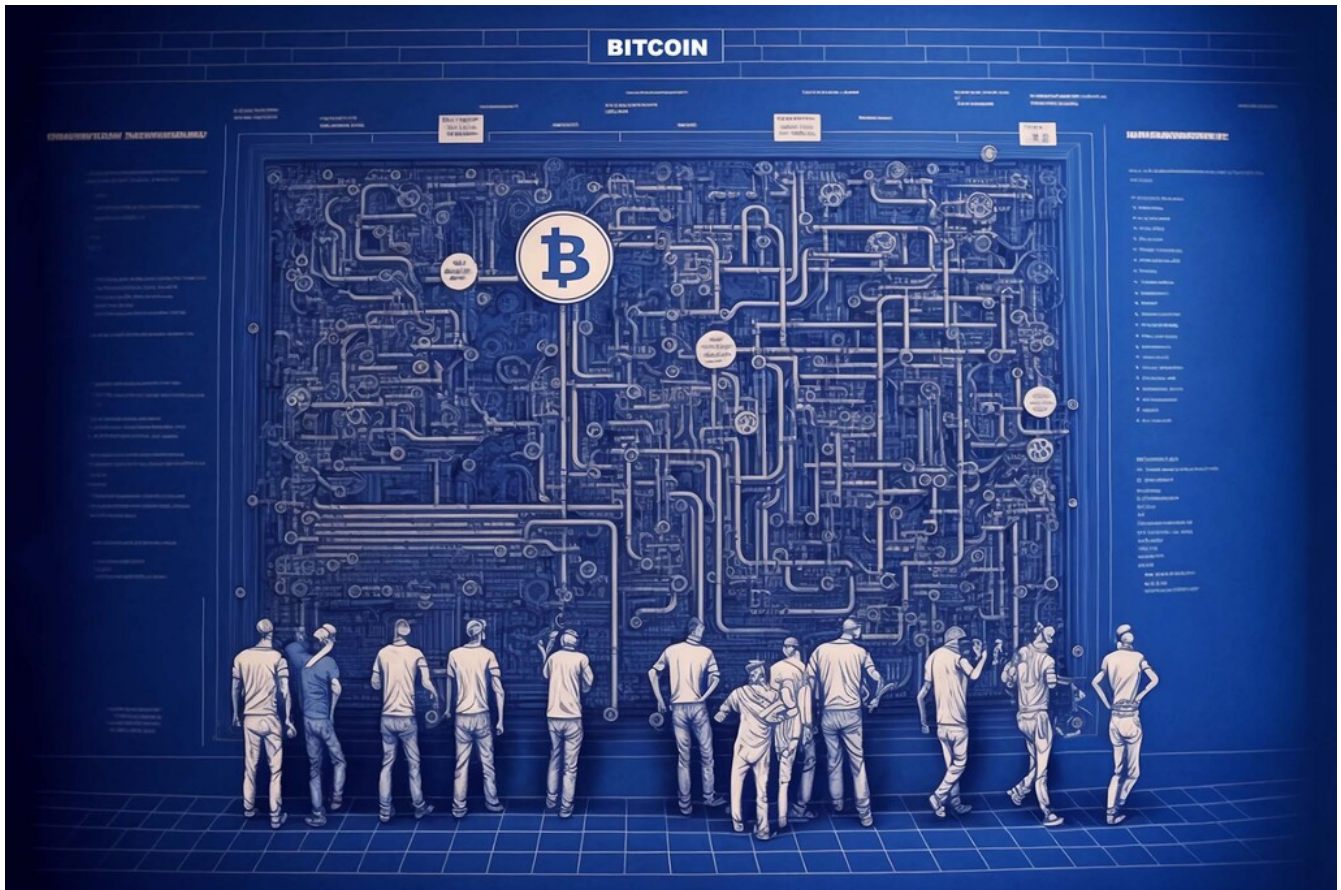
De fato, quando ocorre uma divisão e a cadeia com as regras mais gerais se torna mais forte do que a cadeia com as regras mais rígidas, ocorrerá uma reorganização. Isso terá um impacto severo em todas as transações realizadas no ramo eliminado. Por essas razões, é realmente importante tentar evitar divisões de cadeia a todo custo.

5.4. Conclusão

O Bitcoin cresce e evolui com o tempo. Diferentes mecanismos de atualização foram usados ao longo dos anos e a curva de aprendizado é íngreme. Métodos cada vez mais sofisticados e robustos continuam sendo inventados, à medida que aprendemos mais sobre como a rede reage.

Para manter o Bitcoin em harmonia, os soft forks têm se mostrado o caminho a seguir, mas a grande questão ainda não foi totalmente respondida: como implantar soft forks com segurança sem causar discórdia?

Chapter 6. Open Source



Bitcoin é construído utilizando software de código aberto. Neste capítulo, analisamos o que isso significa, como funciona a manutenção do software, e como o software de código aberto no Bitcoin permite o desenvolvimento sem permissões. Exploramos brevemente *seleção criptográfica*, que trata da escolha e uso de bibliotecas em sistemas criptográficos. O capítulo inclui uma seção sobre o processo de revisão do Bitcoin, seguida por outra sobre as formas como os desenvolvedores de Bitcoin são financiados. A última seção discute como a cultura de código aberto do Bitcoin pode parecer realmente estranha de fora, e por que essa estranheza percebida é, na verdade, um sinal de boa saúde.



A maioria dos softwares de Bitcoin, e especialmente o Bitcoin Core, são de código aberto. Isso significa que o código-fonte do software é disponibilizado ao público para escrutínio, modificação e redistribuição. A definição de código aberto em <https://opensource.org/osd> inclui, entre outros, os seguintes pontos importantes:

Free Redistribution

A licença não deve restringir nenhuma parte de vender ou doar o software como um componente de uma distribuição de software agregada contendo programas de várias fontes diferentes. A licença não deve exigir um royalty ou outra taxa por essa venda.

Source Code

O programa deve incluir código-fonte e permitir a distribuição tanto em código-fonte quanto em forma compilada. Onde alguma forma de um produto não é distribuída com código-fonte, deve haver um meio bem divulgado de obter o código-fonte por não mais do que um custo razoável de reprodução, preferencialmente baixando pela Internet sem custo. O código-fonte deve ser a forma preferida em que um programador modificaria o programa. Código-fonte deliberadamente ofuscado não é permitido. Formas intermediárias, como a saída de um pré-processador ou tradutor, não são permitidas.

Derived Works

A licença deve permitir modificações e trabalhos derivados, e deve permitir que eles sejam distribuídos sob os mesmos termos da licença do software original.

— The Open Source Definition, Open Source Initiative website



O Bitcoin Core adere a essa definição sendo distribuído sob a [Licença MIT](#):

The MIT License (MIT)

Copyright (c) 2009-2022 The Bitcoin Core developers

Copyright (c) 2009-2022 Bitcoin Developers

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

Como mencionado em [Section 2.1](#), é importante que os usuários possam verificar que o software Bitcoin que utilizam “funciona conforme anunciado”. Para isso, eles devem ter acesso irrestrito ao código-fonte do software que desejam verificar.

Nas seções a seguir, mergulhamos em alguns outros aspectos interessantes do software de código aberto no Bitcoin.

6.1. Manutenção do software



O código-fonte do Bitcoin Core é mantido em um repositório Git hospedado no [GitHub](#). Qualquer pessoa pode clonar esse repositório sem pedir permissão e, em seguida, inspecionar, construir ou fazer alterações nele localmente. Isso significa que existem milhares de cópias do repositório espalhadas por todo o mundo. Estas são todas cópias do mesmo repositório, então o que torna esse repositório específico do GitHub tão especial? Tecnicamente, não é especial de forma alguma, mas socialmente tornou-se o ponto focal do desenvolvimento do Bitcoin.



O especialista em Bitcoin e segurança Jameson Lopp explica isso muito bem em um [post no blog](#) intitulado “Quem controla o Bitcoin Core?”:

Bitcoin Core é um ponto focal para o desenvolvimento do protocolo Bitcoin, em vez de um ponto de comando e controle. Se ele deixasse de existir por qualquer motivo, um novo ponto focal surgiria - a plataforma de comunicação técnica sobre a qual se baseia (atualmente o repositório GitHub) é uma questão de conveniência, em vez de definição / integridade do projeto. Na verdade, já vimos o ponto focal do desenvolvimento do Bitcoin mudar de plataformas e até de nomes!

— Jameson Lopp, Who Controls Bitcoin Core? (2018)

Ele continua explicando como o software do Bitcoin Core é mantido e protegido contra alterações maliciosas no código. A principal conclusão de todo o artigo está resumida em seu final:

Ninguém controla o Bitcoin.

Ninguém controla o ponto focal para o desenvolvimento do Bitcoin.

— Jameson Lopp, Who Controls Bitcoin Core? (2018)



O desenvolvedor do Bitcoin Core Eric Lombrozo fala mais sobre o processo de desenvolvimento em seu [post no Medium](#) intitulado “O Processo de Mesclagem (Merge) do Bitcoin Core”.

Qualquer pessoa pode fazer um fork do repositório de código base e fazer alterações arbitrárias em seu próprio repositório. Eles podem compilar um cliente a partir de seu próprio repositório e executá-lo, se quiserem. Eles também podem fazer compilações binárias para outras pessoas executarem.

Se alguém quiser mesclar uma mudança que fez em seu próprio repositório no Bitcoin Core, pode enviar um pull request. Uma vez submetido, qualquer pessoa pode revisar as mudanças e comentar sobre elas, independentemente de ter ou não acesso de commit ao Bitcoin Core.

— Eric Lombrozo on Medium.com, The Bitcoin Core Merge Process (2017)

Vale a pena notar que os pull requests podem levar muito tempo para serem mesclados ao repositório pelos mantenedores, e isso geralmente se deve à falta de revisão, ver [Section 6.5](#), que é frequentemente causada pela falta de *revisores*.

Lombrozo também fala sobre o processo que envolve mudanças de consenso, mas isso está um pouco além do escopo deste capítulo. Veja [Chapter 5](#) para mais informações sobre como o protocolo Bitcoin é atualizado.

6.2. Desenvolvimento sem permissões

Estabelecemos que qualquer pessoa pode escrever código para o Bitcoin Core sem pedir permissão, mas isso não significa necessariamente que ele será mesclado ao repositório Git principal. Isso afeta qualquer modificação, desde a alteração de esquemas de cores da interface gráfica do usuário até a forma como as mensagens peer-to-peer são formatadas e até mesmo as regras de consenso, ou seja, o conjunto de regras que define uma blockchain válida.

Provavelmente, igualmente importante é que os usuários são livres para desenvolver sistemas em cima do Bitcoin, sem pedir permissão. Vimos inúmeros projetos de software bem-sucedidos que foram construídos sobre o Bitcoin, como:

Lightning Network

Uma rede de pagamento que permite o pagamento rápido de quantias muito pequenas. Requer poucas transações de Bitcoin na cadeia. Existem várias implementações interoperáveis, como [Core Lightning](#), [LND](#), [Eclair](#), e [Lightning Dev Kit](#).

CoinJoin

Várias partes colaboram para combinar seus pagamentos em uma única transação para dificultar o agrupamento de endereços (explicado em [Section 3.4](#)). Existem várias implementações.

Sidechains

Esse sistema pode bloquear uma moeda na blockchain do Bitcoin para desbloqueá-la em outra

blockchain. Isso permite que bitcoins sejam movidos para outra blockchain, chamada de sidechain, para usar os recursos disponíveis nessa sidechain. Exemplos incluem [Elements da Blockstream](#).

OpenTimestamps

Permite que você [timestamp um documento](#) na blockchain do Bitcoin de maneira privada. Você pode então usar esse timestamp para provar que um documento deve ter existido antes de um certo tempo.

Sem desenvolvimento sem permissões, muitos desses projetos não teriam sido possíveis. Como afirmado em [Section 1.3](#), se os desenvolvedores tivessem que pedir permissão para construir protocolos em cima do Bitcoin, apenas os protocolos permitidos pelo comitê central de desenvolvedores seriam desenvolvidos.

