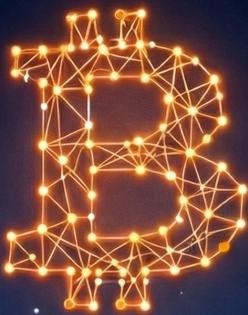


Del Bit al Bitcoin



Entendiendo Blockchain
desde sus Principios
Fundamentales

01010.01100

1

Carlos D. Alegre



Índice

0 Capítulo 0: Bienvenido al Viaje	11
0.1 Qué es Este Libro	11
0.2 Qué Aprenderás	11
0.3 Una Advertencia Sobre la Propaganda	11
0.4 ¿Por Qué Aprender Esto?	13
0.5 Una Nota Sobre el Tono	14
0.6 Una Nota Sobre la Precisión de los Datos y el Uso de IA	14
0.7 La Promesa	14
0.8 Empecemos	15
1 Capítulo 1: El Bit	17
1.1 La Información Más Simple Posible	17
1.2 De Interruptores de Luz a Ordenadores	17
1.3 Binario: El Lenguaje del Dos	18
1.4 Ocho Interruptores = Un Byte	19
1.5 Toda la Complejidad Se Construye Desde Esto	19
1.6 Bajo el Capó: Cómo los Ordenadores Almacenan un Bit, Físicamente	20
2 Capítulo 2: Logos - Mapeando Significado a Números	21
2.1 El Problema del Significado	21
2.2 La Tabla ASCII: Un Acuerdo Social	22
2.3 Mi BRO Binario	22
2.4 Más Allá de las Letras: Todo Son Números	23
2.5 La Idea Profunda: La Información es Acuerdo	23
2.6 Los Lenguajes Funcionan de la Misma Manera	24
2.7 Por Qué Esto Importa para Bitcoin	24
3 Capítulo 3: Algoritmos - Siguiendo Reglas	27
3.1 La Analogía de la Receta	27
3.2 Un Algoritmo Simple: Binario a Letras	28
3.3 Los Ordenadores No “Piensan”	28
3.4 El Poder del Determinismo	29
3.5 Los Algoritmos Pueden Ser Simples o Complejos	29
3.6 Los Ordenadores Siguen Recetas Perfectamente (y Estúpidamente)	30
3.7 Por Qué Esto Importa para Bitcoin	30
3.8 La Transición a Protocolos	31
4 Capítulo 4: Protocolos - Ordenadores Hablando	33
4.1 El Protocolo de la Alergia	33
4.2 El Internet es Solo Protocolos	34
4.3 Los Protocolos Están en Todas Partes	35

4.4	El Protocolo de Internet (IP)	35
4.5	Bitcoin es un Protocolo También	36
4.6	Los Protocolos Permiten Confianza Sin Autoridad	36
4.7	Por Qué los Protocolos Importan para Bitcoin	37
4.8	El Efecto de Red	37
4.9	Los Protocolos son Estándares Vivos	38
5	Capítulo 5: El Problema de la Confianza	39
5.1	Alice, Bob, y Carol	40
5.2	Soluciones Históricas	40
5.3	La Pregunta Fundamental	41
5.4	Por Qué Esto Importa Hoy	41
5.5	La Solución: Cifrado	42
5.6	La Configuración para lo que Viene	42
6	Capítulo 6: Cifrado Simétrico - Secretos Compartidos	45
6.1	El Cifrado César	45
6.2	El Principio de la Contraseña	46
6.3	Por Qué se Llama “Simétrico”	47
6.4	Cifrado Simétrico Moderno	48
6.5	La Falla Fatal	48
6.6	Las Relaciones a Larga Distancia No Funcionaban	49
6.7	Por Qué Esto Importa para Bitcoin	49
7	Capítulo 7: Cifrado Asimétrico - El Truco Mágico y Cómo Crean Monederos Cripto	53
7.1	La Función Unidireccional	53
7.2	Un Ejemplo Para los Curiosos: Multiplicación de Números Primos	54
7.3	Las Dos Claves: Pública y Privada	55
7.4	Cómo Funciona	56
7.5	La Magia Real: Cifrado RSA	57
7.6	Por Qué Esto Importa para Bitcoin	57
7.7	Propiedades Interesantes	59
7.8	Lo Que Hemos Resuelto	60
7.9	Resumen	60
8	Capítulo 8: Computación Cuántica - ¿La Amenaza del Futuro?	61
8.1	La Preocupación	61
8.2	¿Qué es la Computación Cuántica?	61
8.3	Qué se Rompe y Qué No	62
8.4	Por Qué las Criptomonedas Pueden Adaptarse	62
8.5	Todos Tienen Este Problema	62
8.6	La Búsqueda del Tesoro: El Bitcoin de Satoshi	63
9	Capítulo 9: Dinero como Bits Sincronizados	65
9.1	¿Qué Hace que Algo Sea Dinero?	65
9.1.1	Propiedad 1: Escasez (No Puedes Crearlo Fácilmente)	65
9.1.2	Propiedad 2: Verificabilidad (Puedes Probar Lo Que Tienes)	66

9.1.3 Propiedad 3: Sin Doble Gasto (No Puedes Gastar el Mismo Dinero Dos Veces)	66
9.1.4 Propiedad 4: Transferibilidad (Puedes Enviarlo)	66
9.1.5 Propiedad 5: Propiedad (Realmente Lo Controlas)	66
9.1.6 Propiedad 6: Fungibilidad (Cada Unidad es Equivalente)	67
9.1.7 Propiedad 7: Divisibilidad (Se Puede Dividir en Partes Más Pequeñas)	67
9.1.8 Propiedad 8: Durabilidad (No Desaparece ni se Descompone)	67
9.2 La Realización: Los Bits Pueden Tener Estas Propiedades	67
9.3 El Problema de la Base de Datos	68
9.4 La Solución Tradicional: Una Base de Datos, Un Controlador	68
9.5 La Idea: Distribuir la Base de Datos	69
9.6 El Problema del Consenso	69
9.7 Las Preguntas Que Necesitamos Responder	71
9.8 Hablando Históricamente	71
10 Capítulo 10: Proof-of-Work - Ganándote el Derecho a Escribir	73
10.1 Tomando Prestada la Jerarquía (Solo por un Momento)	73
10.2 ¿Quién Obtiene el Derecho a Escribir?	74
10.3 Cómo los Humanos Eligen Líderes	74
10.4 La Primera Hazaña Digital: Proof-of-Work	74
10.5 El Rompecabezas	75
10.6 Por Qué Esto Funciona	76
10.7 El Incentivo: Recompensas de Minería	77
10.8 El Milagro de Bitcoin, los Sueños Húmedos de los Frikis, Tal Vez	77
10.9 Liderazgo Temporal	78
10.10Qué Hemos Resuelto (¿Lo Hemos Hecho?) y Qué No	78
10.11El Panorama General	79
10.12¿Qué es una Blockchain, Realmente?	79
10.13Una Nota sobre el Pensamiento de Satoshi	81
11 Capítulo 11: La Blockchain - Encadenando la Historia	83
11.1 El Problema de la Solución Simultánea	83
11.2 La Elección Imposible	84
11.3 La Solución Elegante: Déjalos Competir	84
11.4 ¿Por Cuánto Tiempo Seguimos Haciendo Esto?	85
11.5 Entendiendo la Finalidad	86
11.6 Ahora, ¿Qué es una Blockchain, Realmente?	87
11.7 Opcional: Para los Curiosos (Puedes Saltarte Esto)	88
12 Capítulo 12: La Estructura de Datos Blockchain (Profundización Técnica)	89
12.1 ¿Cómo Funciona Realmente la Estructura de Datos Blockchain?	89
12.1.1 ¿Qué es una Estructura de Datos? ¿Y Qué es la Memoria de un Ordenador?	89
12.1.2 Punteros: Referencias a Otras Ubicaciones	91
12.1.3 Listas Enlazadas: Encadenando Datos Juntos	91
12.1.4 Funciones Hash (Repaso)	92
12.2 Juntándolo Todo: La Estructura de Datos Blockchain	92
12.3 Por Qué Esta Estructura Importa: Historia Evidente de Manipular	92
12.4 ¿Pero No Puedes Simplemente Recalcular Todos los Hashes?	93
12.5 La Regla de la Cadena Más Larga (Repaso)	94

12.6 La Máquina Anti-Manipulación Psicológica	94
12.7 El Ataque del 51%	95
12.8 La Estructura Blockchain Resumida	95
12.9 Por Qué Esto Importa	96
13 Capítulo 13: Ethereum - La Máquina de Computación Consensuada	97
13.1 Recapitulación de Bitcoin: Transacciones Simples	97
13.2 La Idea: Transacciones Programables	98
13.3 Smart Contracts: Lógica Consensuada En Código	98
13.3.1 Ejemplo 1: Préstamos con Garantía	98
13.3.2 Ejemplo 2: Testamento (Heredando Bits)	99
13.3.3 Ejemplo 3: Suscripción (Pagos Recurrentes Automáticos)	99
13.4 La Máquina Virtual de Ethereum (EVM): Todos Ejecutan los Mismos Programas	100
13.4.1 Direcciones: Todavía Basadas en Claves Asimétricas	100
13.4.2 Cómo Funciona:	100
13.5 Por Qué Esto Es Innovación Revolucionaria: Programas Imparables	101
13.6 ¿Qué Podría Esto Permitir? Deja Que Tu Imaginación Vuele	101
13.7 Los Lados Negativos:	102
13.7.1 1. Tienes Que Pagar Por Ello	102
13.7.2 2. ¡Puedes construir cualquier cosa! Espera... ¿cualquier cosa?	103
13.7.3 3. El Código Puede Tener Bugs (bugs significa errores en el código)	103
13.7.4 4. El Diablo Está En Los Detalles Que Estoy Ocultando Por Simplicidad.	104
13.8 Bienvenido al Lado Oscuro	104
13.9 Resumen: Bitcoin Coordina Valor, Ethereum Coordina Valor Y Computación	104
13.10 Hora de Contar Historias...	105
13.11 Pero Ethereum Comenzó con Proof-of-Work. Luego Algo Cambió...	105
14 Capítulo 14: Cuando el Consenso se Divide - La Naturaleza del Acuerdo	107
14.1 La Pregunta Fundamental	107
14.2 Historia 1: El Fork de Ethereum - El Código Es Ley vs. Proteger El Ecosistema	108
14.2.1 The DAO: Un Experimento de \$150 Millones	108
14.2.2 El Hackeo: \$50 Millones Robados	108
14.2.3 El Dilema: Dos Filosofías Incompatibles	109
14.2.4 La Votación: 85% vs. 15%	110
14.2.5 La División: Dos Ethereums	110
14.2.6 Ten Cuidado, El Pasado Puede Perseguirte	110
14.3 Historia 2: Bitcoin vs. Bitcoin Cash - La Guerra del Tamaño de Bloque	111
14.3.1 El Problema: Bitcoin Es Lento	111
14.3.2 El Debate: Bloques Más Grandes vs. Mantenerlos Pequeños (el tamaño importa)	111
14.3.3 El Fork: Bitcoin vs. Bitcoin Cash	112
14.4 El Patrón: La Tecnología Permite, Los Humanos Deciden	112
14.4.1 1. La Tecnología Permite El Fork	112
14.4.2 2. Los Humanos Deciden El Resultado	113
14.4.3 3. Blockchain Hace El Desacuerdo Auditable	113
14.5 Los Forks Son Guerras Civiles, Más o Menos	113
14.5.1 No Puedes Forzar Consenso Global	114
14.5.2 La Historia Del Desacuerdo Se Preserva	114
14.6 La Realización Más Profunda: De Dos Sociedades A Muchas	115

14.7 Suficiente División—¿Por Qué No Unirse?	115
14.8 Lo Que Esto Significa Para Cualquiera	115
14.9 La Máquina Anti-Manipulación Psicológica, Mejorada	116
15 Capítulo 15: ¡Una Sociedad Para Ti! ¡Una Sociedad Para Mí! ¡Una Sociedad Para Todos!	117
15.1 El Trilema de Blockchain (como un dilema pero con tres opciones)	117
15.1.1 Descentralización + Seguridad = Lento (Bitcoin, Ethereum)	118
15.1.2 Escalabilidad + Seguridad = Centralizado	118
15.1.3 Descentralización + Escalabilidad = Caos (Inseguro)	118
15.2 Por Qué Las RBDC Descentralizadas Son Inherentemente Lentas	119
15.3 ¿Pero Necesitamos Consenso Global Lento Para Todo?	119
15.4 Soluciones de Capa 2: Mini-Sociedades Dentro De Una Mega-Sociedad	120
15.4.1 Ejemplos de Capa 2	120
15.4.2 Estados Unidos	120
15.4.3 Un Ejemplo de Corrupción	121
15.4.4 Otro Ejemplo	121
15.4.5 Otro Otro Ejemplo	121
15.5 Las Noticias Temporalmente Malas: Compromisos	121
15.5.1 La Complejidad Técnica Es Real	122
15.5.2 Carga Cognitiva: Demasiadas Opciones	122
15.5.3 Diferentes Supuestos de Confianza Y Accesibilidad de Software	122
15.5.4 Contexto de Edad de la Industria	123
15.5.5 Fragmentación	123
15.6 Resumiendo	124
15.7 La Seguridad Se Comparte Globalmente	124
15.8 El Último Inconveniente, Privacidad	125
16 Capítulo 16: Pruebas de Conocimiento Cero - Probar Sin Revelar	127
16.1 ¿Qué Es Una Prueba de Conocimiento Cero?	127
16.2 Una Analogía Simple Y Clásica: El Amigo Daltónico	128
16.3 El Avance Matemático	128
16.4 Las Tres Propiedades de las Pruebas de Conocimiento Cero	129
16.4.1 1. Completitud	129
16.4.2 2. Solidez	129
16.4.3 3. Conocimiento Cero	130
16.5 ¿Cómo Ayuda Esto a las Blockchains?	130
16.6 Ejemplos del Mundo Real	130
16.6.1 Ejemplo 1: Transacciones Privadas	130
16.6.2 Ejemplo 2: Probar Que Eres Mayor de 18	130
16.6.3 Ejemplo 3: Votación Privada	131
16.6.4 Ejemplo 4: Probar Solvencia Sin Revelar Saldos	131
16.7 Por Qué Esto Importa	131
16.8 La Pega: Complejidad y Rendimiento	132
16.8.1 Complejidad	132
16.8.2 Rendimiento	132
16.9 El Futuro: Coordinación Privada a Escala	132
16.10 Pero Todavía No Estamos Allí	133

16.11 Una Nota sobre Ordenadores Cuánticos	133
17 Capítulo 17: Lo Que Todo Esto Significa Para La Humanidad	135
17.1 Lo Que Era Imposible Antes	135
17.2 ¿Qué Cambió?	136
17.2.1 Matemáticas	136
17.2.2 Incentivos	136
17.2.3 Consenso Social	137
17.3 El Cambio en las Dinámicas de Poder	137
17.3.1 Antes: Guardianes Centralizados	137
17.3.2 Despues: Coordinación Distribuida	137
17.3.3 La Máquina Anti-Manipulación Psicológica	138
17.4 Implicaciones del Mundo Real	138
17.4.1 Dinero: No Puede Ser Congelado, Censurado, o Arbitrariamente Inflado	139
17.4.2 Gobernanza: Votación Transparente, Neutralidad Creíble, OAD	139
17.4.3 Identidad: Posee Tus Datos, Portables A Través de Plataformas	139
17.4.4 Coordinación: Capa 2 = Mini-Sociedades	139
17.4.5 Privacidad: Conocimiento Cero = Probar Sin Revelar	140
17.5 Despedida de Flami: Dejando la Zona de Propaganda	140
17.6 El Núcleo Filosófico	140
17.6.1 El Consenso Es Social, No Técnico	140
17.6.2 El Valor Es Consensual En Mundos Complejos	141
17.6.3 La Coordinación Es Voluntaria	141
17.6.4 La Tecnología Permite, Los Humanos Deciden	141
17.7 La Invitación	142
17.8 El Mensaje Central	142
17.9 Hemos Desbloqueado Nuevas Formas de Coordinación	143
18 Capítulo 18: El Diablo Está En Los Detalles - Una Bofetada de Realidad	145
18.1 Parte 1: ¿Es Bitcoin Realmente “Dinero”?	146
18.1.1 Lo Que Te Dije	146
18.1.2 La Bofetada de Realidad	146
18.1.3 Propiedades adicionales que Bitcoin NO tiene completamente (todavía):	147
18.1.4 La Verdad Equilibrada	147
18.2 Parte 2: El Problema de la Centralización	148
18.2.1 Lo Que Te Conté	148
18.2.2 La Bofetada de Realidad	148
18.2.3 Concentración de minería:	148
18.2.4 Concentración geográfica:	148
18.2.5 Concentración de riqueza:	149
18.2.6 Concentración de staking de Ethereum:	149
18.2.7 Concentración de desarrollo:	150
18.2.8 La Verdad Equilibrada	151
18.3 Parte 3: El Problema de Energía de Bitcoin	151
18.3.1 Lo Que Te Conté	151
18.3.2 La Bofetada de Realidad	151
18.4 Parte 4: Las Exageraciones	153

18.4.1 “Probar la propiedad de activos digitales sin registro central”:	153
18.4.2 “Coordinar sin intermediarios de confianza”:	154
18.4.3 “Cosas físicamente imposibles ahora son posibles”:	154
18.5 Parte 5: Los Contratos Inteligentes No Son Tan Inteligentes Ni Imparables	154
18.5.1 Lo Que Te Dije	154
18.5.2 La Bofetada de Realidad	154
18.6 Parte 6: Las Capas 2 No Lo Resuelven Todo	156
18.6.1 Lo Que Te Dije	156
18.6.2 La Bofetada de Realidad	156
18.7 Parte 7: El Mito de la Ausencia de Confianza	158
18.7.1 Lo Que Te Dije	158
18.7.2 La Bofetada de Realidad	158
18.8 Parte 8: El Problema del Caso de Uso	161
18.8.1 El Elefante en la Habitación	161
18.9 Parte 9: El Problema de la Ideología	162
18.9.1 Lo Que Te Dije	162
18.9.2 La Bofetada de Realidad	162
18.10 Parte 10: Entonces, ¿Qué Deberías Hacer?	163
18.11 Cerrando La Bola de Demolición	164
18.11.1 Me gustan los trenes.	164
18.11.2 La Verdad Equilibrada	164
19 Capítulo 19: Del Bit al Bitcoin - Resumen Final	169
19.1 El Viaje	169
19.1.1 Parte 1: Fundamentos - ¿Qué Es La Información?	169
19.1.2 Parte 2: Confianza y Criptografía - ¿En Quién Puedes Confiar?	170
19.1.3 Parte 3: Redes de Consenso - El Avance	170
19.1.4 Parte 4: Evolución e Implicaciones - Lo Que Todo Esto Permite	171
19.2 Recapitulación: Los Términos “Raros” (Ahora Deberían Sentirse Más Naturales)	172
19.2.1 Bit	172
19.2.2 Logos	172
19.2.3 Algoritmo	172
19.2.4 Protocolo	172
19.2.5 Hash	172
19.2.6 Criptografía Asimétrica	172
19.2.7 Consenso	173
19.2.8 Proof-of-Work	173
19.2.9 Blockchain	173
19.2.10 Tecnología de Sincronización de Datos Descentralizada y RBDC	173
19.2.11 Base de datos y red	173
19.2.12 Smart Contracts	173
19.2.13 Forks	173
19.2.14 Capa 2	174
19.2.15 Conocimiento Cero	174
19.3 Lo Que Ahora Entiendes	174
19.4 Ahora Puedes...	174
19.5 La Meta-Visión	175
19.6 Del Bit al Bitcoin	175

20 Capítulo 20: Epílogo	177
20.1 Por Qué Escribí Esto	177
20.1.1 La Barrera Emocional	178
20.1.2 El Objetivo del Libro	178
20.2 La Ley de Hierro de la Oligarquía - Por Qué Importa la Educación	179
20.2.1 La Oligarquía de Frikis	180
20.3 Permiso del Autor - Este Libro Es Gratis Para Siempre	180
20.4 Por Qué Esto Importa Más Allá de la Tecnología	181
20.5 Del Bit al Bitcoin	181

0

Capítulo 0: Bienvenido al Viaje

0.1 Qué es Este Libro

Este es un libro sobre comprensión—no sobre exageración publicitaria.

Has oído las palabras de moda: Bitcoin, blockchain, criptomonedas, Web3, NFTs, smart contracts (contratos inteligentes). Has visto los titulares: fortunas hechas y perdidas, promesas revolucionarias, advertencias apocalípticas.

Pero, ¿qué es esta tecnología, realmente?

Este libro responderá esa pregunta desde los primeros principios. Empezaremos con el bloque de construcción más básico—el bit—y construiremos, paso a paso, hasta que entiendas cómo funciona Bitcoin, qué hacen realmente las blockchains (cadenas de bloques), y por qué esta tecnología importa.

No necesitas un título en informática. No necesitas ser bueno en matemáticas. Solo necesitas curiosidad.

0.2 Qué Aprenderás

Parte 1: Fundamentos (Capítulos 1-8) Los conceptos técnicos básicos: bits, teoría de la información, algoritmos, criptografía, firmas digitales, funciones hash. Estas son las herramientas que hacen posible todo lo demás.

Parte 2: Bitcoin (Capítulos 9-12) Cómo funciona Bitcoin: dinero como bits sincronizados, Proof-of-Work (Prueba de Trabajo), consenso, la estructura de datos blockchain. Entenderás el avance que inició todo esto.

Parte 3: Más Allá de Bitcoin (Capítulos 13-16) Ethereum y smart contracts, soluciones de escalado de Capa 2 (Layer 2), Pruebas de Conocimiento Cero (Zero-Knowledge Proofs). La evolución de la tecnología blockchain y qué nuevas capacidades habilita.

Parte 4: Filosofía y Realidad (Capítulos 17-19) Qué significa todo esto para la humanidad—las promesas, las compensaciones, las implicaciones filosóficas. Y lo más importante, la comprobación de la realidad.

0.3 Una Advertencia Sobre la Propaganda

Este libro, hasta el Capítulo 17, ha sido escrito con un tono propagandístico.

No todas las afirmaciones que leerás son 100% verdaderas—el diablo está en los detalles, y el diablo aparece en el Capítulo 18.

¿Por qué propaganda? Porque las tecnologías revolucionarias son difíciles de introducir. Si empezara con todos los problemas, limitaciones y fracasos, dejarías de leer en la página 3.

Así que he tomado el rol de un entusiasta “cripto” para mantenerte comprometido. Los **detalles técnicos**—bits, algoritmos, criptografía, mecanismos de consenso—son precisos, y puedes confiar en ellos. Pero las **implicaciones sociales**—potencial revolucionario, cronogramas de adopción, impacto social—están simplificadas, a veces exageradas, para mantener el impulso.

El Capítulo 18 demolerá esta propaganda con una comprobación exhaustiva de la realidad.



Figura 1: Bola de demolición

Obtendrás el sueño primero. Luego obtendrás la realidad. Y al entender ambos, podrás formar tu propia opinión informada.

0.4 ¿Por Qué Aprender Esto?

Aún así, esta tecnología es revolucionaria—permite a los humanos hacer cosas que eran literalmente **físicamente imposibles** antes. Es como cuando se inventaron los coches.

Pero debido a la naturaleza intrínseca de las tecnologías revolucionarias, si no simplificas su

introducción a alguien que nunca ha oído hablar de ellas, hace que sea mucho más difícil empezar a entenderlas. Por lo tanto, **bienvenido a la zona de propaganda de este libro.** Disfrútala, pero ten en cuenta que no es 100% precisa.

Verás a nuestro amigo Flami () cuando salgamos de la máquina de propaganda.

Sin embargo, ya puedes leer más razones sobre por qué aprender todo esto en el Capítulo 20. No entenderás ni un bit de ese capítulo sin leer todos los anteriores, pero algunas secciones principales pueden entenderse ahora y puede que incluso te motiven a ir a través del resto de este libro aburrido en ciertas partes.

0.5 Una Nota Sobre el Tono

Si eres un lector serio y el flamenco y mi tono ligeramente informal te molestan, lo siento. La mayoría de los frikis reales que conocen esta tecnología profundamente son así.

Además, tengo 22 años, nacido en una era donde nada tiene sentido, donde los valores están constantemente difuminados o destruidos, así que disfrutamos del absurdo—como este flamenco: .

Espero que las diferencias culturales de nuestra brecha generacional no jueguen un papel crucial en tu lectura y comprensión del valor real codificado en este libro.

0.6 Una Nota Sobre la Precisión de los Datos y el Uso de IA

El autor tiene múltiples proyectos al mismo tiempo, por lo tanto, intenté delegar todo el trabajo que este libro requiere a la IA. Algunos datos han sido buscados y verificados por IA, no todos los números presentados pueden ser precisos, sin embargo, los conceptos, dinámicas y formas de entender la tecnología sí lo son.

La mayor parte de este libro está escrito por IA, pero las metáforas, analogías y estructura general vinieron del cerebro del autor.

0.7 La Promesa

Al final de este libro, entenderás:

- Cómo funciona realmente Bitcoin (no solo lo que la gente dice sobre él)
- Qué hace diferente a la tecnología blockchain de las bases de datos normales
- Por qué los smart contracts son poderosos (y limitados)
- Qué habilitan las Pruebas de Conocimiento Cero
- Las compensaciones honestas, limitaciones y fracasos
- Si esta tecnología es adecuada para casos de uso específicos
- Cómo pensar críticamente sobre las afirmaciones de blockchain

No te convertirás en un ingeniero de blockchain, pero entenderás lo suficiente para atravesar la exageración y el pánico, separar la señal del ruido, y tomar decisiones informadas sobre esta tecnología.

0.8 Empecemos

¡Sin más preámbulos, sumerjámonos!

Comienza con el Capítulo 1: El Bit—la unidad de información más simple, y la base de todo lo digital.

Este libro es gratuito porque el conocimiento debería ser accesible. Si encuentras valor en él, compártelo con otros por favor.

1

Capítulo 1: El Bit

Todo empieza con una elección: encendido o apagado, sí o no, 0 o 1.

Levántate y camina hasta el interruptor de luz más cercano. Adelante, esperaré.

Ahora púlsalo. Encendido. Apagado. Encendido. Apagado.

Felicidades. Acabas de realizar la operación más fundamental de toda la informática. Has creado un **bit**.

1.1 La Información Más Simple Posible

Un bit es la unidad de información más pequeña que puede existir. Es una elección entre dos estados:

- **Encendido o Apagado**
- **Sí o No**
- **Verdadero o Falso**
- **1 o 0**

Eso es todo. No existe nada más simple.

Piénsalo: no puedes tener “medio encendido” o “algo así como sí”. Un interruptor de luz está arriba o abajo. Una puerta está abierta o cerrada. Una moneda muestra cara o cruz. Dos estados. Una elección.

Fíjate que, incluso si nos ponemos tan filosóficos como podamos, las cosas existen o no existen. Ser o no ser. Dos estados. Una elección. “Medio ser” o “medio no ser” no tiene sentido lógico.

Esta simplicidad es su poder. Esta simplicidad establece las bases de la lógica y los sistemas modernos de comunicación informática.

1.2 De Interruptores de Luz a Ordenadores

Tu ordenador—en el que probablemente estés leyendo esto ahora mismo—está hecho de miles de millones de pequeños interruptores. No interruptores de luz físicos que puedas pulsar con el dedo, sino interruptores electrónicos microscópicos llamados **transistores** que pueden encenderse y apagarse millones de veces por segundo.

Cada transistor contiene un bit: encendido (1) o apagado (0).

Ahora mismo, mientras lees esta frase, miles de millones de transistores dentro de tu dispositivo se están encendiendo y apagando en patrones precisos. Algunos están guardando las letras de este texto. Otros están rastreando dónde están tus ojos en la pantalla. Otros aún están llevando el tiempo, gestionando la memoria, renderizando colores.

Todo ello—cada página web, cada foto, cada vídeo, cada canción—son solo patrones de bits. Miles de millones de pequeños interruptores, organizados correctamente.

1.3 Binario: El Lenguaje del Dos

Llamamos a este sistema **binario** porque está basado en dos estados (*bi* = dos).

En nuestra vida cotidiana, contamos usando diez dígitos: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Probablemente hacemos esto porque tenemos diez dedos. Cuando nos quedamos sin dígitos, añadimos otra columna y empezamos de nuevo: 10, 11, 12...

Los ordenadores cuentan usando solo dos dígitos: 0 y 1. Cuando se quedan sin dígitos, añaden otra columna. Veamos esto lado a lado:

Humano: 0

1

2

3

4

5

6

7

8

9

¡Nos quedamos sin símbolos! Así que empezamos de nuevo: ponemos un 1, y añadimos un 0

10

11

12...

Binario: 0

1

¡Nos quedamos sin símbolos! Así que empezamos de nuevo: ponemos un 1, y añadimos un 0

10

11

¡Nos quedamos sin símbolos otra vez! Empezar de nuevo con 1, añadir 00 detrás -> 100

100

101

110

111

1000...

Parece extraño al principio, pero es la misma idea—solo que con dos símbolos en lugar de diez.

1.4 Ocho Interruptores = Un Byte

Hagamos esto concreto.

Imagina que tienes ocho interruptores de luz en fila, u ocho bits en fila:

[APAGADO]							
0	0	0	0	0	0	0	0

Cada interruptor puede estar apagado (0) o encendido (1). Eso nos da **256 patrones diferentes posibles** ($2^8 = 256$). Si tienes curiosidad sobre cómo calcular las combinaciones con matemáticas, te animo a investigar y ser curioso. Por ahora, esos detalles no serán necesarios.

Llamamos a ocho bits agrupados juntos un **byte**.

Un byte puede representar:

- Cualquier número del 0 al 255
- Una letra del alfabeto (A-Z, solo hay 26 de ellas—¡tenemos 256 patrones posibles con los que trabajar!)
- Un carácter de puntuación
- Una instrucción para que el ordenador ejecute
- El valor de color de un píxel
- Mil otras cosas

Aquí hay algunos ejemplos de lo que nuestros ocho interruptores podrían significar:

[APAGADO]	[APAGADO]	-> 0						
[APAGADO]	[APAGADO]	[APAGADO]	[APAGADO]	[APAGADO]	[APAGADO]	[ENCENDIDO]		-> 1
[APAGADO]	[ENCENDIDO]	[APAGADO]	[APAGADO]	[APAGADO]	[APAGADO]	[APAGADO]		-> 64
[ENCENDIDO]	[APAGADO]	[APAGADO]	[ENCENDIDO]	[APAGADO]	[APAGADO]	[APAGADO]		-> 144
[ENCENDIDO]		-> 255						

El patrón de interruptores determina el valor. Cambia un interruptor, cambia el significado.

1.5 Toda la Complejidad Se Construye Desde Esto

Aquí está lo que podría parecer imposible: todo lo digital que has experimentado—cada juego, cada película, cada canción, cada sitio web, cada foto, cada mensaje—está construido enteramente a partir de patrones de bits codificando el significado que les dimos.

¿Tu canción favorita? Una secuencia muy larga de 0s y 1s que, cuando se interpreta correctamente, recrea ondas sonoras.

¿Una foto de tu familia? Millones de bits codificando el color y el brillo de cada pequeño píxel.

¿Esta frase que estás leyendo? Cada letra es un patrón específico de 8 bits que tu ordenador sabe cómo mostrar.

¿La blockchain de Bitcoin? Un archivo masivo compartido hecho de bits, siguiendo reglas precisas sobre qué patrones son válidos.

Todo son bits.

La magia no está en los bits mismos. La magia está en cómo los **interpretamos**—cómo asignamos **significado** a los patrones.

Y eso nos lleva a la siguiente gran idea: si los bits son solo patrones, ¿quién decide qué significan esos patrones? ¿Cómo se convierte 01001000 en la letra “H”? ¿Cómo acordamos que una cierta secuencia de millones de bits representa una fotografía y no ruido aleatorio? ¿Cómo acordamos qué transistores representarán los bits que finalmente forman un bit-coin (moneda de bits)?

La respuesta es profunda y simple: **los humanos decidimos**. Inventamos reglas, acordamos estándares, y construimos sistemas que siguen esas reglas.

Eso es de lo que trata el próximo capítulo.

Los lenguajes como el español funcionan de la misma manera—símbolos en forma de sonidos específicos, juntados, creando lo que de otra manera sería ruido aleatorio pero al que damos significado mediante acuerdo. Los bits son solo un nivel más fundamental de la misma idea, uno que es “fácil” de representar con objetos físicos.

1.6 Bajo el Capó: Cómo los Ordenadores Almacenan un Bit, Físicamente

Un bit no se almacena realmente como un “0” o “1”—esos son solo símbolos que usamos. Físicamente, un bit se almacena como:

En un transistor (ordenadores modernos): - Un interruptor diminuto hecho de silicio - “0” = sin carga eléctrica fluyendo en el silicio - “1” = carga eléctrica fluyendo en el silicio - Miles de millones caben en un chip más pequeño que tu uña

En la memoria (RAM): - Pequeñas “baterías” que mantienen o liberan carga.

En un disco duro: - Regiones magnéticas que apuntan al norte o al sur

Los detalles físicos difieren, pero el concepto permanece: **dos estados distinguibles** que llamamos 0 y 1.

Podríamos, teóricamente, hacer un ordenador con puertas y cuerdas. Puerta abierta significa 1 y puerta cerrada significa 0. Tirar de una cuerda podría encender o apagar una puerta. Sería lento e impráctico, pero el principio es el mismo.

Podríamos hacer Bitcoin con puertas—puerta-coin si quieres. Pero los transistores de silicio, hasta ahora, son lo mejor que tenemos, en el sentido de que son simplemente mucho más rápidos y pequeños.

Idea Clave: Toda la complejidad en la informática, toda la tecnología digital, toda la criptomoneda y blockchain—todo se construye desde esta simple base: el bit. Una elección entre dos estados. Encendido o apagado. Sí o no. 0 o 1.

Domina esta idea, y todo lo demás se vuelve comprensible. Se sigue lógicamente.

A continuación, veremos cómo estos patrones sin significado se convierten en información significativa.

2

Capítulo 2: Logos - Mapeando Significado a Números

Los patrones no significan nada hasta que decidimos que significan algo.

Terminamos el último capítulo con una pregunta: Si los bits son solo patrones de 0s y 1s, ¿quién decide qué significan esos patrones?

La respuesta es tanto profunda como simple: **lo hacemos nosotros**. Los humanos. Nos lo inventamos todo. Y eso no es una debilidad—es todo el punto.

2.1 El Problema del Significado

Imagina que te entrego esta secuencia:

01001000 | 01000101 | 01001100 | 01001100 | 01001111

¿Qué es? Para tu ordenador, son solo cinco grupos de ocho interruptores—algunos encendidos, algunos apagados. El patrón existe, pero no tiene significado inherente. No es “secretamente” nada. Es solo... un patrón.

Pero si te digo que estamos usando un sistema llamado **ASCII** (American Standard Code for Information Interchange - Código Estándar Estadounidense para el Intercambio de Información), de repente esos patrones se convierten en:

01001000 → H
01000101 → E
01001100 → L
01001100 → L
01001111 → O

HELLO.

Los bits no cambiaron. El patrón no cambió. Lo que cambió fue que acordamos un **mapeo**: una tabla que dice “este número significa esta letra”.

Esta es la idea del **logos**—significado asignado a la forma. La palabra, el patrón, el símbolo que lleva significación porque colectivamente acordamos que lo hace.

2.2 La Tabla ASCII: Un Acuerdo Social

En los años 1960, un grupo de personas se sentó en una sala y tomó una decisión. Decidieron:

- El número 65 representaría la letra ‘A’
- El número 66 representaría la letra ‘B’
- El número 67 representaría la letra ‘C’
- ...y así sucesivamente

Podrían haber elegido cualquier número. Podrían haber hecho que ‘A’ fuera igual a 200, o 7, o 42, o 43, 69 incluso, si se despertaron cachondos. Fue arbitrario. Pero una vez que estuvieron de acuerdo y lo escribieron, se convirtió en un **estándar**.

ASCII fue estandarizado por primera vez en **1963**, con versiones revisadas siguiendo en 1967 y 1968.

Aquí hay un pequeño pedazo de la tabla ASCII:

Decimal (conteo humano)	Binario	Carácter
65	01000001	A
66	01000010	B
67	01000011	C
...		
72	01001000	H
...		
90	01011010	Z

Ahora, cuando tu ordenador ve 01000001, sabe mostrar la letra ‘A’ en tu pantalla—no porque haya algo mágico en ese patrón, sino porque alguien lo decidió, todos estuvieron de acuerdo, y todos usamos la misma tabla.