É comum que sistemas como os listados acima sejam licenciados como software de código aberto, o que, por sua vez, permite que as pessoas contribuam, reutilizem ou revisem seu código sem pedir permissão. O código aberto tornou-se o padrão ouro de licenciamento de software Bitcoin.

6.3. Desenvolvimento pseudônimo

Não ter que pedir permissão para desenvolver software Bitcoin traz uma opção interessante e importante: você pode escrever e publicar código, no Bitcoin Core ou em qualquer outro projeto de código aberto, sem revelar sua identidade.



Muitos desenvolvedores escolhem essa opção operando sob um pseudônimo e tentando mantê-lo desvinculado de sua verdadeira identidade. Os motivos para fazer isso podem variar de desenvolvedor para desenvolvedor. Um usuário pseudônimo é ZmnSCPxj. Entre outros projetos, ele contribui para o Bitcoin Core e para o Core Lightning, uma das várias implementações do Lightning Network. [Ele escreve](#) em sua página na web:

Eu sou ZmnSCPxj, uma pessoa da Internet gerada aleatoriamente. Meus pronomes são ele/dele.

Entendo que os seres humanos têm o desejo instintivo de conhecer minha identidade. No entanto, acho que minha identidade é largamente irrelevante e prefiro ser julgado pelo meu trabalho.

Se você está se perguntando se deve doar ou não, e se questiona sobre meu custo de vida ou minha renda, entenda que, falando propriamente, você deve doar para mim com base na utilidade que encontra em meus artigos e meu trabalho sobre Bitcoin e Lightning Network.

— ZmnSCPxj em sua página no GitHub



No caso dele, a razão para usar um pseudônimo é ser julgado por seus méritos e não por quem a pessoa ou pessoas por trás do pseudônimo é ou são. Curiosamente, ele revelou em um [artigo no CoinDesk](#) que o pseudônimo foi criado por um motivo diferente.

Minha razão inicial [para usar um pseudônimo] era simplesmente que eu estava preocupado em cometer um erro colossal; assim, ZmnSCPxj foi originalmente planejado para ser um pseudônimo descartável que poderia ser abandonado em tal caso. No entanto, parece que ele conquistou uma reputação majoritariamente positiva, então eu o mantive.

— Muitos desenvolvedores de Bitcoin estão optando por usar pseudônimos – Por bons motivos no CoinDesk (2021)



Usar um pseudônimo realmente permite que você fale mais livremente sem colocar sua reputação pessoal em risco caso diga algo estúpido ou cometa um grande erro. Como se viu, o pseudônimo dele se tornou muito respeitado e, em 2019, [ele até recebeu uma doação para desenvolvimento](#), o que por si só é um testemunho da natureza sem permissão do Bitcoin.

Provavelmente, o pseudônimo mais conhecido no Bitcoin é Satoshi Nakamoto. Não está claro por que ele escolheu ser pseudônimo, mas com o benefício da retrospectiva, foi provavelmente uma boa decisão por vários motivos:

- Como muitas pessoas especulam que Nakamoto possui uma grande quantidade de bitcoins, é imperativo para sua segurança financeira e pessoal manter sua identidade desconhecida.
- Como sua identidade é desconhecida, não há possibilidade de processar alguém, o que dificulta a ação de várias autoridades governamentais.
- Não há uma pessoa autoritária a quem recorrer, tornando o Bitcoin mais meritocrático e resiliente contra chantagens.

Observe que esses pontos não se aplicam apenas a Satoshi Nakamoto, mas a qualquer pessoa que trabalhe com Bitcoin ou que possua quantidades significativas da moeda, em graus variados.

6.4. Criptografia de seleção

Desenvolvedores de código aberto frequentemente utilizam bibliotecas de código aberto desenvolvidas por outras pessoas. Isso é uma parte natural e incrível de qualquer ecossistema saudável. Mas o software Bitcoin lida com dinheiro real e, à luz disso, os desenvolvedores precisam ser extremamente cuidadosos ao escolher as bibliotecas de terceiros das quais dependerão.



Em uma filosófica [palestra filosófica sobre criptografia](#), Gregory Maxwell quer redefinir o termo “criptografia”, que ele acredita ser muito restrito. Ele explica que, fundamentalmente, *a informação deseja ser livre*, e faz sua definição de criptografia baseada nisso:

Criptografia é a arte e a ciência que usamos para combater a natureza fundamental da informação, dobrá-la à nossa vontade política e moral e direcioná-la para fins humanos contra todas as chances e esforços para se opor a ela.

— Gregory Maxwell, *Criptografia de Seleção do Bitcoin* (2015)

Ele então introduz o termo *criptografia de seleção*, referindo-se à arte de selecionar ferramentas criptográficas, e explica por que isso é uma parte importante da criptografia. Isso gira em torno de como selecionar bibliotecas, ferramentas e práticas criptográficas, ou como ele diz, “o cripto sistema de escolher cripto sistemas”.

Usando exemplos concretos, ele mostra como a criptografia de seleção pode facilmente dar muito errado, e também propõe uma lista de perguntas que você poderia fazer a si mesmo ao praticá-la. Abaixo está uma versão resumida dessa lista:

1. O software é adequado para seus propósitos?
2. As considerações criptográficas estão sendo levadas a sério?
3. O processo de revisão... existe um?
4. Qual é a experiência dos autores?
5. O software é documentado?
6. O software é portátil?
7. O software é testado?
8. O software adota as melhores práticas?

Embora isso não seja o guia definitivo para o sucesso, pode ser muito útil passar por esses pontos ao fazer criptografia de seleção.

Devido às questões mencionadas acima por Maxwell, o Bitcoin Core se esforça bastante para [minimizar sua exposição a bibliotecas de terceiros](#). Claro, você não pode erradicar todas as dependências externas, caso contrário, teria que escrever tudo por conta própria, desde renderização de fontes até a implementação de chamadas de sistema.

6.5. Revisão

Esta seção é intitulada “Revisão”, em vez de “Revisão de Código”, porque a segurança do Bitcoin depende fortemente de revisão em múltiplos níveis, não apenas do código-fonte. Além disso, diferentes ideias requerem revisão em diferentes níveis: uma mudança nas regras de consenso exigiria uma revisão mais profunda em mais níveis em comparação com uma mudança no esquema de cores ou a correção de um erro de digitação.

No caminho para a adoção final, uma ideia geralmente passa por várias fases de discussão e

revisão. Algumas dessas fases estão listadas abaixo:

1. Uma ideia é postada na lista de e-mails Bitcoin-dev
2. A ideia é formalizada em uma Proposta de Melhoria do Bitcoin (BIP)
3. O BIP é implementado em um pull request (PR) para o Bitcoin Core
4. Mecanismos de implantação são discutidos
5. Alguns mecanismos de implantação concorrentes são implementados em pull requests para o Bitcoin Core
6. Os pull requests são mesclados no branch master
7. Os usuários escolhem se querem ou não usar o software

Em cada uma dessas fases, pessoas com diferentes pontos de vista e experiências revisam as informações disponíveis, seja o código-fonte, um BIP ou apenas uma ideia vagamente descrita. As fases geralmente não são realizadas de maneira estrita de cima para baixo; de fato, várias fases podem acontecer simultaneamente, e às vezes você vai e volta entre elas. Diferentes pessoas também podem fornecer feedback durante diferentes fases.



Um dos revisores de código mais prolíficos no Bitcoin Core é Jon Atack. Ele escreveu [um post no blog](#) sobre como revisar pull requests no Bitcoin Core. Ele enfatiza que um bom revisor de código se concentra em como adicionar valor da melhor forma.

Como iniciante, o objetivo é tentar adicionar valor, com simpatia e humildade, enquanto aprende o máximo possível.

Uma boa abordagem é não fazer disso sobre você, mas sim "Como posso servir da melhor forma?"

— Jon Atack, Como Revisar Pull Requests no Bitcoin Core (2020)

Ele destaca o fato de que a revisão é um fator limitante no Bitcoin Core. Muitas boas ideias ficam presas em um limbo onde nenhuma revisão ocorre, pendentes. Observe que revisar não é apenas benéfico para o Bitcoin, mas também uma ótima maneira de aprender sobre o software enquanto adiciona valor a ele, ao mesmo tempo. A regra geral de Atack é revisar de 5 a 15 PRs antes de fazer qualquer PR seu. Novamente, seu foco deve ser em como servir melhor à comunidade, não em como fazer seu próprio código ser mesclado. Além disso, ele destaca a importância de fazer a revisão no nível certo: é hora de nits e erros de digitação, ou o desenvolvedor precisa de uma revisão mais orientada ao conceito?

Uma boa primeira pergunta ao começar uma revisão pode ser: "O que é mais necessário aqui neste momento?" Responder a essa pergunta requer experiência e contexto acumulado, mas é uma pergunta útil para decidir como você pode adicionar mais valor no menor tempo possível.

— Jon Atack, Como Revisar Pull Requests no Bitcoin Core (2020)

A segunda metade do post consiste em algumas orientações práticas úteis sobre como realmente realizar a revisão e fornece links para documentação importante para leitura adicional.



A desenvolvedora do Bitcoin Core e revisora de código Gloria Zhao escreveu [um artigo](#) contendo perguntas que ela geralmente faz a si mesma durante uma revisão. Ela também afirma o que considera uma boa revisão.

Pessoalmente, acho que uma boa revisão é aquela em que me fiz muitas perguntas específicas sobre o PR e fiquei satisfeita com as respostas para elas.

...[snip]...

Naturalmente, começo com perguntas conceituais, depois perguntas relacionadas à abordagem e, por fim, perguntas sobre a implementação. Geralmente, acho inútil deixar comentários relacionados à sintaxe C++ em um PR em rascunho, e me sentiria rude ao voltar a "isso faz sentido?" depois que o autor já abordou mais de 20 de minhas sugestões de organização de código.

— Gloria Zhao, Perguntas Comuns de Revisão de PR no GitHub (2022)

A ideia dela de que uma boa revisão deve se concentrar no que é mais necessário em um momento específico está alinhada com o conselho de Jon Atack. Ela propõe uma lista de perguntas que você pode se fazer em vários níveis do processo de revisão, mas enfatiza que essa lista não é de forma alguma exaustiva nem uma receita pronta. A lista é ilustrada com exemplos da vida real do GitHub.

6.6. Financiamento

Muitas pessoas trabalham com desenvolvimento de código aberto para o Bitcoin, seja para o Bitcoin Core ou para outros projetos. Muitos o fazem em seu tempo livre sem receber nenhuma compensação, mas alguns desenvolvedores também são pagos para fazê-lo.

Empresas, indivíduos e organizações que têm interesse no sucesso contínuo do Bitcoin podem doar fundos para desenvolvedores, seja diretamente ou por meio de organizações que, por sua vez, distribuem os fundos para desenvolvedores individuais. Há também várias empresas focadas em Bitcoin que contratam desenvolvedores qualificados para deixá-los trabalhar em tempo integral no Bitcoin.

6.7. Choque cultural

Às vezes, as pessoas têm a impressão de que há muitas brigas internas e debates acalorados intermináveis entre os desenvolvedores de Bitcoin, e que eles são incapazes de tomar decisões.



Por exemplo, o mecanismo de implantação do Taproot, descrito em [Section 5.2.3](#), foi discutido por um longo período de tempo durante o qual se formaram dois “campos”. Um queria “falhar” a atualização se os mineradores não tivessem votado esmagadoramente nas novas regras após um determinado momento, enquanto o outro queria impor as regras após esse momento, não importando o que acontecesse. Michael Folkson resume os argumentos dos dois campos em um [e-mail](#) para a lista de e-mails Bitcoin-dev.



O debate continuou aparentemente para sempre, e era realmente difícil ver qualquer consenso sobre isso se formando em breve. Isso deixou as pessoas frustradas e, como resultado, a intensidade aumentou. Gregory Maxwell (como usuário nullc) preocupou-se [no Reddit](#) que as discussões prolongadas tornariam a atualização menos segura.

Neste ponto, esperar mais não está acrescentando mais revisão e certeza. Em vez disso, a demora adicional está drenando inércia e aumentando potencialmente o risco, à medida que as pessoas começam a esquecer detalhes, atrasando o trabalho no uso subsequente (como suporte a carteiras) e não investindo tanto esforço adicional na revisão quanto investiriam se sentissem confiança sobre o cronograma de ativação.