Esto es tanto un estándar como un protocolo. Un estándar es una especificación acordada (la tabla ASCII en sí), mientras que un protocolo es un conjunto de reglas para la comunicación (cómo los ordenadores usan esa tabla para intercambiar texto). En la práctica, estos términos se usan a menudo indistintamente cuando se habla de acuerdos compartidos. Al igual que acordamos que el sonido “gato” se refiere a un pequeño animal peludo, acordamos que 01000001 se refiere a la letra ‘A’.

2.3 Mi BRO Binario

Vamos a convertirnos en un bro, como dicen los adolescentes, y escribámoslo en binario: **BRO**

Primero, busca cada letra en la tabla ASCII:

Letra	Decimal	Binario
B	66	01000010
R	82	01010010
O	79	01001111

Así que “BRO” en binario es:

01000010 01010010 01001111

Eso es todo. Esa es la palabra, representada como 24 interruptores (3 letras por 8 bits cada una). Cambia un interruptor, y es una palabra diferente. Cambia la tabla que estamos usando, y el mismo patrón podría significar algo completamente diferente.

Aquí es donde se pone interesante: Si alguien en los años '60 hubiera decidido que 66 significaba 'B', 82 significaba 'R', pero 79 significaba 'A' en lugar de 'O', entonces nuestros bits 01000010 01010010 01001111 deletrearían "BRA" para nosotros hoy en lugar de "BRO".

Imagina que tu ordenador dice "I really love my bro" (realmente amo a mi colega), pero alguien lo interpreta con una tabla diferente, y se convierte en "I really love my bra" (realmente amo mi sujetador). Algunos se ruborizarían. Este juego de palabras funciona en inglés porque las letras 'O' y 'A' crean significados completamente diferentes. Esta es una demostración del poder del significado acordado, y la importancia de los estándares y protocolos para entender y coordinar la comunicación sin malentendidos.

Pruébalo tú mismo: Si tienes curiosidad y quieres un ejercicio mental, busca la tabla ASCII en línea y codifica tu nombre. Cada letra es un número, cada número es un patrón de bits. Cualquier cosa que TÚ signifiques es codificable.

2.4 Más Allá de las Letras: Todo Son Números

El mismo truco funciona para todo lo digital:

Imágenes: Divide la imagen en pequeños cuadrados llamados píxeles, cada uno con un color. Cada color es un número (por ejemplo, Rojo=255, Verde=128, Azul=64), y almacena millones de estos números en un archivo. Cuando abres el archivo, el ordenador reconstruye la imagen leyendo esos números de vuelta.

Sonido: Mides la presión del aire miles de veces por segundo, y cada medición se convierte en un número. Almacena esos números en secuencia, los reproduce de vuelta, y tus altavoces recrean la onda sonora.

Vídeo: Un vídeo es solo muchas imágenes (llamadas fotogramas) mostradas en secuencia rápida, más una banda sonora. Todos números, todos bits.

Bitcoin: Los saldos de las cuentas son números. Las cantidades de las transacciones son números. Las firmas criptográficas son números. La blockchain entera es solo una secuencia muy larga de bits siguiendo reglas específicas.

Fíjate en el patrón: **Todo se convierte en un número. Cada número se convierte en bits. Asignamos significado a esos bits.**

2.5 La Idea Profunda: La Información es Acuerdo

Aquí está la realización clave: **La información no existe "ahí fuera" en el universo. La creamos al acordar un significado.**

La secuencia 01001000 no es "la letra H". Es solo un patrón. Pero cuando miles de millones de personas usan ordenadores que siguen el estándar ASCII, *se convierte* en la letra H para propósitos prácticos. El significado emerge del acuerdo colectivo.

Esto podría sonar abstracto, pero es crucial para entender Bitcoin, porque Bitcoin es la misma idea llevada más lejos:

- El oro es valioso porque acordamos que es valioso (y porque es útil a veces, raro, portable, etc.—pero captas la primera idea por ahora: **el acuerdo es una fuente de valor**)
- Los dólares son valiosos porque acordamos que son valiosos (y el gobierno lo hace cumplir)
- **Bitcoin es valioso porque los participantes acuerdan que es valioso—y el acuerdo está codificado en software que todos ejecutan**

¿Los bits en sí mismos? Sin significado. Pero ¿la *interpretación coordinada* de esos bits? Eso crea dinero, contratos, organizaciones, economías enteras.

2.6 Los Lenguajes Funcionan de la Misma Manera

Piensa en la palabra “gato”.

G-A-T-O

Son solo cuatro sonidos que hacemos con nuestras bocas, o cuatro símbolos que escribimos en papel. No hay nada inherentemente “gatuno” sobre el sonido “gato”. En inglés, es “cat”. En japonés, es “neko” (). Mismo animal, diferentes sonidos, diferentes símbolos.

Pero dentro de las comunidades de habla hispana, todos acordamos: el sonido “gato” se refiere a ese pequeño animal peludo. Cambia el acuerdo, y los mismos sonidos podrían significar algo más completamente. En francés, “chat” (pronunciado diferente) significa gato.

Los bits funcionan de la misma manera, solo que a un nivel más fundamental. Estamos asignando significado a las formas de estados de transistores en lugar de las formas de sonidos de boca o trazos de pluma.

El lenguaje es significado mapeado a patrones de sonido. La escritura es significado mapeado a patrones visuales. La informática es significado mapeado a patrones eléctricos.

¿Y Bitcoin? Bitcoin es **valor mapeado a patrones de bits**, con el mapeo hecho cumplir no por gobiernos o bancos, sino por matemáticas y código representando incentivos alineados que su comunidad acuerda que son valiosos.

2.7 Por Qué Esto Importa para Bitcoin

Podrías estar preguntándote: ¿por qué estamos hablando de tablas ASCII en un libro sobre Bitcoin?

Porque Bitcoin está construido sobre la misma base. Son bits siguiendo reglas, donde las reglas están socialmente acordadas.

- ¿Una clave privada de Bitcoin? Un número de 256 bits.
- ¿Una transacción de Bitcoin? Un patrón específico de bits con una estructura particular.
- ¿La blockchain? Una secuencia de estos patrones, enlazados con hashes criptográficos.
- ¿Hashes? ¿Estamos hablando de drogas? No te preocunes, llegaremos allí, y será intuitivo y lógico.

El protocolo de Bitcoin es como la tabla ASCII, pero para dinero: - “Este patrón de bits representa 1 BTC” - “Este patrón prueba que Alice lo posee” - “Este patrón lo transfiere de Alice a Bob”

Nada de esto es “real” en el sentido de que el oro es real. Pero es real en el sentido de que el lenguaje es real: existe porque colectivamente actuamos como si lo hiciera, siguiendo reglas compartidas.

En realidad, sí, alguna parte *es* real: millones de transistores en ordenadores muy específicos alrededor del mundo, sincronizados de una manera muy específica. ESO ES BITCOIN.

Y eso es poderoso. Porque a diferencia del oro (pesado, difícil de dividir, difícil de transportar) o el dinero gubernamental (controlado por autoridades centrales), Bitcoin es: - Información pura (muévelo a la velocidad de la luz) - Perfectamente divisible (hasta 0.00000001 BTC) - Globalmente accesible (cualquiera con internet puede participar) - **Y las reglas son transparentes, auditables, y hechas cumplir por matemáticas codificadas y algoritmos representando ciertos incentivos**

Pero nos estamos adelantando. Todavía necesitamos entender cómo los ordenadores realmente *hacen* cosas con estos bits. ¿Cómo siguen las reglas? ¿Cómo toman 01001000 y lo convierten en la letra ‘H’ en tu pantalla?

Eso es de lo que trata el próximo capítulo: **algoritmos**—los manuales de instrucciones que hacen útiles a los ordenadores.

Idea Clave: La información es significado que asignamos a patrones. Los ordenadores no “entienden” nada—siguen mapeos acordados (como ASCII) que traducen bits en cosas que los humanos reconocen. Toda la información digital, incluido el dinero, está construida sobre esta base de acuerdo social codificado en protocolos.

A continuación, veremos cómo los ordenadores realmente *procesan* esta información. ¿Cómo toman reglas como “01001000 = H” y las ejecutan miles de millones de veces por segundo? Ese es el poder de los **algoritmos**—y probablemente son más simples de lo que podrías esperar.

3

Capítulo 3: Algoritmos - Siguiendo Reglas

Los ordenadores no piensan. Siguen recetas, muy rápido, muy precisamente.

Hemos establecido que los bits son solo patrones, y asignamos significado a esos patrones a través de mapeos acordados como ASCII. Pero, ¿cómo hacen realmente *algo* los ordenadores con esta información?

La respuesta: **algoritmos**. Palabra elegante, concepto simple. Un algoritmo es solo un conjunto de instrucciones—una receta para resolver un problema o completar una tarea.

3.1 La Analogía de la Receta

Piensa en hacer un bocadillo de jamón y queso. Aquí está el algoritmo:

1. Consigue dos rebanadas de pan
2. Consigue el jamón
3. Consigue el queso
4. Consigue un cuchillo
5. Pon una rebanada de jamón en una rebanada de pan
6. Pon una rebanada de queso encima del jamón
7. Pon la segunda rebanada de pan encima
8. ¡Listo!

Eso es un algoritmo—una secuencia de pasos que, si se siguen exactamente, produce el resultado deseado.

La propiedad clave: Si sigues los mismos pasos con los mismos ingredientes, obtienes el mismo bocadillo cada vez. Esto se llama ser **determinista**: la misma entrada lleva a la misma salida, siempre. Piensa en entrada como “información inicial” o “símbolos iniciales”, y piensa en salida como “información final” o “símbolos finales”.

Los ordenadores funcionan de la misma manera. Dale a un ordenador un algoritmo y algunos datos de entrada (de ahora en adelante, si lees la palabra “datos” también puedes pensarla como “información”), y ejecutará los pasos exactamente como están escritos, produciendo la misma salida cada vez. ¿Y si no? Bueno, has sido hackeado, o deberías contactar al fabricante de tu ordenador—estroppearon algún transistor, cable u otro componente físico.

3.2 Un Algoritmo Simple: Binario a Letras

Escribamos un algoritmo que tome datos binarios y los convierta en texto usando la tabla ASCII que aprendimos en el Capítulo 2.

Entrada: Una cadena de bits (por ejemplo, 01001000 01001001)

Salida: Texto legible por humanos (por ejemplo, “HI”)

Algoritmo:

Paso 1: Dividir la entrada en grupos de 8 bits
 01001000 | 01001001

Paso 2: Convertir cada grupo de binario a decimal
 01001000 = 72
 01001001 = 73

Paso 3: Buscar cada número decimal en la tabla ASCII
 72 = H
 73 = I

Paso 4: Juntar las letras
 Salida: "HI"

Eso es todo. Cuatro pasos. Síguelos con precisión, y 01001000 01001001 se convierte en “HI” cada vez.

Tu ordenador hace esto miles de millones de veces por segundo. Cuando abres un archivo de texto, cargas una página web, o lees esta frase, tu ordenador está ejecutando algoritmos que convierten bits en píxeles, sonidos, letras e imágenes.

3.3 Los Ordenadores No “Piensan”

Aquí hay algo crucial que entender: **Los ordenadores no piensan. No entienden. Solo siguen instrucciones.**

Cuando tu ordenador muestra la letra ‘H’, no “sabe” qué significa ‘H’. No entiende el concepto de letras o lenguaje. Simplemente siguió este algoritmo:

```
SI bits = 01001000
ENTONCES mostrar patrón de píxeles #72 del archivo de fuente
```

Eso es todo. Sin comprensión, sin inteligencia, sin conciencia. Solo: ve este patrón, haz esta acción.

Esto podría parecer decepcionante, pero en realidad es profundo. Al seguir reglas simples muy rápido, los ordenadores pueden hacer cosas que *parecen* inteligentes:

- Traducir idiomas (siguiendo reglas gramaticales + búsquedas en diccionario)
- Jugar al ajedrez (siguiendo reglas de evaluación de movimientos)
- Reconocer caras (siguiendo reglas de coincidencia de patrones)
- Enrutar tus correos (siguiendo reglas de protocolo de red)
- **Procesar transacciones de dinero digital (siguiendo reglas de validación)**

Todo son algoritmos. Todas instrucciones. Todo determinista.

3.4 El Poder del Determinismo

Determinista significa: dada la misma entrada, siempre obtienes la misma salida.

Esta propiedad es crítica para que los ordenadores trabajen juntos. Imagina si tu ordenador y mi ordenador pudieran mirar los mismos bits y obtener resultados *diferentes*—la comunicación sería imposible, y la coordinación se desmoronaría completamente.

Bitcoin confía completamente en esta propiedad. Miles de ordenadores alrededor del mundo ejecutan los mismos algoritmos sobre los mismos datos, y todos llegan a las mismas conclusiones sobre qué es válido y qué no.

Por esto funciona Bitcoin. No por magia, sino porque los algoritmos deterministas garantizan que todos siguiendo las mismas reglas terminan con la misma “verdad”—información predecible, verificable.

¿Qué es un minero? ¿Un bloque? ¿Una transacción? Llegaremos allí, no te preocupes. Por ahora, solo entiende que todos estos conceptos están construidos sobre algoritmos que los ordenadores siguen con precisión.

3.5 Los Algoritmos Pueden Ser Simples o Complejos

Algunos algoritmos son triviales:

Algoritmo: Sumar dos números

Entrada: 5, 3

Paso 1: Sumar los números juntos

Salida: 8

Otros son increíblemente complejos:

Algoritmo: Renderizar un fotograma de videojuego 3D

Entrada: Posición del jugador, datos del mundo, reglas de iluminación, física...

Paso 1: Calcular qué objetos son visibles

Paso 2: Aplicar iluminación y sombras

Paso 3: Aplicar texturas a las superficies

Paso 4: Calcular reflejos

Paso 5: Aplicar desenfoque de movimiento

Paso 6: Convertir coordenadas 3D a píxeles 2D

... (cientos de pasos más)

Salida: Un fotograma del juego (1/60 de segundo)

Pero ambos son la misma idea fundamental: una secuencia de instrucciones que transforma entrada en salida.

Bitcoin usa ambos tipos—algoritmos simples y complejos. Pero todo es determinista. Todo es solo seguir reglas.

3.6 Los Ordenadores Siguen Recetas Perfectamente (y Estúpidamente)

Aquí hay un ejemplo clásico que muestra tanto el poder como la limitación de los algoritmos.

Imagina que le das a un ordenador este algoritmo:

Algoritmo: Hacer un bocadillo

Paso 1: Poner jamón en el pan

Un humano entendería: conseguir pan, abrir el paquete de jamón, extenderlo, etc.

Pero un ordenador fallaría inmediatamente. “¿Poner”? ¿Qué significa eso? “¿Jamón”? ¿Dónde está? “¿En”? ¿Cuál es la posición?

Los ordenadores necesitan *cada paso* deletreado explícitamente:

Paso 1: Localizar objeto etiquetado "pan" en sistema de coordenadas

Paso 2: Localizar objeto etiquetado "jamón"

Paso 3: Mover jamón a coordenada X, Y, Z sobre el pan

... (cientos de micro-pasos)

Por esto programar es difícil—debes pensar como un ordenador: descomponer todo en los pasos más pequeños posibles, no asumir nada, definir todo.

Pero una vez que lo haces: el ordenador ejecuta esos pasos impecablemente, miles de millones de veces, sin cansarse nunca o cometer un error.

3.7 Por Qué Esto Importa para Bitcoin

Bitcoin es una colección de algoritmos.

Estos algoritmos definen cosas como: - Cómo verificar que alguien realmente posee el dinero que está intentando gastar - Cómo asegurarse de que el mismo dinero no se gasta dos veces - Cómo acordar el orden de las transacciones - Cómo recompensar a las personas que ayudan a asegurar la red

Ordenadores por todo el mundo ejecutan estos mismos algoritmos. Todos siguen las mismas reglas. Dada la misma información, todos llegan a la misma conclusión sobre qué es válido.

No se necesita confianza. Solo matemáticas.

Bueno, no exactamente “sin confianza”. Todavía confías en: - Los algoritmos están correctamente implementados - La mayoría de los participantes están siguiendo las reglas - Tu hardware de ordenador no te está mintiendo

Pero no necesitas confiar en ninguna persona o institución única. Puedes verificar todo tú mismo ejecutando los algoritmos.

Esto es lo que la gente quiere decir con “trustless” (sin confianza) (aunque “trust-minimized” (confianza minimizada) es más preciso). Eventualmente confías en matemáticas y código ejecutando un consenso acordado, no en banqueros y gobiernos, que podrían poner el dinero donde les dijiste... o no.

3.8 La Transición a Protocolos

Los algoritmos dicen a los ordenadores individuales qué hacer. Pero, ¿qué pasa con los ordenadores hablando entre *sí*?

Eso requiere **protocolos**—reglas acordadas para la comunicación. Y eso es exactamente lo que exploraremos en el próximo capítulo.

Protocolos, como... ¡El Protocolo de Internet! Voilà, esa palabra que todos usamos pero poco entendemos: Internet.

Porque Bitcoin no es solo un ordenador ejecutando algoritmos. Son miles de ordenadores coordinándose a través de internet, todos ejecutando los mismos algoritmos, todos hablando el mismo protocolo o estándar, todos convergiendo en la misma “verdad” (información ordenada interpretada con el mismo significado).

Y *ahí* es cuando se pone realmente interesante.

Idea Clave: Los algoritmos son conjuntos de instrucciones que los ordenadores siguen determinísticamente. La misma entrada lleva a la misma salida, siempre. Este determinismo es lo que permite a miles de ordenadores verificar independientemente la misma información y llegar a las mismas conclusiones sin confiar entre sí. Los ordenadores no “entienden” nada—solo siguen reglas perfectamente, miles de millones de veces por segundo.

A continuación, veremos cómo los ordenadores se coordinan a través de redes. ¿Cómo habla tu ordenador con un servidor en otro país? ¿Cómo se mantienen sincronizados miles de nodos de Bitcoin? Ese es el poder de los **protocolos**—y son más simples de lo que parece la palabra.

Tal vez no esperabas salir de este libro con una comprensión intuitiva del famoso Internet. Espero que sea tan fascinante para ti como lo fue para mí cuando lo aprendí por primera vez hace años.

4

Capítulo 4: Protocolos - Ordenadores Hablando

Los protocolos son acuerdos sociales entre máquinas.