— Gregory Maxwell no Reddit, O desenvolvimento do Taproot está indo rápido ou devagar demais?

Eventualmente, essa disputa foi resolvida graças a uma nova proposta de David Harding e Russel O'Connor chamada Speedy Trial, que implicava em um período de sinalização comparativamente mais curto para que os mineradores bloqueassem a ativação do Taproot, ou falhassem rapidamente. Se eles ativassem durante esse período, então o Taproot seria implantado aproximadamente 6 meses depois. Esta atualização é coberta em mais detalhes em [Chapter 5](#).

Alguém que não está acostumado ao processo de desenvolvimento do Bitcoin provavelmente pensaria que esses debates acalorados parecem muito ruins e até tóxicos. Existem pelo menos dois fatores que fazem parecerem ruins, aos olhos de algumas pessoas:

- Em comparação com empresas de código fechado, todos os debates acontecem abertamente, sem edição. Uma empresa de software como o Google nunca permitiria que seus funcionários debatessem publicamente recursos propostos; de fato, no máximo publicaria uma declaração sobre a posição da empresa sobre o assunto. Isso faz com que as empresas pareçam mais harmônicas em comparação com o Bitcoin.

- Como o Bitcoin é permissivo, qualquer pessoa pode expressar suas opiniões. Isso é fundamentalmente diferente de uma empresa de código fechado, que tem um punhado de pessoas com opinião, geralmente pessoas de mentalidade semelhante. A quantidade de opiniões expressas dentro do Bitcoin é simplesmente impressionante em comparação, por exemplo, com o PayPal.

A maioria dos desenvolvedores de Bitcoin argumentaria que essa abertura traz um ambiente bom e saudável e até que é necessária para produzir o melhor resultado.



Como mencionado em [\[threats\]](#), o segundo ponto acima pode ser muito benéfico, mas vem com uma desvantagem. Um atacante pode usar táticas de procrastinação, como as descritas no [Manual de Sabotagem Simples](#), para distorcer o processo de tomada de decisão e desenvolvimento.



Outra coisa que vale a pena mencionar é que, como observado em [Section 6.4](#), como o Bitcoin é dinheiro e o Bitcoin Core protege quantidades incomensuráveis de dinheiro, a segurança neste contexto não é tratada de maneira leviana. Isso explica por que desenvolvedores experientes do Bitcoin Core podem parecer muito obstinados, uma atitude que é geralmente justificada. De fato, um recurso com uma justificativa fraca não será aceito. O mesmo aconteceria se ele quebrasse os builds reproduzíveis (descritos em [Section 2.1](#)), adicionasse novas dependências, ou se o código não seguisse as [melhores práticas do Bitcoin](#).

Desenvolvedores novos (e antigos) podem ficar frustrados com isso. Mas, como é costume no software de código aberto, você sempre pode fazer um fork do repositório, mesclar o que quiser em seu próprio fork, e construir e executar seu próprio binário.

6.8. Conclusão

O Bitcoin Core e a maioria dos outros softwares de Bitcoin são de código aberto, o que significa que qualquer pessoa pode distribuir, modificar e usar o software como quiser. O repositório do Bitcoin Core no GitHub é atualmente o ponto focal do desenvolvimento do Bitcoin, mas esse status pode mudar se as pessoas começarem a desconfiar de seus mantenedores, ou do próprio site.

O código aberto permite o desenvolvimento permissivo no, e sobre o Bitcoin. Se você escreve código, revisa código ou protocolos; o código aberto é o que permite que você faça isso, de forma pseudônima ou não.

O processo de desenvolvimento em torno do Bitcoin é radicalmente aberto, o que pode fazer o Bitcoin parecer um lugar tóxico e ineficiente, mas é isso que mantém o Bitcoin resiliente contra atores mal-intencionados.

Chapter 7. Escalabilidade



Neste capítulo, exploramos como o Bitcoin escala e como não escala. Começamos analisando como as pessoas raciocinaram sobre escalabilidade no passado. Em seguida, a maior parte deste capítulo explica várias abordagens para escalar o Bitcoin, especificamente escalabilidade vertical, horizontal, interna e em camadas. Cada descrição é seguida por considerações sobre se a abordagem interfere ou não na proposta de valor do Bitcoin.

No espaço do Bitcoin, diferentes pessoas atribuem diferentes definições à palavra “escalar”. Alguns a concebem como o aumento da capacidade de transações da blockchain, outros acreditam que equivale a usar a blockchain de maneira mais eficiente, e outros veem isso como o desenvolvimento de sistemas em cima do Bitcoin.

No contexto do Bitcoin, e para os propósitos deste livro, definimos escalabilidade como *aumentar a capacidade de uso do Bitcoin sem comprometer sua resistência à censura*. Essa definição abrange vários tipos de mudanças, por exemplo:

- Fazer com que as entradas de transações usem menos bytes
- Melhorar o desempenho da verificação de assinaturas
- Reduzir o uso de largura de banda da rede peer-to-peer
- Agrupamento de transações
- Arquitetura em camadas

Em breve, mergulharemos em diferentes abordagens para escalabilidade, mas vamos começar com uma breve visão geral da história do Bitcoin no contexto da escalabilidade.

7.1. História



A escalabilidade tem sido um ponto focal de discussão desde a gênese do Bitcoin. A primeira frase do [primeiro e-mail](#) em resposta ao anúncio de Satoshi sobre o whitepaper do Bitcoin na lista de discussão Cryptography foi justamente sobre escalabilidade:

Satoshi Nakamoto escreveu:

- > Eu tenho trabalhado em um novo sistema de dinheiro eletrônico que é totalmente
- > peer-to-peer, sem uma terceira parte confiável.
- >
- > O paper está disponível em:
- > <http://www.bitcoin.org/bitcoin.pdf>

Nós realmente precisamos de um sistema assim, mas da maneira como entendendo sua proposta, não parece escalar para o tamanho necessário.

— James A. Donald e Satoshi Nakamoto, Cryptography mailing list (2008)

A conversa em si pode não ser muito interessante nem precisa, mas mostra que a escalabilidade tem sido uma preocupação desde o início.

As discussões sobre escalabilidade atingiram seu pico de interesse entre 2015-2017, quando havia muitas ideias diferentes circulando sobre se e como aumentar o limite de tamanho máximo de bloco. Essa foi uma discussão bastante desinteressante sobre a alteração de um parâmetro no código-fonte, uma mudança que não resolveu fundamentalmente nada, mas empurrou o problema da escalabilidade para o futuro, acumulando dívida técnica.



Em 2015, uma conferência chamada [Scaling Bitcoin](#) foi realizada em Montreal, com uma conferência de acompanhamento seis meses depois em Hong Kong e, posteriormente, em vários outros locais ao redor do mundo. O foco era precisamente como abordar a escalabilidade. Muitos desenvolvedores de Bitcoin e outros entusiastas se reuniram nessas conferências para discutir várias questões e propostas de escalabilidade. A maioria dessas discussões não girava em torno de aumentos de tamanho de bloco, mas em soluções mais de longo prazo.



Após a conferência de Hong Kong em dezembro de 2015, Gregory Maxwell [resumiu sua visão](#) sobre muitos dos problemas que haviam sido debatidos, começando com uma filosofia geral sobre escalabilidade.

Com a tecnologia disponível, existem trade-offs fundamentais entre escalabilidade e descentralização. Se o sistema for muito caro, as pessoas serão forçadas a confiar em terceiros em vez de aplicar independentemente as regras do sistema. Se o uso de recursos da blockchain do Bitcoin, em relação à tecnologia disponível, for muito grande, o Bitcoin perderá suas vantagens competitivas em relação aos sistemas legados porque a validação será muito cara (excluindo muitos usuários), forçando a confiança de volta ao sistema. Se a capacidade for muito baixa e nossos métodos de transação forem muito ineficientes, o acesso à cadeia para resolução de disputas será muito caro, novamente empurrando a confiança de volta ao sistema.

— Gregory Maxwell, Aumentos de capacidade para o sistema Bitcoin (2015)

Ele fala sobre o trade-off entre throughput e descentralização. Se você permitir blocos maiores, algumas pessoas serão empurradas para fora da rede porque não terão os recursos para validar os blocos. Por outro lado, se o acesso ao espaço de blocos se tornar mais caro, menos pessoas poderão pagar para usá-lo como um mecanismo de resolução de disputas. Em ambos os casos, os usuários são empurrados para serviços confiáveis.

Ele continua resumindo as várias abordagens para escalabilidade apresentadas na conferência. Entre elas estão verificações de assinaturas mais eficientes computacionalmente, *testemunha segregada* incluindo uma mudança no limite de tamanho de bloco, um mecanismo de propagação de blocos mais eficiente em termos de espaço e a construção de protocolos em camadas sobre o Bitcoin. Muitas dessas abordagens já foram implementadas.

7.2. Abordagens de escalabilidade

Como mencionado anteriormente, a escalabilidade do Bitcoin não precisa necessariamente estar relacionada ao aumento do limite de tamanho de bloco ou outros limites. Agora, vamos passar por algumas abordagens gerais para escalabilidade, algumas das quais não sofrem do trade-off entre throughput e descentralização mencionado na seção anterior.

7.2.1. Escalabilidade vertical

Escalabilidade vertical é o processo de aumentar os recursos de computação das máquinas que processam os dados. No contexto do Bitcoin, estas seriam os full nodes, ou seja, as máquinas que validam a blockchain em nome de seus usuários.

A técnica mais comumente discutida para escalabilidade vertical no Bitcoin é o aumento no limite de tamanho de bloco. Isso exigiria que alguns full nodes atualizassem seu hardware para acompanhar as crescentes demandas computacionais. A desvantagem é que isso acontece às custas da centralização, como discutido na seção anterior e mais detalhadamente em [Section 1.2](#).

Além dos efeitos negativos na descentralização dos full nodes, a escalabilidade vertical também pode impactar negativamente a descentralização da mineração do Bitcoin (explicado em [Section 1.1](#)) e a segurança de maneiras menos óbvias. Vamos ver como os mineradores “deveriam” operar.

Suponha que um minerador mine um bloco na altura 7 e publique esse bloco na rede Bitcoin. Levará algum tempo para que esse bloco seja amplamente aceito, o que se deve principalmente a dois fatores:

- A transferência do bloco entre os pares leva tempo devido a limitações de largura de banda.
- A validação do bloco leva tempo.

Enquanto o bloco 7 está sendo propagado pela rede, muitos mineradores ainda estão minerando em cima do bloco 6 porque ainda não receberam e validaram o bloco 7. Durante esse tempo, se algum desses mineradores encontrar um novo bloco na altura 7, haverá dois blocos concorrentes nessa altura. Só pode haver um bloco na altura 7 (ou em qualquer outra altura), o que significa que um dos dois candidatos deve se tornar obsoleto.

Em resumo, blocos órfãos acontecem porque leva tempo para cada bloco se propagar, e quanto mais tempo leva a propagação, maior a probabilidade de blocos obsoletos.

Suponha que o limite de tamanho de bloco seja removido e que o tamanho médio dos blocos aumente substancialmente. Os blocos se propagariam mais lentamente pela rede devido às limitações de largura de banda e ao tempo de verificação. Um aumento no tempo de propagação também aumentará as chances de blocos obsoletos.

Os mineradores não gostam de ter seus blocos órfãos porque perdem sua recompensa de bloco, então farão o que puderem para evitar esse cenário. As medidas que podem tomar incluem:

- Adiar a validação de um bloco recebido, também conhecido como *mineração sem validação*, discutido mais detalhadamente em [Section 8.2.4.4](#). Os mineradores podem apenas verificar o proof-of-work do cabeçalho do bloco e minerar em cima dele, enquanto baixam o bloco completo e o validam.
- Conectar-se a um pool de mineração com maior largura de banda e conectividade.

A mineração sem validação prejudica ainda mais a descentralização dos full nodes, pois o minerador acaba confiando nos blocos recebidos, pelo menos temporariamente. Também prejudica a segurança em certo grau, porque uma parte do poder computacional da rede está potencialmente construindo em uma blockchain inválida, em vez de construir na cadeia mais forte e válida.

O segundo ponto tem um efeito negativo na descentralização da mineração, veja [Section 1.1](#), porque geralmente os pools com melhor conectividade e largura de banda são também os maiores, fazendo com que os mineradores gravitem em direção a alguns poucos grandes pools.

7.2.2. Escalabilidade horizontal

A escalabilidade horizontal refere-se a técnicas que dividem a carga de trabalho entre várias máquinas. Embora essa seja uma abordagem de escalabilidade predominante entre sites e bancos de dados populares, não é facilmente aplicada ao Bitcoin.



Muitas pessoas se referem a essa abordagem de escalabilidade do Bitcoin como *sharding*.

Basicamente, consiste em permitir que cada full node verifique apenas uma parte da blockchain. Peter Todd pensou muito sobre o conceito de sharding. Ele escreveu um [post no blog](#) explicando o sharding em termos gerais e também apresentando sua própria ideia chamada *treechains*. O artigo é uma leitura difícil, mas Todd faz alguns pontos que são bastante digeríveis.

Em sistemas sharded, a "defesa do full node" não funciona, pelo menos diretamente. O objetivo é que nem todos tenham todos os dados, então você tem que decidir o que acontece quando eles não estão disponíveis.

— Peter Todd, *Why Scaling Bitcoin With Sharding Is Very Hard* (2015)

Em seguida, ele apresenta várias ideias sobre como lidar com o sharding ou escalabilidade horizontal. Perto do final do post, ele conclui:

Há um grande problema, no entanto: !@#\$, o acima é complexo em comparação ao Bitcoin! Mesmo a versão "infantil" de sharding - meu esquema de linearização em vez de zk-SNARKS - é provavelmente uma ou duas ordens de magnitude mais complexa do que usar o protocolo Bitcoin é agora, ainda assim agora uma grande % das empresas neste espaço parece ter jogado a toalha e usado provedores de API centralizados. Implementar o acima e colocá-lo nas mãos dos usuários finais não será fácil.

Por outro lado, a descentralização não é barata: usar o PayPal é uma ou duas ordens de magnitude mais simples do que o protocolo Bitcoin.

— Peter Todd, *Why Scaling Bitcoin With Sharding Is Very Hard* (2015)

A conclusão que ele tira é que o sharding *pode* ser tecnicamente possível, mas viria ao custo de uma complexidade tremenda. Dado que muitos usuários já acham o Bitcoin muito complexo e preferem usar serviços centralizados, será difícil convencê-los a usar algo ainda mais complexo.

7.2.3. Escalabilidade interna


Enquanto a escalabilidade horizontal e vertical historicamente funcionaram bem em sistemas centralizados como bancos de dados e servidores web, eles não parecem ser adequados para uma rede descentralizada como o Bitcoin devido aos seus efeitos centralizadores.

Uma abordagem que recebe pouca apreciação é o que podemos chamar de *escalabilidade interna*, que se traduz em “fazer mais com menos”. Refere-se ao trabalho contínuo constantemente realizado por muitos desenvolvedores para otimizar os algoritmos já em vigor, para que possamos fazer mais dentro dos limites existentes do sistema.



As melhorias alcançadas por meio da escalabilidade interna são impressionantes, para dizer o mínimo. Para dar uma ideia geral das melhorias ao longo dos anos, Jameson Lopp [realizou testes de](#)

[benchmark](#) na sincronização da blockchain, comparando muitas versões diferentes do Bitcoin Core desde a versão 0.8.

Desempenho de download inicial de blocos de várias versões do Bitcoin Core. No eixo Y está a altura do bloco sincronizado e no eixo X está o tempo que levou para sincronizar até essa altura.


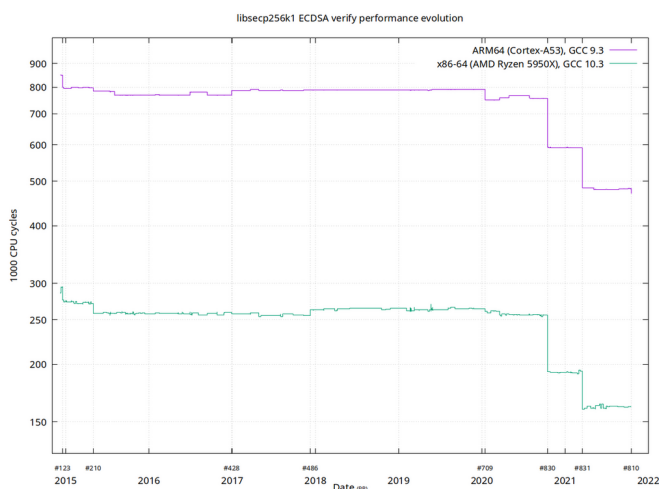
As diferentes linhas representam diferentes versões do Bitcoin Core. A linha mais à esquerda é a mais recente, ou seja, a versão 0.22, que foi lançada em setembro de 2021 e levou 396 minutos para sincronizar completamente. A mais à direita é a versão 0.8 de novembro de 2013, que levou 3452 minutos. Todo esse aprimoramento - aproximadamente 10x - se deve à escalabilidade interna.

As melhorias poderiam ser categorizadas como economia de espaço (RAM, disco, largura de banda, etc.) ou economia de poder computacional. Ambas as categorias contribuem para as melhorias no diagrama acima.



Um bom exemplo de melhoria computacional pode ser encontrado na biblioteca [libsecp256k1](#), que, entre outras coisas, implementa os primitivos criptográficos necessários para criar e verificar assinaturas digitais. Pieter Wuille é um dos contribuidores dessa biblioteca e escreveu uma [thread no Twitter](#) mostrando as melhorias de desempenho alcançadas por meio de várias pull requests.

Desempenho da verificação de assinaturas ao longo do tempo, com pull requests significativas marcadas na linha do tempo.



O gráfico mostra a tendência para dois tipos diferentes de CPU de 64 bits, a saber, ARM e x86. A diferença de desempenho se deve às instruções mais especializadas disponíveis no x86 em comparação à arquitetura ARM, que tem menos e mais instruções genéricas. No entanto, a tendência geral é a mesma para ambas as arquiteturas. Observe que o eixo Y é logarítmico, o que faz com que as melhorias pareçam menos impressionantes do que realmente são.



Também existem vários bons exemplos de melhorias na economia de espaço que contribuíram para o aprimoramento do desempenho. Em um [post no blog Medium](#) sobre a contribuição do Taproot para a economia de espaço, o usuário Murch compara quanto espaço de bloco uma assinatura de limite 2-de-3 exigiria, usando Taproot de várias maneiras, bem como não usando.

Input	Native Segwit	Taproot (P2TR)		
		key path	script path	
Single-sig	68.5 vB P2WPKH	57.5 vB	–	–
2-of-3	104.5 vB P2WSH		82.75 vB MuSig leaf	107.5 vB 2-of-2 leaf

Output	Native Segwit	Taproot (P2TR)
Single-sig	31 B P2WPKH	43 B
2-of-3	43 B P2WSH	

created by @murchandamus

Figure 6. Economia de espaço para diferentes tipos de gasto, versões Taproot e legadas.

Uma multisig 2-de-3 usando Segwit nativo exigiria um total de $104,5 + 43 \text{ vB} = 147,5 \text{ vB}$, enquanto o uso mais econômico de espaço do Taproot exigiria apenas $57,5 + 43 \text{ vB} = 100,5 \text{ vB}$ no caso de uso padrão. Na pior das hipóteses e em casos raros, como quando um signatário padrão não está disponível por algum motivo, o Taproot usaria $107,5 + 43 \text{ vB} = 150,5 \text{ vB}$. Você não precisa entender todos os detalhes, mas isso deve lhe dar uma ideia de como os desenvolvedores pensam em economizar espaço - cada pequeno byte conta.

Além da escalabilidade interna no software do Bitcoin, há algumas maneiras pelas quais os usuários também podem contribuir para a escalabilidade interna. Eles podem fazer suas transações de forma mais inteligente para economizar nas taxas de transação, ao mesmo tempo em que reduzem suas pegadas nos requisitos dos full nodes. Duas técnicas comumente usadas para esse fim são chamadas de agrupamento de transações e consolidação de saídas.

A ideia do agrupamento de transações é combinar vários pagamentos em uma única transação, em vez de fazer uma transação por pagamento. Isso pode economizar muitas taxas para você e, ao mesmo tempo, reduzir a carga no espaço de bloco.

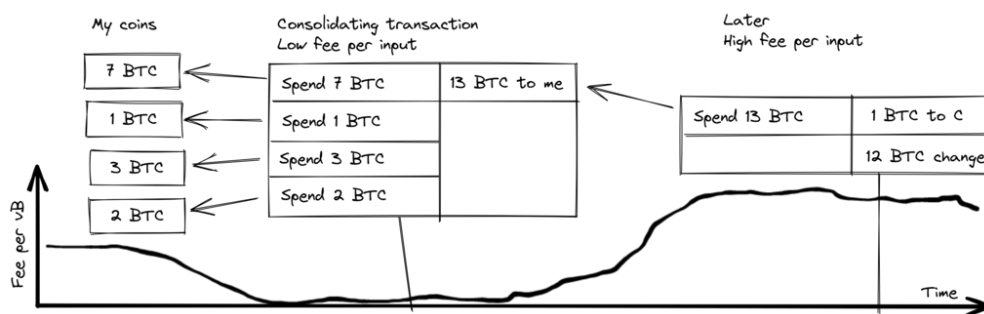
1 tx per payment More data, expensive		batched payments Less data, cheaper	
Spend 3 BTC	5 BTC to A	Spend 4 BTC	5 BTC to A
Spend 3 BTC	1 BTC change	Spend 12 BTC	5 BTC to B
Spend 4 BTC	5 BTC to B		5 BTC to C
Spend 2 BTC	1 BTC change		1 BTC change
Spend 4 BTC	5 BTC to C		
Spend 2 BTC	1 BTC change		

Figure 7. Agrupamento de transações combina vários pagamentos em uma única transação para economizar em taxas.

A consolidação de saídas refere-se a aproveitar os períodos de baixa demanda por espaço de bloco

para combinar várias saídas em uma única saída. Isso pode reduzir seu custo de taxa mais tarde, quando você precisar fazer um pagamento enquanto a demanda por espaço de bloco estiver alta.

Consolidação de saídas. Derreta suas moedas em uma grande moeda quando as taxas estiverem baixas para economizar taxas mais tarde.



Pode não ser óbvio como a consolidação de saídas contribui para a escalabilidade interna. Afinal, a quantidade total de dados da blockchain é até ligeiramente aumentada com este método. No entanto, o conjunto UTXO, ou seja, o banco de dados que acompanha quem possui quais moedas, encolhe porque você gasta mais UTXOs do que cria. Isso alivia o fardo para os full nodes manterem seus conjuntos UTXO.

Infelizmente, no entanto, essas duas técnicas de *gerenciamento de UTXO* podem ser prejudiciais para sua própria privacidade ou para a de seus destinatários. No caso do agrupamento, cada destinatário saberá que todas as saídas agrupadas são de você para outros destinatários (exceto possivelmente o troco). No caso da consolidação de UTXO, você revelará que as saídas que você consolida pertencem à mesma carteira. Portanto, você pode ter que fazer um trade-off entre eficiência de custos e privacidade.

7.2.4. Escalabilidade em camadas

A abordagem mais impactante para escalabilidade é provavelmente a em camadas. A ideia geral por trás das camadas é que um protocolo pode liquidar pagamentos entre usuários sem adicionar transações à blockchain. Isso já foi discutido brevemente em [Chapter 2](#) e [Section 3.7](#).

Um protocolo em camadas começa com duas ou mais pessoas concordando em uma transação inicial que é colocada na blockchain, como ilustrado em [Figure 8](#).