Hemos cubierto cómo los ordenadores almacenan información (bits), cómo asignan significado a los patrones (estándares como ASCII), y cómo procesan esa información (algoritmos). Pero hay una pieza crucial que falta:

¿Cómo hablan los ordenadores entre sí?

Cuando cargas una página web, envías un correo, o haces una videollamada, tu ordenador está comunicándose con otros ordenadores—a menudo a miles de kilómetros de distancia. ¿Cómo funciona eso? ¿Cómo se entienden entre sí?

La respuesta: **protocolos**. Reglas acordadas para la comunicación.

4.1 El Protocolo de la Alergia

Empecemos con un ejemplo simple. Imagina que Alice quiere contarle un secreto al ordenador de Bob, pero le preocupa que alguien más podría estar haciendo pasar por Bob. Necesita una manera de verificar su identidad. Por cierto, Alice es la doctora de Bob.

Aquí está el asunto: Bob es alérgico a los plátanos. Este es su historial médico—información privada que solo Bob conoce. Alice puede usar esto para verificar que realmente está hablando con Bob, sin que Bob tenga que revelar su historial médico a nadie que esté escuchando en la red.

Así que ella inventa un protocolo:

Protocolo de Alice:

Regla: Cualquiera que afirme ser Bob debe responder correctamente "¿A qué eres alérgico?" Solo el Bob real conoce la respuesta.

Ahora míralo en acción:

Alice: "¿A qué eres alérgico?"

Bob: "Plátanos."

Alice: "Vale, eres Bob. Aquí está el secreto: Melissa está saliendo con Bryan."

Esto es un protocolo. Un conjunto de reglas que ambas partes siguen para comunicarse exitosamente. Una vez que la identidad se verifica con información que solo Bob conoce, Alice puede compartir el secreto de forma segura, y comienza la comunicación.

¿Qué pasa si alguien más lo intenta?

Alice: "¿A qué eres alérgico?"

Carol (haciéndose pasar por Bob): "Eh... ¿cacahuetes?"

Alice: "Incorrecto. No eres Bob. Vete."

El protocolo funciona porque ambas partes conocen las reglas de antemano (preguntar por alergias en este caso), seguir las reglas prueba la identidad (Bob conoce información privada), y no seguir las reglas significa rechazo.

¡Esto es una red! Alice y Bob coordinándose a través de reglas acordadas. Escala esto a miles de millones de ordenadores, y tienes el Internet.

Nota: Los ordenadores hacen esto con cosas más complejas que alergias o datos médicos. Lo hacen con **criptografía**—funciones matemáticas que prueban la identidad sin revelar secretos. Profundizaremos en esto en la próxima parte del libro. No te preocupes, seguirá siendo intuitivo.

4.2 El Internet es Solo Protocolos

Cuando tecleas “instagram.com” en tu navegador, aquí está lo que realmente sucede (simplificado):

Tu ordenador: "Oye, dame instagram.com"

Servidor de Instagram: "Vale, aquí están los datos de la página web"

Tu ordenador: "Guay. Por cierto, soy Bob (aquí está mi información de inicio de sesión que pruebo)"
 Instagram: "¡Oh hola Bob! Aquí está tu feed personalizado. Bonitas fotos de gatos, por cierto"

Esta conversación sigue un protocolo llamado **HTTP** (Hypertext Transfer Protocol - Protocolo de Transferencia de Hipertexto). Cada navegador web y cada servidor web hablan este protocolo. Todos están de acuerdo en el formato:

Formato de SOLICITUD:

GET /home HTTP/1.1

Host: instagram.com

Cookie: session_id=abc123

Formato de RESPUESTA:

HTTP/1.1 200 OK

Content-Type: text/html

[datos de la página web aquí...]

Este formato extraño, en lugar de preguntar sobre alergias, está preguntando cosas como: Oye tío, ¿cuál es el tamaño de tu pantalla? Tú: así de grande. Ahora el servidor sabe qué tan grandes son las imágenes que tiene que enviarte, por ejemplo. Y se hacen muchas más preguntas, claro, pero esta es la dinámica básica que deberías entender.

Tu navegador no necesita “saber” que Instagram—o cualquier sitio web o aplicación específica—existe. Instagram no necesita “saber” sobre tu navegador o ordenador específico. Simplemente ambos siguen el protocolo HTTP, así que pueden comunicarse sin haberse conocido nunca.

Este es el poder de los protocolos: Extraños pueden coordinarse sin conocerse nunca, siempre que sigan las mismas reglas.

4.3 Los Protocolos Están en Todas Partes

Piensa en la comunicación humana. También tenemos protocolos:

El Protocolo del Lenguaje Español: - Las palabras tienen significados acordados (¿recuerdas ASCII?) - Las reglas gramaticales estructuran las oraciones - El contexto ayuda a desambiguar - Si ambas personas siguen estas reglas, la comunicación funciona

El Protocolo de Llamada Telefónica: 1. Persona A: “¿Hola?” 2. Persona B: “Hola, ¿es [nombre]?” 3. Persona A: “Sí, al habla.” 4. Persona B: [expone el propósito de la llamada: “¡Bob se está ahogando con un plátano, por favor llama a una ambulancia!”]

No pensamos en estos como “protocolos” porque nos son tan naturales, pero de hecho son protocolos en cierto sentido—reglas acordadas que permiten la coordinación.

Los ordenadores también necesitan protocolos, pero no pueden improvisar como los humanos. Necesitan especificaciones **exactas**:

Protocolo de Correo Electrónico (SMTP - Simplificado):

Paso 1: Conectar al servidor de correo en el puerto 25

Paso 2: Decir "HELO" para presentarte

Paso 3: Decir "MAIL FROM: remitente@ejemplo.com"

Paso 4: Decir "RCPT TO: destinatario@ejemplo.com"

Paso 5: Decir "DATA" y enviar el contenido del correo

Paso 6: Decir "QUIT" para cerrar la conexión

Cada cliente de correo y cada servidor de correo sigue estos pasos exactos. Por eso Gmail puede enviar correos a Outlook, que puede enviar a ProtonMail, que puede enviar de vuelta a Gmail.

Todos hablan el mismo protocolo.

4.4 El Protocolo de Internet (IP)

El grande. El protocolo que hace posible el Internet.

Cuando envías datos a través del Internet, se dividen en pequeños trozos llamados **paquetes**. Cada paquete tiene: - **Los datos** (parte de tu mensaje, como una foto de gato que quieras publicar en Instagram) - **Dirección de origen** (de dónde vino—tu dispositivo) - **Dirección de destino** (a dónde va—el servidor de Instagram)

Piénsalo como enviar una carta: el contenido de la carta son tus datos, la dirección de remite es tu dirección IP (IP = Internet Protocol - Protocolo de Internet) de origen, y la dirección del destinatario es tu dirección IP de destino.

Los enrutadores a lo largo del camino miran la dirección de destino y reenvían el paquete hacia su destino. No necesitan saber qué hay dentro—solo siguen el protocolo: “Lee la dirección de destino, reenvía al siguiente salto.”

¿La dirección de tu ordenador? Algo como 192.168.1.5 o 203.0.113.42.

¿La dirección de Instagram? Algo como 31.13.64.35.

Cada dispositivo en el Internet tiene una dirección. El Protocolo de Internet (IP) es el conjunto de reglas sobre cómo formatear paquetes y enrutarlos entre direcciones.

Esto es el Internet. No una cosa física, no una nube, no un éter mágico—solo miles de millones de ordenadores siguiendo el mismo protocolo para enviarse paquetes entre sí. A veces a través de electricidad en cables grandes o pequeños, a veces a través de ondas electromagnéticas, pero siempre siguiendo las mismas reglas.

4.5 Bitcoin es un Protocolo También

Ahora aquí está la conexión: **Bitcoin es un protocolo.**

Al igual que HTTP define cómo se comunican los navegadores web y los servidores, Bitcoin define cómo se comunican los nodos en la red Bitcoin. Un nodo es una palabra elegante para ordenador, o cualquier dispositivo que funciona con procesamiento binario de información en una red.

El Protocolo de Bitcoin especifica cosas como: - Cómo formatear una transacción - Cómo transmitirla a la red - Cómo validarla - Cómo agrupar transacciones en bloques - Cómo acordar cuál bloque es el siguiente - Cómo recompensar a los mineros - Cómo prevenir el doble gasto

Cada nodo de Bitcoin ejecuta software que sigue este protocolo. Cuando sucede una nueva transacción:

Nodo A: "Oigan todos, aquí hay una nueva transacción: Alice → Bob, 0.5 BTC"

[transmite a todos los nodos conectados, transmitir significa como, enviarlo a todos]

Nodo B lo recibe:

Paso 1: ¿Es válida la firma? (Verificar)

Paso 2: ¿Tiene Alice 0.5 BTC? (Verificar)

Paso 3: ¿Se ha gastado esto antes? (Verificar)

Paso 4: ¡Válido! → Almacenarlo, reenviarlo a mis pares

Nodo C lo recibe del Nodo B:

[Ejecuta la misma validación]

¡Válido! → Almacenarlo, reenviarlo a mis pares

[La transacción se propaga por la red en segundos]

Sin servidor central. Nadie “a cargo.” Solo miles de ordenadores siguiendo el mismo protocolo, validando independientemente las mismas reglas, convergiendo en la misma verdad.

4.6 Los Protocolos Permiten Confianza Sin Autoridad

Aquí está la realización profunda:

Con los sistemas tradicionales, necesitas una autoridad de confianza: los bancos validan tus transacciones, los proveedores de correo electrónico (como Gmail u Outlook—no lo mismo que el protocolo de correo en sí) entregan tus mensajes y pueden leerlos, y los gobiernos emiten tus documentos de identidad.

Pero con protocolos, puedes tener **coordinación sin autoridad**: - El Internet funciona porque todos siguen IP, no porque alguien “dirija” el Internet - El correo electrónico funciona porque todos siguen SMTP, no porque una empresa controle el correo - Bitcoin funciona porque todos siguen el protocolo de Bitcoin, no porque alguien controle Bitcoin

El protocolo es la autoridad. Las reglas son transparentes, auditables, y hechas cumplir por matemáticas y código, no por instituciones.

TÚ decides qué consenso ejecutar con tu código. ¿Quiero un suministro máximo de 21 millones? ¿Más? ¿Menos? Estableces lo que quieres al representarlo en software, y si suficientes personas están de acuerdo—más algunos detalles de ingeniería que compartiremos después—eso se convierte en el estándar, el protocolo.

4.7 Por Qué los Protocolos Importan para Bitcoin

Bitcoin resuelve el problema de “¿en quién confías para controlar, emitir o almacenar el dinero?” reemplazando la confianza en instituciones con confianza en un protocolo.

En lugar de confiar en un banco para mantener registros precisos, no congelar tu cuenta, no inflar el suministro, y no censurar tus transacciones—confías en el protocolo de Bitcoin (reglas transparentes), matemáticas (la criptografía funciona), incentivos (los mineros y nodos siguen las reglas porque es rentable), y la mayoría (51%+ son honestos, haciendo que los ataques sean caros o irracionales).

Esta es la idea: Los protocolos pueden coordinar extraños a escala global sin que nadie esté “a cargo.”

El Internet demostró esto para la información. Bitcoin se inspiró en ello y decidió interpretar esa información como dinero. Pero Bitcoin no es solo una entidad—son todas las personas, al mismo tiempo.

4.8 El Efecto de Red

Aquí está por qué los protocolos se vuelven poderosos:

Una vez que suficientes personas adoptan un protocolo, se convierte en el estándar. El correo electrónico no ganó porque fuera perfecto—ganó porque todos lo usaban. El Protocolo de Internet no ganó porque fuera óptimo—ganó porque todos lo adoptaron.

Bitcoin es lo mismo. A medida que más personas ejecutan nodos de Bitcoin, aceptan pagos de Bitcoin, y mantienen Bitcoin, la red se vuelve más valiosa—no porque Bitcoin sea técnicamente “mejor” que las alternativas, sino porque **más personas siguen el protocolo**.

Esto se llama el **efecto de red**: El valor de una red crece exponencialmente con el número de participantes.

- Un teléfono = inútil
- Dos teléfonos = una conexión
- Diez teléfonos = 45 conexiones posibles
- Un millón de teléfonos = medio billón de conexiones posibles

La seguridad de Bitcoin proviene de esto. Cuantos más nodos, más difícil atacar. Cuantos más mineros, más caro reescribir la historia. Cuantos más usuarios, más valiosa la red.

Más adelante generalizaremos la idea de una red blockchain, para que no solo entiendas Bitcoin, sino que también salgas de este libro con la habilidad de empezar a entender otros protocolos de red blockchain también (como Ethereum y Solana, por ejemplo).

Es muy importante que entiendas y aprendas a pensar en ellos en general, porque **TÚ ELIGES QUÉ PROTOCOLO EJECUTAS CON TU ORDENADOR**—y por lo tanto, qué información eventualmente y realmente obtiene un poco más de valor gracias a tu “creencia,” tu interpretación de esos bits coordinados siendo algo que quieras usar como dinero, por ejemplo.

4.9 Los Protocolos son Estándares Vivos

Una última cosa: Los protocolos pueden evolucionar, pero es difícil.

Para cambiar un protocolo, necesitas **consenso** entre todos los participantes. De lo contrario, rompes la compatibilidad.

Ejemplo: Si Gmail de repente decidiera cambiar cómo envía correos, y Outlook no se actualizara, los correos entre ellos dejarían de funcionar.

Por esto los cambios de protocolo son lentos (necesitando acuerdo generalizado), cuidadosamente coordinados (todos los participantes deben actualizarse), y raros (demasiado arriesgado cambiar a menudo).

Bitcoin es lo mismo. Los cambios al protocolo de Bitcoin requieren consenso entre mineros (un tipo especial de nodo—ordenadores que siguen algunas reglas extra), nodos, y usuarios. Esto es intencional—previene que nadie cambie arbitrariamente las reglas.

Exploraremos cómo funciona este consenso con mucho más detalle después. Por ahora, solo entiende: **Los protocolos son acuerdos sociales codificados en software, coordinando ordenadores a escala global.**

Idea Clave: Los protocolos son reglas acordadas para la comunicación. Permiten a extraños coordinarse sin confiar en ninguna autoridad central. El Internet es un protocolo (IP). Bitcoin es un protocolo (la red Bitcoin). Cuando millones siguen el mismo protocolo, obtienes coordinación emergente—nadie a cargo, pero todos siguiendo las mismas reglas, convergiendo en la misma verdad.

A continuación, entramos en **Parte 2: Confianza y Criptografía**. Porque aquí está el problema: Si los ordenadores están hablando a través del Internet, cualquiera puede escuchar. ¿Cómo envías secretos por canales públicos? ¿Cómo te aseguras de que estás hablando con alguien que conoces? ¿Cómo pruebas quién eres sin revelar tu contraseña para que nadie pueda usarla en tu nombre? Eso es lo que resuelve la criptografía—y es uno de los fundamentos de la seguridad de Bitcoin.

5

Capítulo 5: El Problema de la Confianza

El internet es público por defecto. Más abierto que las piernas de tu ex.

Hemos establecido que los ordenadores se comunican usando protocolos—reglas acordadas que permiten la coordinación a través del globo. El Internet funciona porque miles de millones de dispositivos siguen las mismas reglas para enviarse paquetes de datos entre sí.

Pero aquí está el problema: **El Internet es una red pública.**

Si visualizas los ordenadores conectados entre sí con líneas, parece una red:

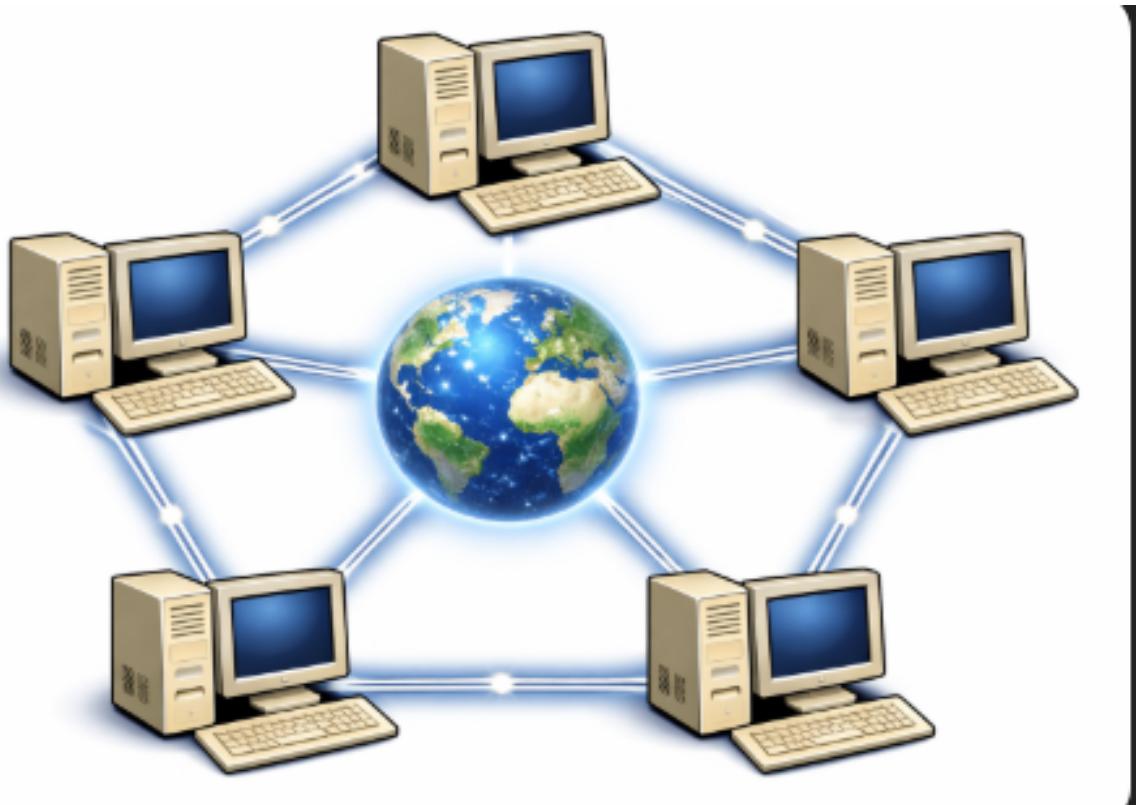


Figura 5.1: internet

Cuando envías datos a través del Internet, no viajan directamente de tu ordenador al destino. En su lugar, saltan a través de docenas de ordenadores intermedios—enrutadores, conmutadores,

servidores—cada uno reenviando tu paquete más cerca de su destino.