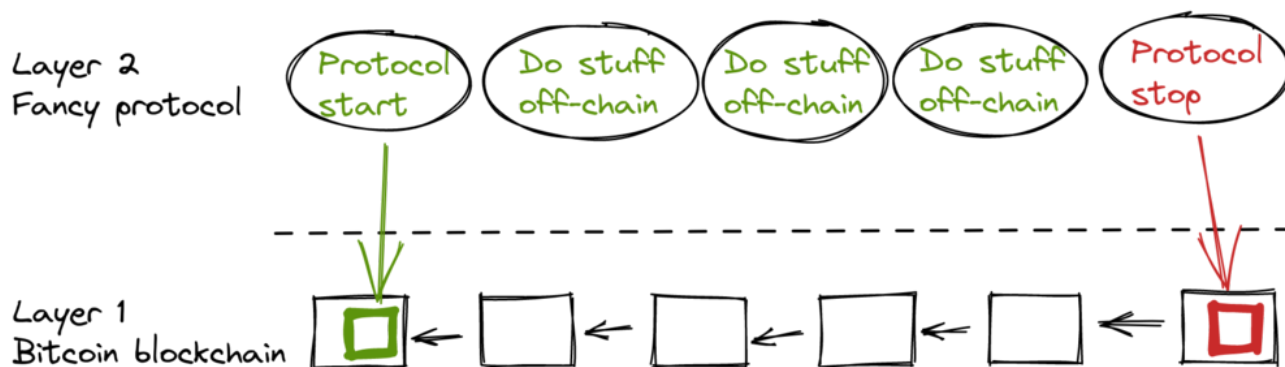


Figure 8. Um protocolo típico de camada 2 sobre o Bitcoin, camada 1.

Como essa transação inicial é criada varia entre os protocolos, mas um tema comum é que os

participantes criam uma transação inicial não assinada e um número de transações de punição pré-assinadas, que gastam a saída da transação inicial de várias maneiras. Posteriormente, a transação inicial é totalmente assinada e publicada na blockchain, e as transações de punição podem ser totalmente assinadas e publicadas para punir uma parte malcomportada. Isso incentiva os participantes a manter suas promessas para que o protocolo possa funcionar de maneira *trustless*.

Uma vez que a transação inicial esteja na blockchain, o protocolo pode fazer o que se propôs a fazer. Por exemplo, pode realizar pagamentos super rápidos entre os participantes, implementar algumas técnicas de melhoria de privacidade ou realizar scripts mais avançados que não seriam suportados pela blockchain do Bitcoin.

Não detalharemos como funcionam protocolos específicos, mas, como você pode ver em [Figure 8](#), a blockchain é raramente usada durante o ciclo de vida do protocolo. Toda a ação acontece *fora da cadeia*. Vimos como isso pode ser uma vitória para a privacidade se feito corretamente, mas também pode ser uma vantagem para a escalabilidade.



Em um [post no Reddit](#) intitulado " `Uma viagem à lua requer um foguete com várias etapas ou caso contrário a equação do foguete vai acabar com você... colocar todos em um estilo de carro palhaço em uma catapulta e esperar pelo sucesso está fora de questão. `", Gregory Maxwell explica por que a estratificação é nossa melhor chance de fazer o Bitcoin escalar em ordens de magnitude.

Ele começa enfatizando a falácia de ver Visa ou Mastercard como os principais concorrentes do Bitcoin e destacando como aumentar o tamanho máximo de bloco é uma abordagem ruim para atender a essa competição. Em seguida, ele fala sobre como fazer uma diferença real usando camadas.

Então-- Isso significa que o Bitcoin não pode ser um grande vencedor como tecnologia de pagamentos? Não. Mas para atingir o tipo de capacidade necessária para atender às necessidades de pagamento do mundo, precisamos trabalhar de maneira mais inteligente.

Desde o seu início, o Bitcoin foi projetado para incorporar camadas de maneira segura através de sua capacidade de contratos inteligentes (O que, você acha que isso foi colocado lá apenas para as pessoas filosofarem sobre "DAOs" sem sentido?). Na prática, usaremos o sistema Bitcoin como um juiz robótico altamente acessível e perfeitamente confiável e conduziremos a maior parte de nossos negócios fora da sala de audiência-- mas transacionaremos de forma que, se algo der errado, tenhamos todas as provas e acordos estabelecidos para que possamos ter confiança de que o tribunal robótico corrigirá a situação. (Geek sidebar: Se isso parecer impossível, vá ler este antigo post sobre transação cut-through)

Isso é possível precisamente por causa das propriedades centrais do Bitcoin. Um sistema base censurável ou reversível não é muito adequado para construir um processamento de transações de camada superior em cima... e se o ativo subjacente não for sólido, não há muito sentido em transacionar com ele.

— Gregory Maxwell, r/Bitcoin no Reddit (2016)

A analogia com o juiz é bastante ilustrativa de como a estratificação funciona: esse juiz deve ser incorruptível e nunca mudar de ideia, caso contrário, as camadas acima da camada base do Bitcoin não funcionarão de forma confiável.

Ele continua destacando um ponto sobre serviços centralizados. Geralmente, não há problema em confiar em um servidor central com quantias triviais de Bitcoin para fazer as coisas acontecerem: isso também é escalabilidade em camadas.

Muitos anos se passaram desde que Maxwell escreveu o trecho acima, e suas palavras ainda estão corretas. O sucesso da Lightning Network prova que a estratificação é realmente um caminho a seguir para aumentar a utilidade do Bitcoin.

7.3. Conclusão

Discutimos várias maneiras pelas quais se pode querer escalar o Bitcoin, aumentar a capacidade de uso do Bitcoin. A escalabilidade tem sido uma preocupação no Bitcoin desde seus primeiros dias.

Sabemos hoje que o Bitcoin não escala bem verticalmente (“compre hardware maior”) ou horizontalmente (“verifique apenas partes dos dados”), mas sim internamente (“faça mais com menos”) e em camadas (“construa protocolos em cima do Bitcoin”).

Chapter 8. Quando dá tudo errado



O Bitcoin é construído por pessoas. Pessoas escrevem o software e, em seguida, essas mesmas pessoas executam o software. Quando uma vulnerabilidade de segurança ou um bug grave é descoberto – existe realmente uma distinção entre os dois? – essa descoberta é sempre feita por pessoas, de carne e osso. Este capítulo contempla o que as pessoas fazem, o que deveriam fazer e o que não deveriam fazer quando tudo dá errado. A primeira seção explica o termo *divulgação responsável*, que se refere a como alguém que descobre uma vulnerabilidade pode agir de forma responsável para ajudar a minimizar os danos. O restante do capítulo leva você a uma jornada por algumas das vulnerabilidades mais graves descobertas ao longo dos anos e como foram tratadas por desenvolvedores, mineradores e usuários. As coisas não eram tão rigorosas na infância do Bitcoin como são hoje.

8.1. Divulgação responsável

Imagine que você descobre um bug no Bitcoin Core, um bug que permite a qualquer pessoa desligar remotamente um nó Bitcoin Core usando algumas mensagens de rede especialmente criadas. Imagine também que você não é mal-intencionado e gostaria que esse problema permanecesse sem ser explorado. O que você faz? Se você ficar em silêncio sobre isso, provavelmente outra pessoa descobrirá o problema, e você não pode ter certeza de que essa pessoa não será mal-intencionada.



Quando um problema de segurança é descoberto, a pessoa que o descobre deve empregar *divulgação responsável*, um termo frequentemente usado entre desenvolvedores de Bitcoin. O termo é [explicado na Wikipedia](#):

Desenvolvedores de hardware e software frequentemente precisam de tempo e recursos para corrigir seus erros. Muitas vezes, são hackers éticos que encontram essas vulnerabilidades. Hackers e cientistas de segurança de computadores têm a opinião de que é sua responsabilidade social informar o público sobre vulnerabilidades. Esconder problemas pode causar uma sensação de falsa segurança. Para evitar isso, as partes envolvidas coordenam e negociam um período razoável de tempo para reparar a vulnerabilidade. Dependendo do impacto potencial da vulnerabilidade, do tempo necessário para que uma correção de emergência ou solução alternativa seja desenvolvida e aplicada, e de outros fatores, esse período pode variar entre alguns dias e vários meses.

— Wikipedia, artigo sobre Divulgação responsável



Isso significa que, se você encontrar um problema de segurança, deve relatar isso à equipe responsável pelo sistema. Mas o que isso significa no contexto do Bitcoin? Como observado em [Section 6.1](#), ninguém controla o Bitcoin, mas atualmente existe um ponto focal para o desenvolvimento do Bitcoin, nomeadamente o [repositório Bitcoin Core no Github](#). Os mantenedores deste repositório são responsáveis pelo código nele contido, mas não são responsáveis pelo sistema como um todo – ninguém é. No entanto, a prática geral recomendada é enviar um e-mail para security@bitcoincore.org.



Em uma [troca de e-mails](#) intitulada “Responsible disclosure of bugs” de 2017, Anthony Towns tentou resumir o que ele percebia como as melhores práticas atuais. Ele coletou contribuições de várias fontes e diferentes pessoas para formar sua visão sobre o assunto.

- Vulnerabilidades devem ser relatadas via security at bitcoincore.org [0]
- Um problema crítico (que pode ser explorado imediatamente ou já está sendo explorado causando grandes danos) será tratado por:
 - uma correção lançada o mais rápido possível
 - notificação ampla da necessidade de atualizar (ou desativar sistemas afetados)
 - divulgação mínima do problema real, para atrasar ataques [1] [2]
- Uma vulnerabilidade não crítica (porque é difícil ou cara de explorar) será tratada por:
 - correção e revisão realizadas no fluxo normal de desenvolvimento
 - backport de uma correção ou solução alternativa do master para a versão atual lançada [2]
- Os desenvolvedores tentarão garantir que a publicação da correção não revele a natureza da vulnerabilidade, fornecendo a correção proposta para desenvolvedores experientes que não foram informados da vulnerabilidade, dizendo-lhes que ela corrige uma vulnerabilidade e pedindo-lhes para identificar a vulnerabilidade. [2]
- Os desenvolvedores podem recomendar que outras implementações de bitcoin adotem as correções de vulnerabilidade antes que a correção seja lançada e amplamente implementada, se puderem fazer isso sem revelar a vulnerabilidade; por exemplo, se a correção tiver benefícios de desempenho significativos que justifiquem sua inclusão. [3]
- Antes de uma vulnerabilidade se tornar pública, os desenvolvedores geralmente recomendam aos desenvolvedores de altcoins amigáveis que acompanhem as correções. Mas isso só acontece depois que as correções são amplamente implementadas na rede bitcoin. [4]
- Os desenvolvedores geralmente não notificarão desenvolvedores de altcoins que se comportaram de maneira hostil (por exemplo, usando vulnerabilidades para atacar outros, ou que violam embargos). [5]
- Os desenvolvedores do Bitcoin não divulgarão detalhes de vulnerabilidades até que >80% dos nós do bitcoin tenham implementado as correções. Os descobridores de vulnerabilidades são encorajados e solicitados a seguir a mesma política. [1] [6]

Esta lista mostra o quão cuidadoso alguém deve ser ao publicar correções para o Bitcoin, pois a própria correção pode revelar a vulnerabilidade. A quarta bala é particularmente interessante, pois explica como testar se uma correção foi bem disfarçada. De fato, se alguns desenvolvedores realmente experientes não conseguirem identificar a vulnerabilidade, mesmo sabendo que a correção resolve uma, será provavelmente muito difícil para outras pessoas descobri-la.



A discussão que levou a esse e-mail discutia se, quando e como divulgar vulnerabilidades para altcoins e outras implementações do Bitcoin. Não há uma resposta clara aqui. “Ajudar os mocinhos” parece ser a coisa sensata a fazer, mas quem decide quem são eles e onde se traça a linha? Bryan Bishop [argumentou](#) que ajudar altcoins e até mesmo scamcoins a se defenderem contra explorações de segurança era um dever moral.

Não basta defender o bitcoin e seus usuários contra ameaças ativas, existe uma responsabilidade mais geral de defender todos os tipos de usuários e diferentes softwares de muitas ameaças, independentemente da forma que tomem, mesmo que as pessoas estejam usando softwares estúpidos e inseguros que você pessoalmente não mantém, contribui ou defende. Lidar com o conhecimento de uma vulnerabilidade é uma questão delicada e você pode estar recebendo conhecimento com impacto direto ou indireto mais sério do que originalmente descrito.

— Bryan Bishop no tópico “`Responsible disclosure of bugs`”, lista de e-mails Bitcoin-dev (2017)



Outro fator que levou ao e-mail de Towns acima foi um [post](#) de Gregory Maxwell, no qual ele argumentava que as vulnerabilidades de segurança podem ser mais graves do que parecem.

Já vi várias vezes um problema difícil de explorar se revelar trivial quando você encontra a técnica certa, ou um problema menor de DoS se tornar muito mais sério.

Bugs simples de desempenho, implantados de forma experta, podem ser potencialmente usados para dividir a rede—minerador A e exchange B ficam em uma partição, todos os outros em outra... e fazer double-spend (gasto duplo).

E assim por diante. Então, embora eu absolutamente concorde que coisas diferentes devem e podem ser tratadas de maneira diferente, nem sempre é tão claro. É prudente tratar as coisas como mais graves do que você sabe que são.

— Gregory Maxwell no tópico "Responsible disclosure of bugs", lista de e-mails Bitcoin-dev (2017)

Então, mesmo que uma vulnerabilidade pareça difícil de explorar, pode ser melhor presumir que é facilmente explorável e que você apenas ainda não descobriu como.

Ele também menciona como “é um pouco incorreto chamar esta discussão de qualquer coisa sobre divulgação, esta discussão não é sobre divulgação. Divulgação é quando você conta ao fornecedor. Esta discussão é sobre publicação e isso tem implicações muito diferentes. Publicação é quando você tem certeza de que contou aos futuros atacantes”. Essa última observação sobre a distinção entre divulgação e publicação é importante. A parte fácil é a divulgação responsável; a parte difícil é a publicação sensata.

8.2. Infância traumática

O Bitcoin começou como um projeto de uma pessoa (pelo menos é o que o pseudônimo de seu criador sugere), e bitcoin inicialmente tinha pouco ou nenhum valor. Como tal, as vulnerabilidades e correções de bugs não eram tratadas com o mesmo rigor de hoje.



O wiki do Bitcoin tem uma [lista de vulnerabilidades e exposições comuns](#) (CVEs) pelas quais o Bitcoin passou. Esta seção constitui um pequeno exposé de alguns dos problemas de segurança e incidentes dos primeiros anos do Bitcoin. Não vamos cobrir todos eles, mas selecionamos alguns que achamos especialmente interessantes.

8.2.1. 2010-07-28: Gastar moedas de qualquer pessoa (CVE-2010-5141)

Em 28 de julho de 2010, uma pessoa pseudônima chamada ArtForz descobriu um bug na versão 0.3.4 que permitia a qualquer pessoa retirar moedas de qualquer outra pessoa. ArtForz relatou *responsavelmente* isso a Satoshi Nakamoto e a outro desenvolvedor do Bitcoin chamado Gavin Andresen.

O problema era que o operador de script `OP_RETURN` simplesmente encerrava a execução do programa, então se o `scriptPubKey` fosse `<pubkey> OP_CHECKSIG` e `scriptSig` fosse `OP_1 OP_RETURN`, a parte do programa em `scriptPubKey` nunca seria executada. A única coisa que aconteceria seria que `1` seria colocado na pilha e então `OP_RETURN` faria o programa sair. Qualquer valor diferente de zero no topo da pilha após a execução do programa significa que a condição de gasto está cumprida. Como o elemento superior da pilha `1` é diferente de zero, o gasto seria permitido.

Este era o código para o manuseio de `OP_RETURN`:

```
case OP_RETURN:
{
    pc = pend;
}
break;
```

O efeito de `pc = pend;` era que o resto do programa era ignorado, significando que qualquer script de bloqueio em `scriptPubKey` seria ignorado. A correção consistiu em mudar o significado de `OP_RETURN` para que ele falhasse imediatamente, em vez disso.

```
case OP_RETURN:
{
    return false;
}
break;
```

Satoshi fez essa mudança localmente e construiu um binário executável com a versão 0.3.5 a partir dele. Então, ele postou no fórum Bitcointalk “* ALERTA *** Atualize para 0.3.5 O MAIS RÁPIDO POSSÍVEL”, instando os usuários a instalar esta versão binária dele, sem apresentar o código-fonte.

Por favor, atualize para 0.3.5 o mais rápido possível! Corrigimos um bug de implementação onde era possível que transações falsas fossem aceitas. Não aceite transações de Bitcoin como pagamento até que você atualize para a versão 0.3.5!

— Satoshi Nakamoto, fórum Bitcointalk (2010)



A mensagem original foi editada mais tarde e não está mais disponível em sua forma completa. O trecho acima é de uma [resposta com citação](#). Alguns usuários tentaram o binário de Satoshi, mas tiveram problemas com ele. Logo depois, [Satoshi escreveu](#):

Ainda não tive tempo de atualizar o SVN. Espere pela versão 0.3.6, estou construindo agora. Você pode desligar seu nó enquanto isso.

— Satoshi Nakamoto, fórum Bitcointalk (2010)



E 35 minutos depois, [ele escreveu](#)

O SVN foi atualizado com a versão 0.3.6.

Carregando a versão para Windows da 0.3.6 no Sourceforge agora, depois irei recompilar para Linux.

— Satoshi Nakamoto, fórum Bitcointalk (2010)

Nesse ponto, ele também parece ter atualizado a postagem original para mencionar 0.3.6 em vez de 0.3.5:

Por favor, atualize para 0.3.6 o mais rápido possível! Corrigimos um bug de implementação onde era possível que transações falsas fossem exibidas como aceitas. Não aceite transações de Bitcoin como pagamento até que você atualize para a versão 0.3.6!

Se você não puder atualizar para 0.3.6 imediatamente, é melhor desligar seu nó Bitcoin até que você o faça.

Também na 0.3.6, hashing mais rápido:

- otimização do cache de midstate graças a tcatm
 - SHA-256 ASM do Crypto++ graças a BlackEye
- Aceleração total na geração, 2,4x mais rápido.

Download:

<http://sourceforge.net/projects/bitcoin/files/Bitcoin/bitcoin-0.3.6/>

Usuários de Windows e Linux: se você obteve a 0.3.5, ainda precisará atualizar para a 0.3.6.

— Satoshi Nakamoto, fórum Bitcointalk (2010)

Observe a diferença na caracterização do problema em relação à primeira mensagem: “poderia ser exibida como aceita” vs “poderia ser aceita”. Talvez Satoshi tenha minimizado a gravidade do bug em sua comunicação para não chamar muita atenção para o problema real. De qualquer forma, as pessoas atualizaram para a 0.3.6 e funcionou como esperado. Esta questão particular foi resolvida, incrivelmente, sem perdas de bitcoin.

A mensagem de Satoshi também descreveu algumas otimizações de desempenho para mineração. Não está claro por que isso foi incluído em uma correção de segurança crítica, é possível que o objetivo fosse ofuscar o problema real. No entanto, parece mais provável que ele tenha simplesmente lançado o que estava na cabeça do branch de desenvolvimento do repositório Subversion, com a correção de segurança adicionada a ele.

Naquela época, não havia nem de perto tantos usuários quanto hoje, e o valor do bitcoin era próximo de zero. Se essa resposta ao bug fosse executada hoje, seria considerada um verdadeiro show de horror por vários motivos:



- Satoshi fez uma versão binária única da 0.3.5 contendo a correção. Nenhum patch ou código foi fornecido, talvez como uma medida para ofuscar o problema.
- 0.3.5 [nem funcionou](#).
- A correção na 0.3.6 foi, na verdade, um hard fork, como explicado em [Section 5.2](#).

Outra coisa debatível é se é bom ou ruim que os usuários tenham sido solicitados a desligar seus nós. Isso não seria viável hoje, mas naquela época muitos usuários acompanhavam ativamente os fóruns para atualizações e geralmente estavam cientes das coisas. Dado que era possível fazer isso, pode ter sido algo sensato a se fazer.

8.2.2. 2010-08-15 Transbordo de saída combinada (CVE-2010-5139)



Em meados de agosto de 2010, o usuário do fórum Bitcointalk jgarzik, também conhecido como Jeff Garzik, [descobriu que](#) uma certa transação no bloco de altura 74638 tinha duas saídas de valor excepcionalmente alto:

O "valor de saída" neste bloco #74638 é bastante estranho:

```
...
  "out" : [
    {
      "value" : 92233720368.54277039,
      "scriptPubKey" : "OP_DUP OP_HASH160
0xB7A73EB128D7EA3D388DB12418302A1CBAD5E890 OP_EQUALVERIFY OP_CHECKSIG"
    },
    {
      "value" : 92233720368.54277039,
      "scriptPubKey" : "OP_DUP OP_HASH160
0x151275508C66F89DEC2C5F43B6F9CBE0B5C4722C OP_EQUALVERIFY OP_CHECKSIG"
    }
  ]
...
```

92233720368.54277039 BTC? Isso é `UINT64_MAX`, eu me pergunto?

— Jeff Garzik, fórum Bitcointalk (2010)

Presumivelmente, havia um bug que fazia com que a soma de duas saídas `int64` (não `uint64`, como Garzik supôs) transbordasse para um valor negativo -0,00997538 BTC. Qualquer que fosse a soma das entradas, a “soma” das saídas seria menor, tornando esta transação válida de acordo com o código da época.

Nesse caso, o bug foi divulgado e publicado através de uma exploração real. Um resultado infeliz disso foi que cerca de 2x92 bilhões de bitcoins foram criados, o que diluiu severamente a oferta monetária de cerca de 3,7 milhões de moedas que existiam naquela época.



Em um tópico relacionado, [Satoshi postou](#) que apreciaria se as pessoas parassem de minerar (ou *gerar*, como era chamado na época).

Seria útil se as pessoas parassem de gerar. Provavelmente precisaremos refazer um branch em torno do atual, e quanto menos você gerar, mais rápido isso acontecerá.

Um primeiro patch estará na revisão 132 do SVN. Ainda não está carregado. Estou empurrando algumas outras mudanças diversas para fora do caminho primeiro, depois carregarei o patch para isso.

— Satoshi Nakamoto, fórum Bitcointalk (2010)



O plano dele era fazer um soft fork para invalidar transações como a discutida aqui, invalidando assim os blocos (especialmente o bloco 74638) que continham essas transações. Menos de uma hora depois, ele realizou um [commit de patch na revisão 132](#) do repositório Subversion e [postou no fórum](#) descrevendo o que achava que os usuários deveriam fazer:

O patch foi carregado para a revisão 132 do SVN!

Por enquanto, os passos recomendados:

- 1) Desligue o node.
- 2) Baixe os arquivos blk do knightmb. (substitua seus arquivos blk0001.dat e blkindex.dat)
- 3) Faça a atualização.
- 4) Ele deve começar com menos de 74000 blocos. Deixe-o baixar o resto novamente.

Se você não quiser usar os arquivos do knightmb, você pode simplesmente deletar seus arquivos blk*.dat, mas vai sobrecarregar a rede se todo mundo estiver baixando o índice de blocos inteiro ao mesmo tempo.

Construirei as releases em breve.

Ele queria que as pessoas baixassem os dados do blockchain de um usuário específico, chamado knightmb, que publicou seu blockchain como estava em seu disco, nos arquivos blkXXXX.dat e blkindex.dat. O motivo para baixar os dados do blockchain dessa forma, em vez de sincronizar do zero, era reduzir os gargalos de largura de banda da rede.

Havia um grande problema com isso: os dados que os usuários baixariam do knightmb [não eram verificados pelo software Bitcoin](#) na inicialização. O arquivo blkindex.dat continha o conjunto UTXO, e o software aceitaria qualquer dado ali como se já tivesse sido verificado. knightmb poderia ter manipulado os dados para se dar bitcoins ou para outra pessoa.

Novamente, as pessoas pareceram aceitar isso, e a reversão do bloco inválido e seus sucessores foi bem-sucedida. Os mineradores começaram a trabalhar em um novo sucessor para o bloco [74637](#) e, de acordo com o timestamp do bloco, um sucessor apareceu às 23:53 UTC, cerca de 6 horas após o problema ser descoberto. Às 08:10 do dia seguinte, em 16 de agosto, em torno do bloco 74689, a nova cadeia ultrapassou a cadeia antiga, portanto, todos os nodes não atualizados fizeram reorg para seguir a nova cadeia. Este é o reorg mais profundo - 52 blocos - na história do Bitcoin.