Cualquiera a lo largo de ese camino puede leer tus datos.

5.1 Alice, Bob, y Carol

Usemos un escenario clásico de la criptografía.

Alice quiere enviar a Bob un mensaje secreto: “La contraseña es: banana123”

Ella lo envía a través del Internet, y el mensaje viaja a través de: - Su enrutador local - Su proveedor de servicios de internet (ISP) - Múltiples enrutadores troncales - El ISP de Bob - El enrutador local de Bob - Finalmente, el ordenador de Bob

En cada una de estas paradas, alguien podría estar mirando.

Entra Carol. Ella es una fisgona—tal vez trabaja en el ISP, tal vez está ejecutando un enrutador, o tal vez es simplemente alguien que descubrió cómo interceptar tráfico de red (esto se llama **rastreo de paquetes**).

```
Alice envía: "La contraseña es: banana123"
|
[Enrutador 1] <- ;Carol está escuchando aquí!
|
[Enrutador 2]
|
[Enrutador 3] <- ;Carol podría estar escuchando aquí también!
|
Bob recibe: "La contraseña es: banana123"
```

Carol lo ve todo. La contraseña, el mensaje, todo. Alice y Bob no tienen privacidad.

Este es el problema de la confianza: ¿Cómo envías secretos por un canal público cuando cualquiera podría estar escuchando?

5.2 Soluciones Históricas

Este no es un problema nuevo—los humanos han estado intentando enviar mensajes secretos durante miles de años.

Los métodos antiguos incluían mensajeros de confianza (que podían ser capturados o sobornados), cartas selladas (que podían ser abiertas y reselladas), y códigos secretos escritos en papel (que requerían que ambas partes tuvieran el libro de códigos).

La Segunda Guerra Mundial vio la famosa máquina Enigma, el dispositivo de cifrado de Alemania. Los descifradores de códigos aliados pasaron años trabajando para descifrarla, y una vez que lo hicieron, pudieron leer todas las comunicaciones militares alemanas. Guerras se ganaron y perdieron basándose en quién podía mantener sus mensajes secretos.

El patrón: A lo largo de la historia, enviar secretos ha requerido confiar en alguien o algo para que no los intercepte.

5.3 La Pregunta Fundamental

Aquí está lo que hace que el Internet sea diferente de todos los métodos de comunicación anteriores:

El Internet está diseñado para ser abierto. Las bases del Internet se establecieron con **ARPANET en 1969**, que conectaba instituciones de confianza—universidades, laboratorios de investigación, instalaciones gubernamentales. Compartían artículos académicos y datos de investigación, y la privacidad no era la prioridad; la comunicación abierta lo era. El “Internet” moderno como lo conocemos emergió a través de los años 1980 y 1990 con la adopción de los protocolos TCP/IP.

Pero a medida que el Internet creció y conectó al mundo entero, surgió un problema: ¿cómo envías datos privados a través de una red pública? Dentro de un edificio, los cables dedicados funcionan bien. ¿Pero entre ciudades, países, continentes? Necesitas infraestructura—enrutadores, cables, satélites—propiedad y operados por extraños.

No es como una línea telefónica privada entre dos personas. Es una red pública global donde los datos rebotan a través de docenas de ordenadores de extraños antes de llegar a su destino.

Esto crea una paradoja: - Necesitamos que el Internet sea abierto (para que tengamos voz, por lo tanto influencia, en todas partes) - Pero necesitamos que nuestros mensajes sean privados (para que solo las personas que queremos puedan leerlos)

¿Podemos tener privacidad en público?

Durante la mayor parte de la historia humana, la respuesta era: No. No puedes enviar un secreto en público sin que alguien potencialmente lo lea. Necesitas canales privados, mensajeros de confianza, salas seguras.

Pero entonces los matemáticos descubrieron algo notable.

5.4 Por Qué Esto Importa Hoy

Piensa en lo que haces en línea:

- **Banca:** Envías contraseñas y detalles de cuenta. Si Carol intercepta esto, te roba el dinero.
- **Compras:** Ingresas números de tarjetas de crédito que viajan a través de redes públicas.
- **Mensajes:** Registros médicos, documentos legales, fotos personales—todos rebotando a través de ordenadores de extraños.

La pregunta que necesitamos responder: ¿Cómo envían Alice y Bob secretos entre sí cuando Carol está escuchando en la red?

La solución ingenua: “¡Simplemente no dejes que Carol escuche!”

Pero eso es imposible. Carol podría ser: - Una empleada de tu ISP - Una agencia gubernamental con acceso a enrutadores - Una hacker que comprometió una red - La dueña de una red WiFi pública que estás usando - Cualquiera entre tú y tu destino

No puedes controlar quién está mirando. Solo puedes controlar lo que ven.

5.5 La Solución: Cifrado

Probablemente has visto esto en tu navegador web: un pequeño ícono de candado junto al nombre del sitio web. Suele ser verde, y al lado de “https://”, en lugar de “http://”.

```
http://ejemplo.com  <- No seguro (Carol puede leer todo)
https://ejemplo.com <- Seguro (Carol ve galimatías)
```

Esa pequeña “s” al final de “https” significa “secure” (seguro). Significa que tu conexión está **cifrada**.

Cuando visitas un sitio web HTTPS:

- Tu navegador y el sitio web establecen una conexión cifrada
- Todo lo que envías parece ruido aleatorio para cualquiera que esté mirando - Carol puede ver que estás comunicándote, pero no lo que estás diciendo - Es como susurrar en un idioma que solo tú y el sitio web entienden

Por esto funciona la banca en línea. No porque el Protocolo de Internet sea seguro por defecto (no lo es), sino porque ciframos los mensajes que enviamos a través de él.

Puedes imaginarlo como dos personas hablando en un idioma que no conoces, justo frente a ti. Para ti, esa comunicación está cifrada. Como una forma inicial de entenderlo, los ordenadores hacen algo similar pero sobre el internet.

¿Recuerdas ASCII, el estándar que todos entienden? Bueno, imagina que tu ordenador—porque puede computar y asignar significados muy, muy rápido—crea un nuevo mapeo tipo ASCII con el ordenador de destino, y entonces solo ellos saben lo que está pasando. Algo como: “Oye, todas las Bs serán As, todas las As serán Bs y todas las Ys serán Zs”. Entonces la palabra BABY se convierte en ABAZ, y solo ellos lo saben. Más tarde veremos cómo funciona realmente, pero este primer modelo mental te será útil.

Entonces, ¿cómo funciona este cifrado realmente?

5.6 La Configuración para lo que Viene

Necesitamos resolver dos problemas relacionados pero diferentes:

Problema 1: Cómo enviar secretos cuando todos están mirando - Alice quiere enviar a Bob un mensaje privado - Carol está escuchando - El mensaje debe llegar intacto y sin leer

Problema 2: Cómo probar la identidad - Alice recibe un mensaje cifrado - ¿Pero cómo sabe que es realmente de Bob? - ¿Qué pasa si Carol descifró el “idioma” y se está haciendo pasar por Bob? - ¿Cómo pruebas la identidad a largas distancias cuando no pueden verse entre sí?

Aquí están las buenas noticias: **el cifrado resuelve ambos problemas**. Entenderemos intuitivamente cómo más tarde.

Por cierto, lectores, el cifrado usa **matemáticas**. Ambos problemas se basan en el mismo avance fundamental: **criptografía asimétrica**—uno de los descubrimientos matemáticos más importantes del siglo XX.

Pero antes de llegar a esa magia, necesitamos entender primero la versión más simple: **cifrado simétrico**.

Porque el cifrado simétrico plantea el problema que el cifrado asimétrico resuelve. Y entender el problema es la mitad de la batalla.

Idea Clave: El Internet es público por defecto. Cualquiera entre tú y tu destino puede potencialmente leer tus datos. A lo largo de la historia, enviar secretos requería canales privados o mensajeros de confianza. Pero la criptografía moderna permite algo que parece imposible: enviar secretos en público, donde todos pueden ver el mensaje pero nadie puede leerlo. Esta es la base de la privacidad digital—y es esencial para Bitcoin.

Profundicemos en los conceptos muy básicos de cifrado que necesitas para sentirte más seguro y entender cómo funciona Bitcoin.

6

Capítulo 6: Cifrado Simétrico - Secretos Compartidos

Como una cerradura y una llave—pero ambas partes necesitan la misma llave.

Hemos establecido el problema: Alice quiere enviar a Bob un mensaje secreto, pero Carol está escuchando en la red. ¿Cómo pueden comunicarse en privado?

La respuesta empieza simple: **revuelve el mensaje para que solo Bob pueda descifrarlo.**

Esto se llama **cifrado**, y la forma más simple se llama **cifrado simétrico**—donde ambas partes comparten el mismo secreto.

6.1 El Cifrado César

Empecemos con uno de los métodos de cifrado más antiguos: el cifrado César, nombrado así por Julio César quien lo usó para enviar mensajes militares hace más de **2,000 años** (Julio César vivió del 100 a.C. al 44 a.C., haciéndolo aproximadamente hace 2,050 años).

La idea es hermosamente simple: **desplaza cada letra por un número fijo.**

Ejemplo: Desplazar por 3

Alfabeto original: A B C D E F G H I J K L M N Ñ O P Q R S T U V W X Y Z

Desplazado izq. por 1: B C D E F G H I J K L M N Ñ O P Q R S T U V W X Y Z A

Desplazado izq. por 2: C D E F G H I J K L M N Ñ O P Q R S T U V W X Y Z A B

Desplazado izq. por 3: D E F G H I J K L M N Ñ O P Q R S T U V W X Y Z A B C

Ahora codifica un mensaje:

Alfabeto original: A B C D E F G H I J K L M N Ñ O P Q R S T U V W X Y Z

↓ ↓ ↓ ↓

Desplazado por 3: D E F G H I J K L M N Ñ O P Q R S T U V W X Y Z A B C

Mensaje original: HELLO

H → K

E → H

L → O

L → O

O → R

Mensaje cifrado: KHOOR

Alice envía “KHOOR” a través de la red. Carol lo intercepta pero solo ve “KHOOR”—no tiene idea de lo que significa.

Bob recibe “KHOOR” y desplaza en la dirección opuesta, de vuelta por 3:

Alfabeto desplazado:	D E F G H I J K L M N Ñ O P Q R S T U V W X Y Z A B C
	↓ ↓ ↓ ↓
Alfabeto original:	A B C D E F G H I J K L M N Ñ O P Q R S T U V W X Y Z

Mensaje cifrado: KHOOR

K → H
H → E
O → L
O → L
R → O

Mensaje descifrado: HELLO

Esto es cifrado. El mensaje se revuelve usando una regla (en este caso, desplazar 3 posiciones a la derecha en la posición de la letra en el alfabeto), y solo alguien que conoce la regla puede descifrarlo.

La “regla” se llama la **clave**. En este caso, la clave es “3” (la cantidad de desplazamiento). También puedes pensar en la clave como una contraseña que tanto Alice como Bob conocen.

6.2 El Principio de la Contraseña

El cifrado moderno funciona con el mismo principio, solo que con matemáticas mucho más complejas. En lugar de desplazar letras, los ordenadores usan funciones matemáticas que revuelven bits de formas que son extremadamente difíciles de revertir sin la clave.

Piénsalo así:

$f(x)$ → significa una función matemática

Como, por ejemplo: $f(x) = x + 2$

Probablemente has visto esto en el instituto.

x puede ser cualquier número. Algunas funciones incluso tienen 2 entradas:
 $f(x, y) = x + y + 2$

Eliges los valores de x e y (entrada, información inicial), los introduces en la función, y obtienes una salida (información final).

Si elegiste $x = 3$ e $y = 5$:
 $f(3, 5) = 3 + 5 + 2 = 10$

Bueno, también puedes pensar en el cifrado así:

```
f(mensaje, contraseña) = mensaje_revuelto
```

Cifrado: Alice toma su mensaje y una contraseña, los ejecuta a través de una función, y obtiene galimatías revuelto.

Descifrado: Bob toma el mensaje revuelto y la misma contraseña, los ejecuta a través de la función inversa, y recupera el mensaje original.

La función inversa es la función que hace lo “opuesto.” Si sumas uno, el inverso es restar uno. Multiplicar por 3, entonces el inverso es dividir por 3.

Ejemplo:

Mensaje: "La contraseña es banana123"

Contraseña: "secret42"

Función de cifrado compleja:

```
f("La contraseña es banana123", "secret42") = "8x!mQ2$pL9@vN..."
```

Función de descifrado compleja:

```
f_inversa("8x!mQ2$pL9@vN...", "secret42") = "La contraseña es banana123"
```

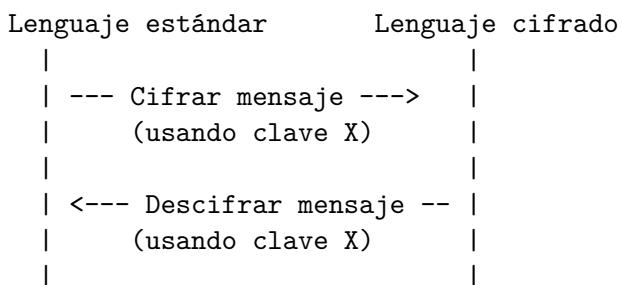
Carol intercepta "8x!mQ2\$pL9@vN..." pero sin conocer la contraseña ("secret42"), no puede descifrarlo. Es solo ruido de aspecto aleatorio para ella.

Esto es cifrado simétrico: Tanto Alice como Bob usan la misma clave secreta (contraseña) para cifrar y descifrar.

Pero aquí está el problema: si Carol conociera la función que usas—las operaciones matemáticas involucradas—podría potencialmente romper el cifrado simplemente adivinando contraseñas. Es fácil ver por qué con el cifrado César, porque solo hay 25 desplazamientos posibles. Así que si Carol sabe que estás usando un cifrado César, puede simplemente intentar desplazar por 1. ¿Todavía no tiene sentido? Vale, desplazar por 2. ¿Aún nada? Intenta desplazar por 3... ¡BINGO! Tiene sentido. Con solo 25 posibilidades, un ordenador puede probar todas ellas muy, muy rápidamente.

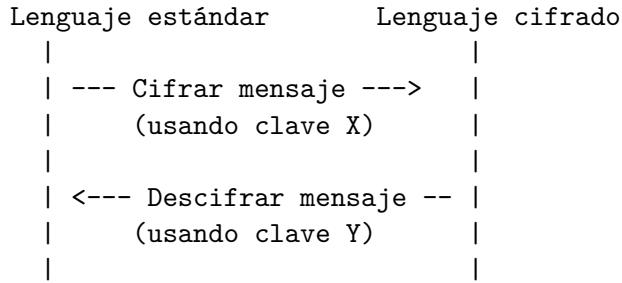
6.3 Por Qué se Llama “Simétrico”

El nombre viene del hecho de que la misma clave funciona en ambas direcciones:



Como una cerradura física y una llave: - Alice cierra la caja con la clave X - Bob abre la caja con la misma clave X - Ambas partes necesitan exactamente la misma clave

Esto es diferente del **cifrado asimétrico** (que cubriremos a continuación), donde Alice y Bob usan claves diferentes:



Pero eso es para después.

6.4 Cifrado Simétrico Moderno

El cifrado César es fácil de romper—solo hay 25 desplazamientos posibles, así que Carol podría probarlos todos en segundos.

El cifrado simétrico moderno usa algoritmos mucho más sofisticados que hacen mucho más que simplemente desplazar caracteres. Un algoritmo famoso y ampliamente usado es:

AES (Advanced Encryption Standard - Estándar de Cifrado Avanzado): - Usado en todas partes: aplicaciones bancarias, sitios web HTTPS, archivos cifrados - En lugar de desplazar letras, revuelve bits usando operaciones matemáticas complejas

El punto: Con una contraseña fuerte, el cifrado simétrico es virtualmente imposible de romper por fuerza bruta. El cifrado César necesita solo 25 cálculos para descifrar, pero con contraseñas suficientemente largas y una secuencia de operaciones suficientemente complejas, tomaría miles de millones de años para que incluso los ordenadores más rápidos prueben todas las contraseñas posibles (claves).

Así que el cifrado simétrico realmente se usa en redes, pero de formas complejas y técnicas diferentes que realmente no necesitas entender aquí. Para los frikis o curiosos, investiguen TLS (Transport Layer Security - Seguridad de Capa de Transporte) y cómo usa cifrado simétrico para transferencia rápida de datos después de establecer una conexión segura. Para la gente normal leyendo esto, simplemente continúen leyendo.

6.5 La Falla Fatal

Entonces hemos resuelto el problema, ¿verdad? ¡Alice y Bob ahora pueden comunicarse de forma segura!

No del todo. Hay un problema masivo: **¿Cómo acuerdan Alice y Bob la contraseña en primer lugar?**

Piénsalo:

Alice quiere enviar a Bob un mensaje cifrado.

Pero primero, necesitan acordar una contraseña.

¿Cómo le dice Alice a Bob la contraseña?

Si la envía por Internet... ¡Carol la intercepta!

Ahora Carol conoce la contraseña y puede descifrar todo.

Este es el problema del huevo y la gallina del cifrado simétrico:

- Tienes matemáticas muy bonitas que permiten enviar mensajes cifrados, pero necesitas una contraseña compartida
- Para compartir la contraseña a largas distancias, necesitas... un canal cifrado
- Pero para tener un canal cifrado, necesitas una contraseña compartida
- Pero para compartir la contraseña... (bucle infinito)

Nota que a cortas distancias Alice puede simplemente encontrarse con Bob en persona y darle la contraseña, pero esto no es práctico en el Internet donde las distancias podrían abarcar continentes.

6.6 Las Relaciones a Larga Distancia No Funcionaban

A lo largo de la historia, la gente resolvió esto encontrándose en persona:

Espías: Dos agentes se encuentran en un callejón oscuro, intercambian libros de códigos cara a cara, luego se comunican de forma segura vía radio.

Militar: Los soldados reciben libros de códigos antes del despliegue. Si el enemigo capture un libro de códigos, todas las comunicaciones están comprometidas.

Banca: Vas al banco en persona, te dan un PIN, luego puedes usar cajeros automáticos y banca en línea.

El patrón: **Secretos pre-compartidos.** Alice y Bob se encuentran en un lugar seguro de antemano y acuerdan una contraseña. Luego pueden comunicarse remotamente usando esa contraseña.

Pero esto no escala al Internet.

No puedes volar a las oficinas centrales de Amazon para intercambiar una contraseña antes de comprar en línea. No puedes encontrarte cara a cara con tu banco antes de usar su sitio web. No puedes visitar físicamente los servidores de Instagram para configurar el cifrado.