Comparado ao problema do OP_RETURN, este foi tratado de uma forma um pouco mais limpa:

- Nenhuma release binária apenas com o patch

- O software lançado funcionou conforme o esperado
- Nenhum hard fork

Os usuários foram convidados a parar de minerar durante este problema também. Podemos discutir se isso é uma boa ideia ou não, mas imagine que você é um minerador e está convencido de que quaisquer blocos acima do bloco ruim eventualmente serão apagados em um reorg profundo: por que desperdiçar recursos minerando blocos condenados?

Você também pode pensar que é um pouco suspeito seguir a sugestão de Nakamoto e baixar o blockchain, incluindo o conjunto UTXO, do disco rígido de um cara aleatório. Se sim, você está certo: isso é suspeito. Mas, dadas as circunstâncias, essa resposta de emergência foi sensata.

Há uma diferença importante entre este caso e o caso anterior do OP_RETURN: este problema foi explorado na prática, e assim uma correção pôde ser feita de maneira mais direta. No caso do OP_RETURN, eles tiveram que ofuscar a correção e fazer declarações públicas que não revelassem diretamente qual era o problema.

8.2.3. 2013-03-11 Problema de bloqueios de banco de dados 0.7.2 - 0.8.0 (CVE-2013-3220)



Uma questão muito interessante e educacionalmente valiosa surgiu em março de 2013. Aparentemente, o blockchain havia se dividido (embora a palavra “fork” seja usada na citação abaixo) após o bloco 225429. Os detalhes deste incidente foram [relatados no BIP50](#). O resumo diz:

Um bloco que tinha um número maior de entradas de transações do que anteriormente visto foi minerado e transmitido. Os nodes com Bitcoin 0.8 conseguiram lidar com isso, mas alguns nodes com versões anteriores ao 0.8 rejeitaram o bloco, causando uma divisão inesperada do blockchain. A cadeia incompatível com versões anteriores ao 0.8 (daqui em diante, a cadeia 0.8) tinha naquela altura cerca de 60% do poder de hash, garantindo que a divisão não fosse resolvida automaticamente (como teria ocorrido se a cadeia anterior ao 0.8 ultrapassasse a cadeia 0.8 em trabalho total, forçando os nodes 0.8 a se reorganizarem para a cadeia anterior ao 0.8).

Para restaurar uma cadeia canônica o mais rápido possível, BTCGuild e Slush rebaixaram seus nodes Bitcoin 0.8 para 0.7 para que seus pools também rejeitassem o bloco maior. Isso colocou a maioria do poder de hash na cadeia sem o bloco maior, eventualmente fazendo com que os nodes 0.8 se reorganizassem para a cadeia anterior ao 0.8.

— Vários desenvolvedores do Bitcoin Core, BIP50 (2013)

A rápida ação que os pools de mineração BTCGuild e Slush tomaram foi imperativa nesta

emergência. Eles conseguiram inclinar a maioria do poder de hash para o branch anterior ao 0.8 da divisão, e assim ajudar a restaurar o consenso. Isso deu aos desenvolvedores tempo para descobrir uma correção sustentável.



Outro ponto muito interessante nesta questão é que a versão 0.7.2 era incompatível consigo mesma, assim como as versões anteriores. Isso é explicado na [seção de causa raiz do BIP50](#):

Com a configuração de bloqueios do BDB insuficientemente alta, ela se tornou implicitamente uma regra de consenso de rede determinando a validade do bloco (embora uma regra inconsistente e insegura, já que o uso de bloqueios poderia variar de node para node).

— Vários desenvolvedores do Bitcoin Core, BIP50 (2013)

Em resumo, o problema é que o número de bloqueios de banco de dados que o software do Bitcoin Core precisa para verificar um bloco não é determinístico. Um node pode precisar de X bloqueios enquanto outro node pode precisar de $X+1$ bloqueios. Os nodes também têm um limite de quantos bloqueios o Bitcoin pode tomar. Se o número de bloqueios necessários exceder o limite, o bloco será considerado inválido. Então, se $X+1$ exceder o limite, mas não X , os dois nodes dividirão o blockchain e discordarão sobre qual branch é válido.

A solução escolhida, além das ações imediatas tomadas pelos dois pools para restaurar o consenso, foi

- limitar os blocos em termos de tamanho e bloqueios necessários na versão 0.8.1
- corrigir versões antigas (0.7.2 e algumas anteriores) com as mesmas novas regras, e aumentar o limite global de bloqueios.

Exceto pelo aumento do limite global de bloqueios no segundo ponto, essas regras foram implementadas temporariamente por um período de tempo predeterminado. O plano era remover esses limites assim que a maioria dos nodes tivesse sido atualizada.

Este soft fork reduziu drasticamente o risco de falha de consenso, e alguns meses depois, em 15 de maio, as regras temporárias foram desativadas em concerto em toda a rede. Note que essa desativação foi, na prática, um hard fork, mas não foi controverso. Além disso, foi lançado junto com o soft fork anterior, então as pessoas que rodavam o software com o soft fork estavam bem cientes de que um hard fork o seguiria. Portanto, a grande maioria dos nodes permaneceu em consenso quando o hard fork foi ativado. Infelizmente, alguns nodes que não atualizaram foram perdidos no processo.

Pode-se questionar se isso seria possível hoje. O cenário de mineração é mais complexo hoje, e, dependendo do poder de hash de cada lado da divisão, pode ser difícil lançar um patch como o do BIP50 rapidamente o suficiente. Provavelmente seria difícil convencer os mineradores no “lado errado” a abrir mão de suas recompensas de blocos.

8.2.4. BIP66

O BIP66 é interessante porque destaca a importância de

- boa criptografia de seleção
- divulgação responsável
- implantação sem revelar a vulnerabilidade
- mineração em cima de blocos verificados



O BIP66 foi uma proposta para endurecer as regras para codificações de assinatura no Bitcoin Script. A [motivação](#) era poder analisar assinaturas com software ou bibliotecas diferentes do OpenSSL e até mesmo versões recentes do OpenSSL. OpenSSL é uma biblioteca de criptografia de uso geral que o Bitcoin Core usava na época.

O BIP foi ativado em 4 de julho de 2015. No entanto, enquanto o acima é verdade, o BIP66 também corrige um problema muito mais grave que não é mencionado no BIP.

A vulnerabilidade



A divulgação completa deste problema foi publicada em 28 de julho de 2015 por Pieter Wuille em um [email para a lista de discussão Bitcoin-dev](#):

Olá a todos,

Gostaria de divulgar uma vulnerabilidade que descobri em setembro de 2014, que se tornou inexplotável quando o limite de 95% do BIP66 foi alcançado no início deste mês.

Descrição curta:

Uma transação especialmente criada poderia ter dividido o blockchain entre nodes:

- usando OpenSSL em sistemas de 32 bits e em sistemas Windows de 64 bits
- usando OpenSSL em sistemas de 64 bits que não são Windows (Linux, OSX, ...)
- usando algumas bases de código não-OpenSSL para análise de assinaturas

— Pieter Wuille na lista de discussão Bitcoin-dev, Divulgação: bug de consenso indiretamente resolvido pelo BIP66 (2015)

O email detalha ainda mais como o problema foi descoberto e exatamente o que o causou. No final, ele apresenta uma linha do tempo dos eventos, e vamos reproduzir alguns dos mais importantes aqui. Alguns deles já foram descritos em [Figure 9](#).

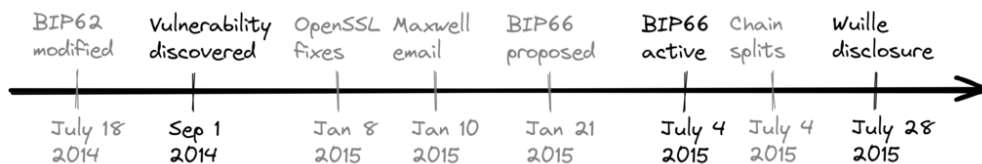


Figure 9. Linha do tempo dos eventos em torno do BIP66. Itens em preto já foram explicados acima.

Antes da descoberta

Sem que ninguém soubesse do problema, ele poderia ter sido resolvido pelo agora retirado BIP62, que era uma proposta para reduzir as possibilidades de maleabilidade de transações. Entre as mudanças propostas no BIP62 estava o endurecimento das regras de consenso para a codificação de assinaturas, ou “strict DER encoding”. Pieter Wuille propôs alguns ajustes no BIP em julho de 2014, que teriam resolvido o problema:

- 2014-Jul-18: Para fazer com que as regras de codificação de assinaturas do Bitcoin não dependam do analisador específico do OpenSSL, modifiquei a proposta do BIP62 para que seu requisito de assinaturas DER estritas também se aplicasse a transações de versão 1. Nenhuma assinatura não DER estava sendo minerada em blocos naquela época, então presumiu-se que isso não teria nenhum impacto. Veja <https://github.com/bitcoin/bips/pull/90> e <http://lists.linuxfoundation.org/pipermail/bitcoin-dev/2014-July/006299.html>. Desconhecido na época, mas se implantado isso teria resolvido a vulnerabilidade.

— Pieter Wuille na lista de discussão Bitcoin-dev, Divulgação: bug de consenso indiretamente resolvido pelo BIP66 (2015)

Devido à amplitude deste BIP, que cobria substancialmente mais do que apenas “strict DER encoding”, ele estava constantemente mudando e nunca chegou perto da implantação. O BIP foi posteriormente retirado porque o Segregated Witness, BIP141, resolveu a maleabilidade das transações de uma maneira diferente e mais completa.

Após a descoberta



O OpenSSL lançou novas versões de seu software com patches que, se usados no Bitcoin desde o início, teriam resolvido o problema. No entanto, usar qualquer nova versão do OpenSSL apenas em uma nova versão do Bitcoin Core tornaria as coisas piores. Gregory Maxwell explica isso em outro [tópico de email](#) em janeiro de 2015:

Enquanto para a maioria dos aplicativos geralmente é aceitável rejeitar ansiosamente algumas assinaturas, o Bitcoin é um sistema de consenso onde todos os participantes devem geralmente concordar sobre a exata validade ou invalidade dos dados de entrada. Em certo sentido, a consistência é mais importante do que a “correção”.

...

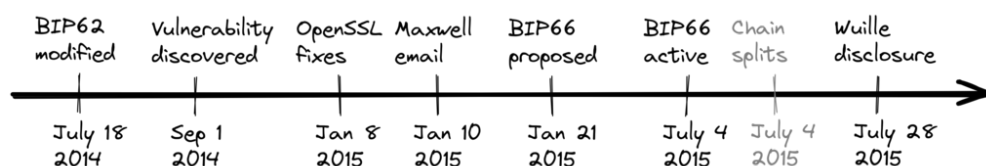
Os patches acima, no entanto, corrigem apenas um sintoma do problema geral: confiar em software não projetado ou distribuído para uso normativo de consenso (em particular o OpenSSL) para comportamento normativo de consenso. Portanto, como uma melhoria incremental, proponho um soft fork direcionado para impor conformidade estrita ao DER em breve, utilizando um subconjunto do BIP62.

— Gregory Maxwell sobre atualização do OpenSSL, lista de discussão Bitcoin-dev

Ele destaca que usar código que não é destinado ao uso em sistemas de consenso representa riscos sérios, e propõe que o Bitcoin implemente strict DER encoding. Este é um exemplo muito claro da importância da boa criptografia de seleção, um termo que discutimos em [Section 6.4](#).

Esses eventos podem dar a impressão de que Gregory Maxwell sabia sobre a vulnerabilidade que Pieter Wuille publicou mais tarde, mas queria ajudar a introduzir uma correção disfarçada como uma medida de precaução, sem chamar muita atenção para o problema real. Pode ser isso, mas é pura especulação.