El Internet conecta extraños que nunca se han conocido y nunca se conocerán. ¿Cómo pueden establecer un secreto compartido sobre un canal público donde los atacantes están escuchando?

Durante siglos, esto parecía imposible.

Cada método de cifrado requería secretos pre-compartidos, y pre-compartir secretos requiere un canal seguro. Pero, ¿cómo creas un canal seguro sin tener ya un secreto compartido?

Parecía una imposibilidad lógica—como pedirle a alguien que abra una puerta cuando la llave está dentro de la habitación cerrada.

Este fue el problema sin resolver de la criptografía hasta los años 1970.

6.7 Por Qué Esto Importa para Bitcoin

Podrías estar preguntándote: ¿por qué estamos hablando de cifrados César en un libro sobre Bitcoin?

Porque la seguridad de Bitcoin depende enteramente de la criptografía. Específicamente:

Las direcciones de Bitcoin—tu monedero—funcionan con claves criptográficas asimétricas.

Cuando “posees” Bitcoin, no posees realmente nada físico. Posees conocimiento de un número secreto (una clave privada, una contraseña que solo tú deberías conocer) que prueba matemáticamente la propiedad. Sin esa clave, el Bitcoin es matemáticamente inaccesible para ti—y para todos los demás. Allá vas: ahora tienes una mejor comprensión de lo que es realmente un monedero de Bitcoin.

Y porque es solo información—un número muy grande usado como entrada en una función matemática—puede almacenarse en cualquier lugar: papel, hardware, titanio, tu cerebro, etc. Tú eliges el medio. ¿Uno portátil? ¿Uno duradero? Depende de ti y tus necesidades. Hablaremos más sobre esto después.

Las transacciones de Bitcoin se firman con claves privadas.

Cuando envías Bitcoin, estás creando una transacción y firmándola con tu clave privada. Esta firma prueba: 1. Posees el Bitcoin que estás gastando. (Bueno, posees el secreto matemático que otorga acceso a ellos.) 2. Autorizaste la transacción. (Porque, supuestamente, no compartiste ese número con nadie, así que debes ser tú quien los está enviando.) 3. La transacción no ha sido manipulada. (Deduciremos lógicamente esta propiedad después.)

Pero, ¿qué significa “firmar”? En la práctica, es calcular la función matemática que vimos antes:

```
f(mensaje, clave_privada) = mensaje_cifrado.
```

Porque esto te identifica como el propietario del Bitcoin, lo llamamos una firma—como las firmas en los cheques bancarios clásicos en papel o contratos. Realmente puedes pensar en las firmas de criptomonedas como cheques bancarios clásicos. Profundizaremos después.

Pero, ¿cómo funciona esto sobre una red pública donde cualquiera puede mirar? ¿Cómo pruebas que posees acceso a Bitcoin sin revelar tu clave privada? ¿Cómo verifican miles de extraños tu firma (hacen el cálculo matemático) sin conocer su entrada (tu contraseña)?

La respuesta reside en propiedades muy especiales y nuevas que puedes lograr si diseñas la función matemática de forma suficientemente inteligente—y eso es lo que llamamos criptografía asimétrica—el avance matemático que hace posible Bitcoin (y la seguridad moderna de Internet).

Pero tuvimos que entender primero el cifrado simétrico, porque nos muestra el problema que resuelve el cifrado asimétrico. Y porque es relativamente fácil de explicar con modelos mentales simples—como las funciones matemáticas que vimos y el cifrado César.

Así que, resumiendo: ahora sabes cómo pensar apropiadamente sobre el cifrado—funciones matemáticas muy complejas que procesan información, mapeando entradas a salidas. Ayudando a los ordenadores a “inventar” instantáneamente un nuevo lenguaje estándar que solo ellos saben hablar.

Dependiendo de sus propiedades, las clasificamos en cifrado simétrico o asimétrico.

Como hemos visto, las propiedades del cifrado simétrico son: - Ambas partes comparten la misma clave secreta (contraseña) - La misma clave se usa para cifrar y descifrar mensajes

Pero esto tiene una complicación central: ¿cómo compartes la clave secreta en primer lugar a largas distancias sin que alguien la intercepte?

Idea Clave: El cifrado simétrico revuelve mensajes usando una contraseña secreta compartida. Es imposible de romper con algoritmos modernos como AES. Pero tiene una falla fatal: ¿cómo compartes la contraseña rápidamente y a largas distancias sin que alguien la intercepte? Encontrarse en persona funciona, pero no escala al Internet donde miles de millones de extraños necesitan comunicarse de forma segura sin conocerse nunca, y rápidamente.

A continuación, exploraremos el avance que resolvió esto: **cifrado asimétrico**—un truco mágico matemático que te permite enviar secretos sin compartir contraseñas, y probar identidad sin revelar secretos. Este es uno de los bloques fundamentales de Bitcoin y toda la seguridad digital moderna.

Por ahora, estamos empezando a entender verdaderamente qué es realmente un monedero cripto. Es solo una contraseña, un montón de información—una elegida muy inteligentemente.

Capítulo 7: Cifrado Asimétrico - El Truco Mágico y Cómo Crean Monederos Cripto

Dos claves en lugar de una: una clave pública que todos pueden ver y una clave privada que solo tú conoces.

Hemos establecido el problema con el cifrado simétrico: Alice y Bob necesitan compartir una contraseña antes de poder comunicarse de forma segura, pero compartir esa contraseña requiere... un canal seguro. Lo cual requiere una contraseña. Bucle infinito.

Durante siglos, esto parecía imposible de resolver.

Entonces, en los años 1970, los matemáticos descubrieron algo notable—un tipo de función matemática con propiedades muy especiales que parecía casi mágica: **fácil de calcular en una dirección, pero extremadamente difícil de revertir.**

Este avance se llama **cifrado asimétrico**, y lo cambió todo.

7.1 La Función Unidireccional

Empecemos con el concepto central: una **función unidireccional**.

Recuerda del Capítulo 6, hablamos sobre funciones:

$$f(x) = x + 2$$

$$\text{Si } x = 3, \text{ entonces } f(3) = 3 + 2 = 5$$

Esta es una función fácil. Si conoces la salida (5) y la función usada para calcularla, puedes fácilmente descifrar la entrada (3). Simplemente resta 2, una sola operación, sumar 2, que tiene una operación inversa simple, restar 2.

Desafío:

Bob usó $f(x) = x + 2$ para cifrar un mensaje.

El resultado de cifrar el mensaje fue 5.

¿Cuál era el mensaje original?

Sabemos entonces que:

$$x + 2 = 5$$

Lo cual con matemáticas simples podemos resolver:

$$x = 5 - 2$$

$$x = 3$$

Respuesta:

Salida = 5

Entrada/Mensaje enviado = 3

Revertir esta función es muy fácil. Para humanos y para ordenadores.

¿Pero qué pasa si tuviéramos una función que fuera fácil de calcular hacia adelante (entrada a salida) pero extremadamente difícil de revertir (salida a entrada)? Y no solo queremos decir “difícil” como “toma unos segundos”—queremos decir “difícil” como **“tomaría a todos los ordenadores en la Tierra miles de millones de años revertirla.”**

¿Existe tal función?

Sí. Muchas de ellas, en realidad. Son la base de la criptografía moderna.

7.2 Un Ejemplo Para los Curiosos: Multiplicación de Números Primos

Aquí hay un ejemplo intuitivo. Si te sientes confundido, no te preocupes demasiado:

Dirección fácil: Multiplicar dos números primos

Toma dos números primos: 89 y 97

Multiplicalos: $89 \times 97 = 8,633$

Esto es fácil. Un ordenador hace esto instantáneamente.

Dirección difícil: Descifrar los números primos originales

Dado: 8,633

Pregunta: ¿Qué dos números primos se multiplican para dar 8,633?

Esto es mucho más difícil. Tendrías que probar dividir por cada número primo hasta que encuentres los números originales.

Ahora imagina usar **números primos realmente, realmente grandes**:

$p = 32,416,190,071$ (primo de 11 dígitos)

$q = 32,416,187,567$ (primo de 11 dígitos)

$$p \times q = 1,050,807,929,418,200,854,057$$

¡Buena suerte revirtiendo eso sin conocer p y q!

Incluso los ordenadores tardan mucho en adivinar y comprobar todas las combinaciones posibles.

Con primos muy grandes (por ejemplo, primos de 617 dígitos usados en claves RSA de 2048 bits comúnmente usadas en muchos sistemas tradicionales), descifrar los números originales se vuelve esencialmente imposible con la tecnología actual. Incluso los superordenadores más rápidos tardarían más que la edad del universo en encontrar tales números.

Nota importante: Bitcoin no usa realmente RSA o factorización de grandes primos. Bitcoin usa criptografía de curva elíptica (específicamente `secp256k1`), que se basa en números de 256 bits (aproximadamente 77 dígitos decimales)—un tipo diferente de función unidireccional basada en matemáticas de curva elíptica en lugar de factorización de primos. Tanto RSA como la criptografía de curva elíptica proporcionan seguridad fuerte a través de diferentes problemas matemáticos que son fáciles en una dirección pero difíciles de revertir.

Esta asimetría—fácil de calcular, difícil de revertir—es la base de todo cifrado asimétrico, ya sea basado en factorización de primos (RSA) o curvas elípticas (Bitcoin).

¿Por qué números primos? Porque estos números solo pueden dividirse por sí mismos y 1, dejando 0 como resto. Esta propiedad es lo que hace que la multiplicación sea fácil pero la operación inversa difícil. No te preocunes demasiado por los detalles matemáticos—si te sientes confundido sobre lo de los números primos, tómate tu tiempo si quieras, pero esto no es esencial. Simplemente puedes aceptar que hay algunas funciones matemáticas que son fáciles de calcular en una dirección pero difíciles de revertir.

7.3 Las Dos Claves: Pública y Privada

Aquí es donde se pone brillante.

Paso 1: Crear las claves (hecho localmente en tu ordenador)

Cualquiera puede generar un par de claves pública/privada usando algoritmos ampliamente disponibles (como RSA, ECC, etc). Los algoritmos son conocimiento público de la misma manera que cualquiera puede describir y calcular la función $f(x) = x + 2$ —solo necesitas un ordenador para calcularlos rápidamente.

Ejecutas el cálculo en tu ordenador:

```
cálculo_generar_claves() => (clave_privada, clave_pública)
```

El cálculo tiene esa propiedad fácil-solo-una-dirección que discutimos: calcular una clave pública desde una clave privada ($f(\text{Clave privada}) \rightarrow \text{Clave pública}$) es la dirección fácil que cualquiera puede calcular, mientras que el reverso ($f(\text{Clave pública}) \rightarrow \text{Clave privada}$) es la dirección difícil—matemáticamente incalculable con la tecnología actual.

Así que, puedes calcular una clave pública que está matemáticamente y únicamente asociada con tu clave privada, pero nadie puede descifrar tu clave privada desde tu clave pública incluso si conocen el algoritmo (función matemática) que usaste para calcularla/computarla.

Paso 2: Compartir la clave pública

Ahora compartes tu clave pública con todos. Publícalo en línea, envíala en correos, publícalo en una base de datos. No importa quién la vea.

La configuración:

En lugar de una clave compartida (como el cifrado simétrico), el cifrado asimétrico usa **dos claves diferentes**:

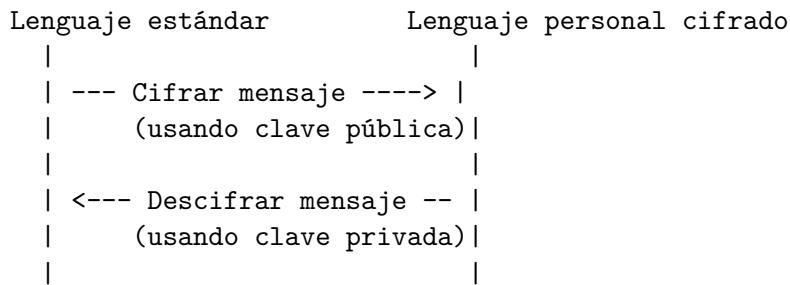
1. **Clave pública** - Compartes esta con todos. Públícalo en internet. Grítala desde los tejados. No importa quién la vea. (Lo mismo que haces con una dirección cripto)
2. **Clave privada** - Mantienes esta en secreto. Nunca la compartas con nadie. Nunca. (Lo mismo que haces con la clave privada del monedero cripto, esa secuencia de 12 o 24 palabras de la que habrás oido hablar)

La magia: Estas dos claves están matemáticamente relacionadas a través de una función unidireccional.

- Puedes **derivar** la clave pública desde la clave privada (dirección fácil)
- **No puedes** derivar la clave privada desde la clave pública (dirección difícil, que puede crear propiedades de anonimato, seguridad, e incluso identificación si se diseña para esos propósitos)

7.4 Cómo Funciona

Así es como funciona la comunicación en el reino digital:



Si alguien quiere hablar con Bob y solo con Bob, cifran su mensaje con la clave pública de Bob. Una vez cifrado, solo Bob puede descifrarlo con su clave privada. Incluso si alguien intercepta el mensaje, no pueden leerlo sin la clave privada de Bob.

Y si Bob quiere comunicarse con cualquier otra persona, puede buscar la clave pública de esa persona y usarla para cifrar su mensaje. Solo esa persona puede descifrarlo con su clave privada.

Solo como un pequeño recordatorio, puedes pensar en el cifrado como traducir de un idioma a otro, como del español a un idioma secreto inventado pero consistente.

Imagina que hay una base de datos pública con las claves públicas de todos. En la práctica no es exactamente así, pero imagínalo—puedes simplemente buscarla y enviar un mensaje privado a cualquiera sin tener que conocerlos o compartir un secreto de antemano.

Ejemplo: Alice y Bob enamorándose

Alice quiere decir "TE AMO, ALICE" a Bob, así que busca la clave pública de Bob y calcula:

```
f("TE AMO, ALICE", clave pública de Bob) = Mensaje cifrado
```

Bob recibe el mensaje cifrado:

```
f_inversa(Mensaje cifrado, clave privada de Bob) = "TE AMO, ALICE"
```

Ahora Bob quiere decir "YO TAMBIÉN TE AMO, BOB". Busca la clave pública de Alice y calcula:

```
f("YO TAMBIÉN TE AMO, BOB", clave pública de Alice) = Mensaje cifrado
```

Alice recibe el mensaje cifrado:

```
f_inversa(Mensaje cifrado, clave privada de Alice) = "YO TAMBIÉN TE AMO, BOB"
```

Si Carol intercepta los mensajes cifrados: - Ve el galimatías cifrado - Podría incluso conocer la clave pública de Bob - Pero **no puede** descifrar el mensaje sin la clave privada de Bob - Y la clave privada de Bob es matemáticamente imposible de derivar desde su clave pública

Esto resuelve el problema de compartir contraseñas.

Alice y Bob nunca tuvieron que conocerse. Nunca tuvieron que compartir un secreto. Bob simplemente publicó su clave pública, y Alice la usó para enviarle un mensaje seguro—todo a través de una red pública donde Carol está mirando todo.

7.5 La Magia Real: Cifrado RSA

El algoritmo de cifrado asimétrico más famoso es **RSA** (nombrado por sus inventores: Rivest, Shamir, y Adleman), que fue inventado en **1977** y publicado en **1978**.

Para los curiosos que quieran profundizar en cómo funciona RSA matemáticamente, hay muchos recursos en línea. Pero para nuestros propósitos, solo debes saber que es un algoritmo ampliamente usado que implementa la propiedad de función unidireccional de la que hemos estado hablando.

Y para los curiosos que quieran saber cómo exactamente publicas tu clave pública en la práctica, usualmente se hace a través de **Infraestructura de Clave Pública (PKI)** o **certificados digitales** (como los certificados SSL/TLS para sitios web). Para Bitcoin específicamente, tu clave pública está relacionada inequívocamente con tu dirección de Bitcoin y se revela a toda la red cuando haces tu primera transacción.

7.6 Por Qué Esto Importa para Bitcoin

Bitcoin no cifra transacciones (son públicas en la blockchain). Lo que esto significa es que todos pueden ver que Bob quiere enviar X bitcoin a Alice, por ejemplo.

Pero Bitcoin usa criptografía asimétrica para algo incluso más importante—probar que Bob es quien quiere enviar la transacción y probar que solo Alice las recibirá: **Probar propiedad sin revelar secretos.**

Recuerda del Capítulo 6, tu monedero de Bitcoin es esencialmente una clave privada. Pero ahora entendamos el panorama completo:

La jerarquía:

```
Clave Privada (número secreto que generas)
```

```
  | (función unidireccional)
```

```
  v
```

```
Clave Pública (derivada de la clave privada, puede compartirse)
```

```
  | (otra función unidireccional)
```

```
  v
```

```
Dirección de Bitcoin (donde la gente te envía Bitcoin)
```

Desglosemos esto:

- Tu **monedero de Bitcoin** es una **clave privada** (un número secreto muy grande)
- Tu **clave pública** se deriva de tu clave privada usando una función unidireccional específica, como RSA.
- Tu **dirección de Bitcoin** (esa larga cadena de letras y números que representa tu monedero y a la cual la gente envía dinero) se calcula desde tu **clave pública** usando otra función unidireccional. Esta usualmente es una función llamada función hash.

¿Por qué este enfoque por capas?

La dirección del monedero no es la clave pública en sí, sino que se deriva de ella. Esto crea propiedades interesantes como:

- Revela menos información sobre tu clave pública, lo cual, aunque nada malo puede pasar si alguien la descifra, es una mejor práctica de ciberseguridad simplemente revelar tan poca información como sea necesario. Los detalles sobre esto son innecesariamente técnicos y más allá del alcance de este libro. Como siempre, si tienes curiosidad, investiga.
- La dirección resultante es más corta y más fácil de compartir que toda la clave pública.

Ejemplo:

Clave pública: 04b0bd634234abdc04b0bd634234abdc04b0bd634234abdc... y más
Dirección de Bitcoin: 1A1zP1eP5QGefi2DMPT (más pequeña y también única)

[estos números fueron inventados, es solo una representación visual]

Si esto representa un número, ¿por qué veo letras? Porque esto está escrito en formato hexadecimal.

Los humanos usamos el formato decimal, basado en 10 dígitos (0-9). Los ordenadores a menudo usan formato binario (basado en 2 dígitos: 0 y 1). El formato hexadecimal está basado en 16 dígitos. Pero solo tenemos 10 símbolos de dígitos (0-9), así que empezamos a usar letras cuando se nos acaban:

SÍMBOLOS ÚNICOS

Decimal: 0 1 2 3 4 5 6 7 8 9

Binario: 0 1

Hexadecimal: 0 1 2 3 4 5 6 7 8 9 A B C D E F

Si realmente no entiendes esto, no te preocunes. Solo piensa que esas letras, de alguna manera, también representan números.

Cuando gastas Bitcoin, creas una **firma digital** usando tu clave privada para señalar que autorizas la transacción. La única parte “digital” de ello es que se hace en un ordenador, pero un término más preciso sería “firma matemática.”

La firma prueba: 1. Posees el Bitcoin (posees la clave privada) 2. Autorizaste la transacción, solo si el punto número 1 es verdadero y eres el único que tiene la clave privada, aquí es donde entra la responsabilidad personal. 3. La transacción no ha sido manipulada. Esto no depende de ti, esta es una propiedad matemática.

La magia: - Cualquiera puede verificar tu firma usando tu clave pública - Pero solo tú puedes crear la firma (requiere tu clave privada) - Y nadie puede descifrar tu clave privada desde tu clave pública

```

El Mundo                                Ordenador de Bob
|                                         |
|                                         |   firmar("Envío 5 BTC a Alice")
|                                         |   [usando clave privada]
|
|   (      Mensaje enviado      )
| <---( "Envío 5 BTC a Alice",    )---|
|   (      salida de firmar     )
|
|

```

Según la imagen anterior, ahora El Mundo tiene 2 cosas: 1. El mensaje: “Envío 5 BTC a Alice” 2. La salida de firmar (la firma) ese mensaje con la clave privada de Bob

Si quieren asegurarse de que solo Bob pudo haber creado ese mensaje, pueden usar la clave pública de Bob para verificar la firma.

Si Bob creó el mensaje, entonces usar las “matemáticas inversas” para firmar debería dar el mensaje. Esto significa:

```
f_inversa(salida de firmar, clave pública de Bob) = "Envío 5 BTC a Alice"
```

Así que el mundo puede simplemente hacer las matemáticas inversas, y si el mensaje coincide, solo puede significar una cosa: este mensaje fue producido por alguien que tiene la clave privada de Bob, que es solo Bob.

El proceso ha sido ligeramente simplificado, el proceso de verificación de la firma es un poco más complejo pero este modelo mental funciona para los intentos y propósitos de este libro. En realidad la función se ve más así:

```
f_verificar(firma, clave pública, mensaje original) = Verdadero/Falso
```

Esto es **criptografía asimétrica al revés**: en lugar de cifrar mensajes, estás firmando mensajes y enviando la firma con ellos. En lugar de “solo Bob puede descifrar,” es “solo Alice pudo haber firmado esto, y todos pueden verificarlo, porque las matemáticas son públicas—todos conocen RSA por ejemplo.”

Exploraremos las firmas digitales en profundidad en el próximo capítulo. Entender la diferencia entre cripto-firmas y cripto-transacciones es crucial para operar con criptomonedas de forma segura.

7.7 Propiedades Interesantes

Resumamos lo que hemos aprendido:

- Cualquiera puede generar un par de claves pública/privada usando algoritmos ampliamente disponibles (como RSA, ECC, etc)
- Los algoritmos son conocimiento público; solo necesitas un ordenador/móvil para calcularlos
- Puedes compartir tu clave pública con cualquiera, y tu dirección de monedero cripto también. Es mejor si solo compartes la dirección del monedero, pero compartir la clave pública también es seguro—nadie te robará dinero solo con eso
- El mundo puede usar tu clave pública para enviarte mensajes cifrados o verificar tus firmas digitales

- Pero solo tú puedes descifrar esos mensajes o crear firmas, porque solo tú tienes la clave privada

7.8 Lo Que Hemos Resuelto

Revisemos lo que nos permite el cifrado asimétrico: la generación de contraseñas únicas y globalmente válidas, comunicación segura sobre canales públicos, y probar identidad sin revelar secretos.

Estas propiedades permitieron cosas como: - Banca en línea segura - Comercio electrónico - Mensajería cifrada agnóstica a la distancia - **Y monederos de criptomonedas**

Todo porque hemos descubierto una manera de crear IDs caseros. Un gran número único imposible de adivinar asociado con otro que puedes compartir, y que cualquiera puede verificar que solo tú creaste ese número compartido, asociando a su creador con él. O en otras palabras, identificando a su creador anónimo con él. Todo asegurado gracias a matemáticas que usan la propiedad fácil-de-calcular-solo-una-dirección.

7.9 Resumen

Si incluso con este nivel de simplificación esto todavía es difícil de captar, no te preocupes, es normal. Estoy intentando explicar cosas para que te sientas más seguro porque entiendes las dinámicas hasta cierto punto. Si esto todavía se siente demasiado complejo, simplemente recuerda una cosa simple:

- EN GENERAL, NUNCA COMPARTAS TU CLAVE PRIVADA CON NADIE.
- NO TU CLAVE PRIVADA, NO TUS MONEDAS.
- SI COMPARTES, ESTÁS CONCEDIENDO ACCESO A ELLAS. ASEGUÍRATE DE QUE SEA SOLO CON PERSONAS EN QUIENES CONFÍAS 101%.

Idea Clave: El cifrado asimétrico usa dos claves en lugar de una. Una clave pública (compartida con todos) y una clave privada (mantenida en secreto). Los mensajes cifrados con la clave pública solo pueden descifrarse con la clave privada. Las claves están matemáticamente relacionadas a través de funciones unidireccionales—fáciles de calcular hacia adelante, esencialmente imposibles de revertir. Esto resuelve el problema de compartir claves: extraños pueden comunicarse de forma segura sin conocerse nunca o pre-compartir secretos.

Ahora que entiendes la criptografía asimétrica—la base matemática de los monederos de criptomonedas y la comunicación segura por internet—hay una pregunta importante que necesitamos abordar: “**¿No romperán los ordenadores cuánticos todo esto?**” Esta es una de las preocupaciones más comunes sobre las criptomonedas. En el próximo capítulo, exploraremos brevemente qué es realmente la computación cuántica, qué puede y no puede romper, y por qué esto no es un escenario apocalíptico para las cripto como algunos imaginan.

8

Capítulo 8: Computación Cuántica - ¿La Amenaza del Futuro?

¿Romperán los ordenadores cuánticos todo lo que acabamos de aprender?

Ahora que entiendes la criptografía asimétrica—la base matemática de los monederos de criptomonedas y la comunicación segura por internet—hay una pregunta importante que necesitamos abordar:

“¿No romperán los ordenadores cuánticos todo esto?”

Esta es una de las preocupaciones más comunes sobre las criptomonedas. Afrontémosla directamente.

8.1 La Preocupación

La criptografía asimétrica se basa en funciones unidireccionales—operaciones matemáticas que son fáciles de calcular hacia adelante pero extremadamente difíciles de revertir. Por ejemplo, multiplicar dos grandes números primos es fácil, pero descifrar qué dos primos se multiplicaron es extremadamente difícil (tomaría a los ordenadores clásicos miles de millones de años). Aquí está la preocupación: **los ordenadores cuánticos podrían ser capaces de revertir estas funciones “unidireccionales” mucho más rápido.**

8.2 ¿Qué es la Computación Cuántica?

Recuerda del Capítulo 1: un bit clásico es 0 o 1—un interruptor que está apagado o encendido.

Un **bit cuántico** (o **qubit**) es diferente. Debido a fenómenos físicos extraños a nivel atómico y temperaturas extremadamente bajas, un qubit puede existir como **0 y 1 al mismo tiempo**. Esto se llama **superposición**.

Bit clásico:

[0] o [1] (un estado a la vez)

Bit cuántico (qubit):

[0 Y 1 simultáneamente] (superposición de estados)

Cuando tienes múltiples qubits en superposición, pueden representar muchas combinaciones posibles a la vez, permitiendo a los ordenadores cuánticos explorar muchas soluciones simultáneamente y

haciendo ciertos tipos de cálculos exponencialmente más rápidos.

Importante: La computación cuántica no es magia. No hace todo más rápido—es muy buena para tipos específicos de problemas (como factorizar números grandes) pero no universalmente mejor en todas las tareas de computación.

8.3 Qué se Rompe y Qué No

Lo que los ordenadores cuánticos podrían romper: - RSA (se basa en factorizar números grandes) - Criptografía de Curva Elíptica / ECDSA (usado por Bitcoin y Ethereum) - Cronograma: Los expertos típicamente estiman que los ordenadores cuánticos capaces de romper la criptografía de clave pública de hoy están **a 10–20 años de distancia**, con algunos situando el riesgo alrededor de los **años 2030**, aunque la incertidumbre y el desacuerdo permanecen sobre los cronogramas exactos.

Lo que permanece seguro: - Cifrado simétrico como AES (solo usa claves más grandes) - Funciones hash (relativamente resistentes) - Algoritmos de criptografía post-cuántica (ya existen y fueron estandarizados por NIST en **agosto de 2024** con el lanzamiento de FIPS 203, 204, y 205, cubriendo algoritmos como Kyber, Dilithium, y SPHINCS+)

8.4 Por Qué las Criptomonedas Pueden Adaptarse

Las criptomonedas son software. El software puede actualizarse.

Bitcoin usa ECDSA para firmas, pero no hay nada que impida un cambio a algoritmos resistentes a la cuántica. El protocolo central—bloques, transacciones, consenso—permanece igual; solo cambia el algoritmo de firma.

Ejemplo de migración:

Actual: Clave Privada → (ECDSA) → Clave Pública → Dirección de Bitcoin

Futuro: Clave Privada → (Algoritmo post-cuántico) → Clave Pública → Nueva Dirección de Bitcoin

Los usuarios generarían nuevos monederos resistentes a la cuántica y transferirían sus fondos. La blockchain continúa, solo que con nuevos algoritmos de firma.

8.5 Todos Tienen Este Problema

Si los ordenadores cuánticos rompen el cifrado de criptomonedas, rompen TODO: - Banca en línea → rota - Comunicaciones militares → rotas - Secretos gubernamentales → rotos - Cada sitio web HTTPS → roto

Las criptomonedas no son únicamente vulnerables. Enfrentan la misma amenaza cuántica que todos los demás—y en realidad son MÁS adaptables porque los protocolos cripto pueden actualizarse vía consenso, mientras que los sistemas tradicionales son notoriamente lentos para cambiar.

El mundo entero tiene incentivo para desarrollar criptografía resistente a la cuántica ANTES de que existan ordenadores cuánticos poderosos. Y ese trabajo ya está bien en marcha.

8.6 La Búsqueda del Tesoro: El Bitcoin de Satoshi

Aquí hay una consecuencia fascinante:

Satoshi Nakamoto, el creador seudónimo de Bitcoin, minó entre **750,000 y 1.1 millones de Bitcoin** en los primeros días (la mayoría de los análisis estiman alrededor de **1 millón de BTC**)—hoy vale decenas de miles de millones de dólares.

Cuando gastas Bitcoin, revelas tu clave pública en la blockchain. Satoshi hizo algunas transacciones, revelando algunas claves públicas.

Si alguien construye un ordenador cuántico lo suficientemente poderoso para romper ECDSA, podría derivar claves privadas desde esas claves públicas reveladas y robar el Bitcoin. Esto crea dos resultados posibles:

1. Quien rompa ECDSA primero encuentra el “tesoro” (miles de millones de dólares en Bitcoin temprano)
2. La red acuerda mover todos esos fondos a una nueva dirección resistente a la cuántica controlada por nadie

Esto no es solo teórico—crea un incentivo real para resolver las amenazas cuánticas antes de que lleguen. Con miles de millones de dólares en juego, la gente toma la amenaza en serio y migrará a tiempo.

Idea Clave: Los ordenadores cuánticos representan una amenaza futura para algunos algoritmos criptográficos, incluyendo aquellos usados por las criptomonedas. Sin embargo, los algoritmos resistentes a la cuántica ya existen, y la migración es posible. Todo el internet enfrenta este mismo desafío, creando fuertes incentivos para soluciones oportunas. Esta es una transición manejable, no una catástrofe.

Ahora que entiendes las herramientas criptográficas—cómo crear IDs digitales, probar propiedad, y firmar transacciones—exploremos qué viene después: **¿Podemos hacer que los bits signifiquen “dinero”?** En el próximo capítulo, veremos cómo los bits pueden tener las propiedades del dinero, por qué necesitamos una base de datos distribuida para almacenarlos, y el desafío fundamental que hace posibles las criptomonedas: el problema del consenso.

9

Capítulo 9: Dinero como Bits Sincronizados

Si podemos asignar significado a los bits, ¿por qué no asignarles el significado de dinero?

Ahora entiendes que la criptografía te permite crear IDs digitales desde tu casa y probar tu intención con firmas. Sabes que la información son solo bits, y los humanos asignamos significado a esos bits.

Aquí está la pregunta natural: **¿Podemos hacer que los bits signifiquen “dinero”?**

La respuesta es sí. Pero primero, entendamos qué necesita ser realmente el “dinero”.

9.1 ¿Qué Hace que Algo Sea Dinero?

A lo largo de la historia, los humanos han usado todo tipo de cosas como dinero: sal (tan valiosa que la palabra “salario” viene de ella), conchas (usadas a través de continentes durante milenios), oro y plata (pesados, pero universalmente valorados), billetes de papel (convenientes, pero solo valiosos porque estamos de acuerdo, y los impuestos), y números de cuentas bancarias digitales (solo entradas en una base de datos).

¿Qué tienen todos estos en común? Comparten ciertas propiedades que los hacen útiles como dinero.

9.1.1 Propiedad 1: Escasez (No Puedes Crearlo Fácilmente)

Si tengo 10 unidades de dinero, no puedo tener repentinamente 11 sin ganarlo, encontrarlo con esfuerzo, o que alguien me lo dé.

Ejemplos: - Oro: Difícil de minar, no puedes simplemente crear más. - Billetes de dólar: El gobierno controla la impresión (no puedes fotocopiar y usarlo). - Sal (históricamente): Requería esfuerzo extraerla del agua del mar.

Contraejemplo: - Hojas: Demasiado abundantes, todos serían “ricos”. - Si alguien pudiera crear dinero libremente, no tendría valor.

Hay más razones filosóficas y sociológicas para justificar que para que algo sea valioso necesita ser escaso. Este libro no trata tanto sobre coordinación social sino más bien sobre la explicación de cómo se pueden usar nuevas tecnologías para ella. Por lo tanto, aquí dejo una intuición y argumento simple explicando por qué el valor debe tener algún grado de escasez:

Para que algo sea valioso necesita ser difícil de crear y escaso. El dinero es la abstracción del valor para que podamos coordinar el comercio. Si alguien pudiera simplemente recoger 100 hojas y decir “Ahora tengo poder sobre lo que tienes que hacer por mí,” el sistema se desmorona. Simplemente

recogeríamos hojas sin parar. Para recompensar el trabajo que realmente crea valor, no podemos simplemente dar valor a cualquier cosa que exista en abundancia accesible.

Para que los bits sean dinero: Necesitamos reglas que prevengan la creación descontrolada. El sistema debe hacer cumplir la escasez difícil de controlar.

9.1.2 Propiedad 2: Verificabilidad (Puedes Probar Lo Que Tienes)

Si afirmo tener 10 unidades, necesito poder probártelo, y tú necesitas poder verificar que es real.

Ejemplos: - Oro: Puedes pesarlo, probar su pureza. - Billetes de dólar: Características de seguridad (marcas de agua, papel especial). - Cuenta bancaria: Puedes comprobar tu saldo, mostrar un extracto.

Contraejemplo: - Si simplemente digo “Tengo 10 unidades de valor,” pero no puedo probarlo, no lo aceptarás.

Para que los bits sean dinero: Necesitamos una manera de comprobar los saldos. Algún registro compartido que todos puedan verificar, consultar, y preferiblemente, instantáneamente.

9.1.3 Propiedad 3: Sin Doble Gasto (No Puedes Gastar el Mismo Dinero Dos Veces)

Si tengo 10 unidades y te las doy, ya no las tengo. No puedo dar también esas mismas 10 unidades a otra persona.

Ejemplos: - Efectivo físico: Te doy un billete de 10€, y ya no lo tengo. - Oro: Te doy la moneda de oro, y ahora está físicamente contigo. - Transferencia bancaria: El banco deduce de mi cuenta, añade a la tuya.

Contraejemplo: - Si pudiera gastar los mismos 10€ contigo Y con Bob, el dinero se rompe. - Esta es la parte complicada con los archivos digitales (se pueden copiar infinitamente).

Para que los bits sean dinero: Cuando te los transfiero, esos bits específicos ya no deben ser “míos.” El sistema debe rastrear la propiedad claramente.

9.1.4 Propiedad 4: Transferibilidad (Puedes Enviarlo)

El dinero es inútil si no puedes darlo a otros a cambio de bienes, servicios, u otro dinero.

Ejemplos: - Efectivo: Lo entregas, transferencia inmediata. - Transferencia bancaria: Lo envías electrónicamente, a través del internet que ahora entendemos. - Oro: Puedes transportarlo (aunque pesado).

Para que los bits sean dinero: Necesitamos un mecanismo para cambiar la propiedad. Una forma de decir “estos bits eran míos, ahora son tuyos.”

9.1.5 Propiedad 5: Propiedad (Realmente Lo Controlas)

Necesitas poseer verdaderamente tu dinero. No solo “tener permiso para usarlo” de otra persona.

Ejemplos: - Efectivo en tu bolsillo: TÚ lo controlas físicamente. - Oro que estás sosteniendo: TÚ lo posees. - Cuenta bancaria: Bueno... el banco realmente lo controla (solo tienes permiso).

El problema de la cuenta bancaria: No posees el dinero; el banco sí. Confías en que te lo devuelvan cuando lo pidas. Pueden congelar tu cuenta, negar el acceso, o incluso quebrar. Tu “dinero” es solo su promesa hacia ti.

Para que los bits sean dinero: Idealmente, TÚ los controlas directamente. Como efectivo en tu bolsillo, pero digital.

9.1.6 Propiedad 6: Fungibilidad (Cada Unidad es Equivalente)

Una unidad de dinero debería ser igual que cualquier otra unidad. Son intercambiables.

Ejemplos: - Billetes de dólar: Cualquier billete de 10\$ = cualquier otro billete de 10\$. - Oro: Una onza de oro puro = cualquier otra onza de oro puro (por peso y pureza).