Então, como proposto por Maxwell, o BIP66 foi criado como um subconjunto do BIP62 que especificava apenas a codificação estrita ao DER. Este BIP aparentemente foi amplamente aceito e implantado em julho, embora duas divisões de blockchain tenham ocorrido ironicamente devido à *mineração sem validação*. Essas divisões são discutidas na próxima seção.



Uma lição importante a se tirar do BIP66 é que os BIPs devem ser mais ou menos *atômicos*, o que significa que devem ser completos o suficiente para fornecer algo útil ou resolver um problema específico, mas pequenos o suficiente para permitir um amplo suporte entre os usuários. Quanto mais coisas você coloca em um BIP, menor é a chance de aceitação.

Divisões devido à mineração sem validação



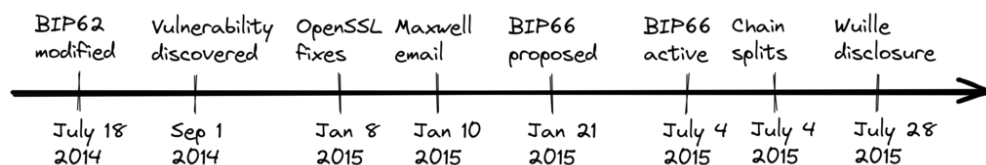
Infelizmente, a história do BIP66 não terminou aí. Quando o BIP66 foi ativado, a situação ficou bastante complicada porque alguns mineradores não validavam os blocos que estavam tentando estender. Isso é chamado de mineração sem validação, ou SPV-mining (como em Simplified Payment Verification). Uma mensagem de alerta foi enviada aos nodes do Bitcoin com um link para [uma página da web descrevendo o problema](#).

No início da manhã de 4 de julho de 2015, o limite de 950/1000 (95%) foi atingido. Logo depois, um pequeno minerador (parte dos 5% não atualizados) minerou um bloco inválido – como era esperado ocorrer. Infelizmente, descobriu-se que cerca de metade do poder de hash da rede estava minerando sem validar completamente os blocos (chamado de mineração SPV), e construíram novos blocos em cima daquele bloco inválido.

— Desenvolvedores do Bitcoin Core, Informação de alerta em [bitcoin.org](#) (2015)

A página de alerta instruiu as pessoas a esperar por 30 confirmações adicionais além das que normalmente fariam caso estivessem usando versões mais antigas do Bitcoin Core.

A divisão mencionada acima ocorreu em 2015-07-04 às 02:10 UTC após o bloco de altura [363730](#). Este problema foi resolvido às 03:50 do mesmo dia, após 6 blocos inválidos terem sido minerados. Infelizmente, o mesmo problema ocorreu novamente no dia seguinte, ou seja, em 2015-07-05 às 21:50, mas desta vez o branch inválido durou apenas 3 blocos.



Os eventos que levaram ao BIP66, sua implantação e suas consequências são um excelente estudo de caso sobre o quão cuidadosos os desenvolvedores de Bitcoin precisam ser. Algumas lições importantes do BIP66:

- O equilíbrio entre abertura e não publicação de uma vulnerabilidade é delicado.
- Implantar correções para vulnerabilidades não publicadas é um jogo arriscado.
- Manter o consenso é difícil.
- Software não destinado a sistemas de consenso geralmente é arriscado.
- Os BIPs devem ser um tanto atômicos.

8.3. Conclusão

O Bitcoin tem bugs. As pessoas que descobrem bugs são incentivadas a divulgá-los responsavelmente aos desenvolvedores do Bitcoin, para que possam corrigir o bug sem revelá-lo publicamente. Idealmente, a correção do bug pode ser disfarçada como uma melhoria de desempenho ou algum outro tipo de cortina de fumaça.

Vimos algumas das questões mais graves que surgiram ao longo dos anos, e como foram tratadas. Algumas foram descobertas publicamente por meio de explorações, enquanto outras foram divulgadas responsavelmente e puderam ser corrigidas antes que atores mal-intencionados tivessem a chance de explorá-las.

Appendix A: Perguntas para Discussão

Essas perguntas para discussão não são apenas um resumo do conteúdo em “Filosofia do desenvolvimento do Bitcoin”, elas são feitas para incentivá-lo a pesquisar mais, então certifique-se de explorar mais.

A.1. Descentralização

- A descentralização é difícil. Por que passamos por todo esse esforço para fazê-la funcionar? Poderíamos optar por uma abordagem híbrida, onde algumas partes são centralizadas e outras não?
- A descentralização introduz o problema do gasto duplo, ou o problema do gasto duplo exige descentralização? Como Satoshi resolveu o problema do gasto duplo?
- Em quais aspectos o Bitcoin ainda é mais propenso à censura, e por que a censura é algo tão ruim? Existem argumentos a favor da censura?
- Afirma-se que o Bitcoin é permissão nula. Existem outros métodos de pagamento que você poderia considerar de permissão nula, desnecessária para seu uso?

A.2. Confiança Nula

- A confiança nula é frequentemente um espectro, não binário. Quais aspectos do Bitcoin são mais confiáveis e quais tipicamente envolvem um nível mais alto de confiança? Eles podem ser mitigados?
- Você quer executar um nó completo para poder validar completamente todas as transações. Você baixa o Bitcoin Core de <https://bitcoin.org/en/download>. Onde você colocou sua confiança, e onde você é totalmente sem confiança?
- É possível construir um sistema de confiança nula em cima de um sistema confiável?

A.3. Privacidade

- Quais são alguns benefícios importantes que um usuário obtém ao manter uma boa privacidade ao interagir com o Bitcoin? Quais são alguns benefícios altruístas para a rede?
- Como o reuso de endereços afeta sua privacidade?
- O Bitcoin usa um modelo UTXO, enquanto algumas criptomoedas alternativas usam um modelo de conta. Quais são as implicações dessa escolha na privacidade?

A.4. Oferta Finita

- Qual é a relação entre a oferta finita do Bitcoin e sua emissão de moedas através da transação de coinbase? Qual é a relação entre a emissão de moedas e o orçamento de segurança, e como eles estão em conflito?
- Quais parâmetros Satoshi poderia ter ajustado para alterar o limite de oferta do Bitcoin? O que mudaria se ele tivesse decidido limitar a oferta em 1 milhão? E se fosse 1 trilhão?

- Por que algumas pessoas defendem um aumento na oferta de Bitcoin? Você acha que isso vai acontecer?

A.5. Atualizações

- O que é o Speedy Trial e por que foi necessário para ativar o Taproot?
- Por que precisamos de um percentual tão alto de mineradores para atualizar em um softfork? Por que o limiar não é apenas 51%?

A.6. Pensamento Adversarial

- O que é um ataque Sybil, e por que uma rede descentralizada é tão propensa a ele?
- Por que é importante que todos os jogadores na rede Bitcoin - e não apenas os desenvolvedores - pensem de forma adversarial?

A.7. Código Aberto



- Apenas um punhado de mantenedores tem as permissões necessárias no GitHub para fazer merge de código no repositório [Bitcoin Core](#). Isso não é contraditório com uma rede permissionless?
- O processo de desenvolvimento de código aberto é suscetível a um ataque Sybil? Se sim, como você o enfrentaria?
- Quais são os benefícios e desvantagens de confiar em bibliotecas de código aberto de terceiros, e qual é a abordagem adotada com o Bitcoin Core?
- De que maneiras precisamos de revisão além da revisão de código? Como determinar quanto de revisão é suficiente?
- Como garantimos que sempre haverá pessoas com expertise suficientes trabalhando no Bitcoin? O que acontece quando não há, e como avaliamos sua integridade e intenções?

A.8. Escalabilidade

- Argumenta-se que o sharding oferece benefícios de escalabilidade ao custo de complexidade. Por que devemos ou não adotar melhorias tecnológicas porque são difíceis de entender, mesmo que pareçam tecnicamente sólidas?
- Quais são alguns exemplos de métodos de escalabilidade interna introduzidos no Bitcoin?
- Por que a escalabilidade vertical é muito mais difícil em um sistema descentralizado? E quanto à escalabilidade horizontal?
- Parece que ainda estamos longe de ter um consenso sobre como poderíamos colocar o mundo inteiro no Bitcoin. Satoshi não deveria ao menos ter pensado em um caminho para chegar lá, antes de minerar o primeiro bloco em 2009?

- Como você classificaria (vertical, horizontal, interna ou não uma técnica de escalabilidade) cada um dos seguintes: sharding, aumento de tamanho de bloco, SegWit, nós SPV, exchanges centralizadas, Lightning Network, diminuição do intervalo de bloco, Taproot, sidechains.

Appendix B: Feedback e contribuição



O projeto é mantido no [GitHub](#). Se você tiver qualquer melhoria, correção ou sugestão para fazer, abra uma nova issue lá ou envie um pull request.

B.1. Build



O livro é escrito em [AsciiDoctor](#). O arquivo principal do livro é [btcphilosophy.adoc](#). Este arquivo então `include::` cada capítulo como um arquivo separado.

Há também um arquivo `asciidoctor` separado, [sources/sources.adoc](#), que coleta todos os recursos copiados aos quais o livro se refere. Eles são mantidos separados para não tornar a navegação no livro muito pesada.

Antes de construir, você precisa instalar algumas dependências:



- `asciidoctor`, Veja [instruções de instalação](#)
- A extensão `asciidoctor-diagram` para `asciidoctor`. Veja [instruções de instalação](#)
- Gnu `make`

Para construir este livro, clone o repositório GitHub:

```
$ git clone https://github.com/bitcoin-dev-philosophy/btcphilosophy.git
$ cd btcphilosophy
```

e depois construa usando qualquer um dos dois métodos abaixo

B.1.1. Manualmente usando `asciidoctor`

Use esta opção se você só quiser construir o livro casualmente e ler o resultado final localmente. Requer apenas que o `asciidoctor` esteja presente em seu computador. Este método não gerará QR codes para links na margem.

```
$ asciidoctor -v btcphilosophy.adoc
```

A flag **-v** é recomendada e instrui o asciidoctor a ser verboso, o que significa que ele mostrará referências inválidas e outros problemas. O comando acima resultará em um arquivo **btcphilosophy.html** em seu diretório atual que você pode visualizar em qualquer navegador, por exemplo:

```
$ brave-browser btcphilosophy.html
```

O material de origem é coletado e mantido como um livro separado na pasta **sources**. Para construí-lo:

```
$ asciidoctor -v sources/sources.adoc
```

Isso resultará em um arquivo **sources/sources.html** que você pode abrir em um navegador da web da mesma forma que o livro principal.

B.1.2. Usando Gnu **make**

Se você tiver acesso ao comando **make** em uma máquina Linux, você pode construir usando

```
$ make
```

Isso criará um diretório **build** e construirá tanto o livro principal (que ficará em **build/btcphilosophy.html**) quanto o livro de fontes (que ficará em **build/sources/sources.html**).

Se você tiver o ImageMagick instalado, o **make** reduzirá o tamanho de algumas imagens para um download mais rápido pela web. Caso contrário, ele escreverá uma mensagem de aviso dizendo que as imagens não serão redimensionadas.

Este método de construção é geralmente preferido, pois mantém a árvore de origem separada dos arquivos de saída do asciidoctor, reduz o tamanho das imagens para torná-las mais apropriadas para servir o livro na web e produz QR codes na margem para os links.

B.1.3. Construir um PDF

Se você tiver o **asciidoctor-pdf** instalado, pode criar um PDF que dá um resultado de melhor qualidade do que primeiro gerar html e depois usar “Print To PDF” a partir disso. Crie o PDF emitindo **asciidoctor-pdf** manualmente ou usando **make pdf**. Para emitir o comando manualmente:

```
$ asciidoctor-pdf btcphilosophy.adoc
```

O resultado será escrito em **btcphilosophy.pdf**. Para usar **make**:

```
$ make pdf
```

O resultado será escrito em `build/btcphilosophy.pdf`.

Este trabalho é uma tradução do original, que foi licenciado sob a Creative Commons Attribution 4.0 International (CC BY 4.0). Foi realizada uma adaptação para incluir uma nova capa com o branding da Scalar School. Para mais informações sobre a licença, acesse: <https://creativecommons.org/licenses/by/4.0/legalcode>