Contraejemplo: - Coleccionables únicos: Cada uno es diferente, no intercambiable.

Para que los bits sean dinero: 1 unidad de bit = 1 unidad de bit. No importa qué bits específicos tengas.

9.1.7 Propiedad 7: Divisibilidad (Se Puede Dividir en Partes Más Pequeñas)

A veces necesitas pagar menos de 1 unidad completa. El dinero debería ser divisible.

Ejemplos: - Dólar: Se puede dividir en centavos (\$0.01). - Oro: Se puede cortar o fundir en cantidades más pequeñas. - Bitcoin: Divisible hasta 8 decimales (0.00000001 BTC, llamado un “satoshi”).

Para que los bits sean dinero: Debería soportar fracciones. No solo números enteros.

9.1.8 Propiedad 8: Durabilidad (No Desaparece ni se Descompone)

El dinero debería durar. Si se degrada rápidamente, no es útil para almacenar valor.

Ejemplos: - Oro: No se oxida ni descompone (dura para siempre). - Monedas: El metal perdura durante décadas. - Billetes de papel: Se desgastan, pero duran unos años.

Contraejemplo: - Comida: Se pudre, no puede usarse como dinero.

Para que los bits sean dinero: La información digital no se descompone (se puede copiar perfectamente), pero necesitas asegurarte de que no se pierda. Si pierdes el acceso (pierdes tu contraseña/claves), se ha ido.

9.2 La Realización: Los Bits Pueden Tener Estas Propiedades

Mira lo que hemos aprendido hasta ahora:

Del Capítulo 2 (Logos): Los humanos asignamos significado a patrones de bits. Hemos asignado significado a bits como letras (ASCII), bits como imágenes (JPEG), y bits como música (MP3). ¿Por qué no bits como dinero?

Del Capítulo 7 (Criptografía Asimétrica): Podemos crear IDs digitales (claves públicas/privadas) y probar propiedad con firmas. Esto resuelve: - **Propiedad:** Tu clave privada = tu control (como efectivo en tu bolsillo, digitalmente). - **Transferibilidad:** Firma un mensaje diciendo “Envío X a Bob” y prueba la intención.

¿Qué falta? La parte de coordinación.

9.3 El Problema de la Base de Datos

Piénsalo simplemente: si los bits representan dinero, ¿dónde los almacena?

Los bits son información, y la información necesita almacenamiento. La información se almacena en bases de datos, que son en última instancia esos dispositivos físicos llamados transistores que tienen 2 estados posibles, cada uno representando un bit, como una puerta. Pero como dijimos, las puertas son transistores muy lentos y torpes. Es mejor simplemente usar ordenadores.

Así que necesitamos una base de datos—solo un ordenador almacenando información en bits—que rastree quién tiene cuánto dinero (saldos) y quién envió dinero a quién (transacciones).

Ejemplo simple:

Base de datos:

- Alice: 50 monedas
- Bob: 30 monedas
- Carol: 20 monedas

Transacción:

- Alice envía 10 monedas a Bob

Nuevo dato (estado) de la base de datos:

- Alice: 40 monedas
- Bob: 40 monedas
- Carol: 20 monedas

Fácil, ¿verdad?

9.4 La Solución Tradicional: Una Base de Datos, Un Controlador

Así es como funcionan los bancos:

La base de datos del banco: - Almacena el saldo de todos. - El banco controla la base de datos. - Cuando “envías dinero,” le pides al banco que actualice la base de datos. - El banco verifica: ¿Tiene Alice 10 monedas? ¡Sí! Vale, actualizar: - Restar 10 de Alice. - Añadir 10 a Bob. - ¡Hecho!

Esto funciona. Un punto central para mirar, para interactuar, centralizado. Pero nota el problema:

Tú no controlas el dinero. El banco sí.

El banco puede congelar tu cuenta, negar tu transacción, o quebrar (tu dinero desaparece). Debes confiar en el banco para mantener registros precisos y darte acceso cuando lo necesites.

¿Recuerdas la **Propiedad 5 (Propiedad)**? No posees verdaderamente tu dinero. Tienes un saldo en la base de datos de otra persona.

9.5 La Idea: Distribuir la Base de Datos

Aquí está la idea: **¿Qué pasa si le damos a TODOS una copia de la base de datos?**

En lugar de un banco teniendo los registros, Alice tiene una copia de la base de datos, Bob tiene una copia, Carol tiene una copia, y miles de otras personas tienen copias también.

Ahora: - Ninguna persona individual la controla (des-centralizado). - Alice puede verificar el saldo de Bob ella misma (solo comprobar su copia). - Nadie puede cambiar secretamente los registros (todos lo notarían).

Esto resuelve algunos problemas: - **Verificabilidad:** Cualquiera puede comprobar el saldo de cualquiera. - **Propiedad:** Ninguna entidad individual controla el acceso, solo tú y tus claves privadas. - **Transparencia:** Todas las transacciones son visibles.

Pero esto todavía tiene problemas... Por ejemplo, no resuelve el problema del doble gasto por sí solo. Aún no tiene la propiedad 3.

9.6 El Problema del Consenso

Aquí es donde se pone complicado.

Si todos tienen una copia de la base de datos, ¿cómo aseguramos que todas las copias permanezcan sincronizadas a medida que los datos cambian con el tiempo? Como los bits en nuestra base de datos están siendo interpretados como dinero, podemos reformular esto: ¿cómo aseguramos que todas las copias permanezcan sincronizadas a medida que el dinero se transfiere entre personas?

Problema 1: Actualizaciones conflictivas

Alice envía 10 monedas a Bob (transacción firmada)

Bob la recibe, actualiza su base de datos: Alice: 10, Bob: 40

Al mismo tiempo, Carol recibe una transacción diferente:

Alice envía 10 monedas a Carol (también firmada por Alice)

Base de datos de Bob: Alice: 0, Bob: 40, Carol: 20

Base de datos de Carol: Alice: 0, Bob: 30, Carol: 30

¿Cuál es correcta? Ambas tienen firmas válidas, y en ambas Alice tenía 10 monedas para gastar

¿Cómo prevenimos esto? ¿Quién decide qué transacción es válida? Ambas no pueden ser válidas, porque Alice estaría gastando doblemente sus 10 monedas.

Puedes decir, envíala primero a Bob y luego a Carol, pero ¿qué pasa si los mensajes llegan al mismo tiempo? Recuerda que estamos en internet—es como el salvaje oeste digital, sin autoridad central para decidir quién puede escribir primero o recibir el mensaje primero.

Bueno, entonces ¿creamos un consenso de siempre enviar los mensajes en ese orden y no tendremos problemas? Podría funcionar para 3 personas que confían entre sí, pero para miles de personas que no deberían necesitar confiar entre sí, ¿cómo aseguramos que todos estén de acuerdo en el mismo orden de transacciones?

1000 ordenadores todos tienen la base de datos

Alice firma: "Envío 10 a Bob"

¿Quién puede añadir esto a la base de datos?

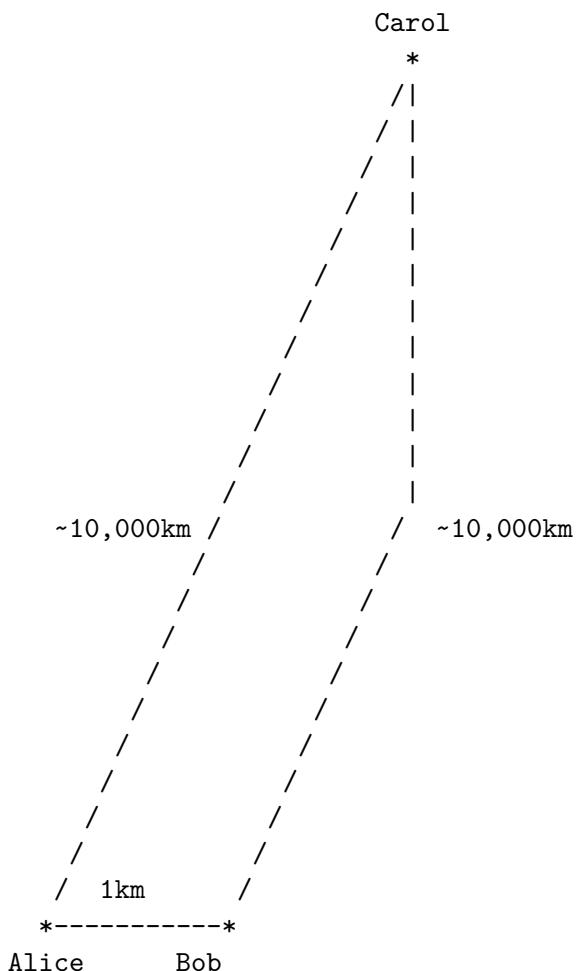
¿Añaden los 1000 ordenadores al exactamente mismo tiempo?

;Qué pasa si alguien añade transacciones falsas? ;Qué pasa si Alice intenta confundir a la red

Problema 2: Latencia (Retrasos por Distancia)

Incluso si todos son honestos, la distancia física crea problemas de sincronización.

Imagina que Alice vive a 1km de Bob, y Carol vive a 10,000km de ambos. Carol le pagará a Alice 5 monedas, y Alice quiere comprar algo de Bob por 10 monedas. Alice solo tiene 8 monedas, así que no puede comprarlo todavía. Pero entonces Carol le envía 5 monedas, dando a Alice suficiente para comprarle a Bob.



Alice intenta hacer transacción con Bob, pero la información—incluso si viaja a la velocidad de la electricidad—tarda tiempo en cruzar el mundo entero, y podría haber retrasos. Bob podría rechazar la transacción de Alice porque en su base de datos Alice solo tiene 8 monedas. Pero en la base de datos de Alice, porque ella tiene una mejor conexión de internet, la transacción de Carol ya llegó y ella cree que ya tiene $8+5=13$ monedas.

¿Quién tiene razón? Ambos son honestos, ambos están siguiendo las reglas, pero sus bases de datos están temporalmente desincronizadas debido a la latencia.

Como puedes ver, ya sea por malicia e intentar gastar doble, o simplemente por la naturaleza de coordinar físicamente información a través de distancias, distribuir una base de datos compartida es un problema muy difícil.

Estas son las preguntas fundamentales:

9.7 Las Preguntas Que Necesitamos Responder

1. **¿Cómo te aseguras de que TODOS los ordenadores guarden los mismos saldos?**
 - Si todos tienen una copia, todos necesitan estar de acuerdo.
 - ¿Cómo sincronizamos miles de bases de datos?
2. **¿Cómo te aseguras de que nadie transfiera o cree dinero sin permiso?**
 - Tenemos firmas (bien), pero ¿cómo haces cumplir que los participantes de la red las verifiquen?
 - ¿Qué pasa si alguien intenta gastar dinero que no tiene?
 - ¿Qué pasa si alguien intenta gastar el mismo dinero dos veces?
3. **¿Cómo alcanzas consenso sobre cómo y con qué límites escribir en esta base de datos compartida?**
 - ¿Quién puede añadir nuevas transacciones?
 - ¿Con qué frecuencia realizamos actualizaciones?
 - ¿Qué reglas deben seguir todos?
 - ¿Cómo hacemos cumplir esas reglas sin una autoridad central?

Este es el problema del consenso.

Y resolver este problema—permitir que miles de extraños mantengan una base de datos idéntica sin confiar en ninguna autoridad única—es lo que hace posible Bitcoin y otras criptomonedas. Es, como puedes ver, un problema muy difícil.

9.8 Hablando Históricamente

Durante miles de años, los humanos usaron objetos físicos como dinero. La sal era valiosa porque preservaba la comida (esencial para la supervivencia). Las conchas eran raras, hermosas, y difíciles de falsificar. El oro era escaso, no se descomponía, tenía valor universalmente acordado, y era divisible. El papel era conveniente pero requería confiar en el emisor (gobierno y banco). Cada era encontró algo que funcionaba para la tecnología y los modelos de confianza de la época.

El dinero físico tenía escasez incorporada (difícil de crear) y propiedad (posesión = propiedad).

El dinero digital con bancos funcionó confiando en una autoridad central para mantener la contabilidad y mantener el valor del dinero estable.

Pero bits sincronizados de forma consensuada—distribuidos a través de miles de ordenadores, sin autoridad central, donde TÚ controlas tu dinero con claves criptográficas—esto es nuevo.

Solo es posible porque Satoshi (quienquiera que sea él, ella, o ellos) introdujo el primer **mecanismo de consenso descentralizado y práctico para dinero digital**—a menudo llamado *consenso Nakamoto*. Esto resolvió el problema del doble gasto en una red abierta combinando proof-of-work, selección de cadena más larga, e incentivos económicos. No es la solución final para toda la

investigación de consenso (ese es un campo más amplio con muchas variantes), pero fue el avance que hizo funcionar Bitcoin.

Idea Clave: El dinero es solo información que acordamos que tiene valor. Los bits pueden representar dinero si tienen las propiedades correctas: escasez, verificabilidad, sin doble gasto, transferibilidad, propiedad, fungibilidad, divisibilidad, y durabilidad. Podemos almacenar esta información en una base de datos y distribuir copias a miles de ordenadores. Pero para que funcione, necesitamos resolver un problema difícil: ¿Cómo acuerdan miles de extraños el mismo estado de base de datos (datos) sin confiar en ninguna autoridad central? Ese es el problema del consenso—y eso es lo que exploraremos a continuación.

Por cierto, cuando digo estado significa lo mismo que los datos actuales en una base de datos, la información actual que la base de datos contiene.

A continuación, veremos cómo los **mecanismos de consenso** resuelven estos problemas. Entenderemos Proof-of-Work, mining (minería), y por qué la estructura blockchain hace que la historia sea evidente de manipular. Las piezas están uniéndose.

10

Capítulo 10: Proof-of-Work - Ganándote el Derecho a Escribir

¿Y si tomáramos prestada la jerarquía temporalmente, pero ganáramos el liderazgo a través del esfuerzo?

Hemos establecido el problema: distribuir una base de datos a través de miles de ordenadores crea caos de sincronización. Actualizaciones conflictivas, retrasos de latencia, y actores maliciosos pueden amenazar con romper el sistema.

Aquí hay un pensamiento natural: **¿Y si temporalmente le diéramos a alguien el poder de decidir?**

10.1 Tomando Prestada la Jerarquía (Solo por un Momento)

Piénsalo: muchos de los problemas que identificamos en el Capítulo 9 se vuelven más fáciles si una persona tiene autoridad:

Problema 1 (Actualizaciones conflictivas): Si Alice envía 10 monedas a Bob Y a Carol al mismo tiempo, ¿quién decide qué transacción es válida? Fácil—la persona a cargo decide. Eligen una, rechazan la otra. Hecho.

Problema 2 (Retrasos de latencia): Si las bases de datos están temporalmente desincronizadas porque la información tarda tiempo en viajar, ¿quién decide cuál es el estado “correcto”? La persona a cargo espera un segundo o dos y declara: “Esta es la versión oficial.” Todos actualizan para coincidir.

Problema 3 (Seguridad): Si Dave intenta inundar la red con transacciones falsas o enviar datos conflictivos a diferentes personas, ¿quién lo detiene? La persona a cargo valida transacciones y solo incluye las legítimas.

La centralización hace la coordinación mucho más simple—por eso los bancos funcionan tan eficientemente. Una autoridad, una base de datos, una fuente de verdad.

Pero no queremos centralización permanente. Eso nos lleva de vuelta al problema del banco—falta de propiedad, cuentas congeladas, control por una sola entidad.

La idea: ¿Y si le damos a alguien autoridad temporal para escribir en la base de datos, pero: - Solo por un período corto (no para siempre como los bancos). - Deben seguir reglas (como “no gastar doble”). - Rotamos quién obtiene este poder (no la misma persona cada vez).

Este es el compromiso elegante: Toma prestada la jerarquía temporalmente, pero distribúyela a lo largo del tiempo.

10.2 ¿Quién Obtiene el Derecho a Escribir?

Vale, entonces le daremos a alguien poder temporal para añadir transacciones, para escribir nueva información a la base de datos. ¿Pero a quién?

No podemos simplemente votar. Los IDs aquí se crean desde casa—Dave podría ejecutar 500 identidades falsas y ganarle a todos en votación. Un ordenador, un voto no funciona cuando cualquiera puede pretender ser 500 ordenadores. Además, no nos conocemos entre nosotros, y no necesitamos hacerlo. ¿Cómo vas a votar por alguien digno de confianza si no lo conoces?

No podemos simplemente elegir al azar. Mismo problema—si Dave ejecuta 500 ordenadores en una red de 700 ordenadores, tiene una probabilidad de 500/700 de ser elegido comparado con 200 participantes honestos. Él gana la mayoría del tiempo. ¿Y si es malvado? ¿El mal gana la mayoría del tiempo? Eso, por definición, no es bueno.

Necesitamos un mecanismo de selección diferente.

10.3 Cómo los Humanos Eligen Líderes

A lo largo de la historia, los humanos han elegido líderes basándose en **hazañas**: - El héroe valiente que defendió la aldea (coraje demostrado a través de la acción). - El doctor superinteligente que inventó una cura (experiencia demostrada a través del logro). - El artesano habilidoso que construyó las mejores herramientas (competencia demostrada a través del trabajo).

Las hazañas requieren: 1. **Habilidad** - Necesitas saber cómo hacer algo. 2. **Esfuerzo** - Necesitas realmente hacer el trabajo. 3. **Prueba** - Otros pueden verificar que lo lograste.

No puedes falsificar una hazaña fácilmente. No puedes afirmar ser un héroe sin luchar. No puedes afirmar haber inventado una cura sin mostrar la medicina funcionando. La hazaña en sí es la prueba.

¿Y si usáramos este mismo principio natural para elegir quién puede escribir en la base de datos?

Requerir una hazaña—algo que requiere esfuerzo, puede verificarse, pero no puede falsificarse fácilmente.

10.4 La Primera Hazaña Digital: Proof-of-Work

En el mundo físico, las hazañas involucran coraje, inteligencia, o artesanía. En el mundo digital, ¿cuál es el equivalente?

Computación, ejecutada por electricidad, energía.

Los ordenadores hacen trabajo computando—ejecutando cálculos, procesando datos, resolviendo problemas matemáticos. Algunos cálculos son fáciles (sumar 2+2), mientras otros son muy difíciles (encontrar un patrón específico en miles de millones de números aleatorios).

Aquí está la idea: Para ganarte el derecho de escribir el próximo lote de transacciones en la base de datos, debes resolver un rompecabezas computacional difícil.

No un rompecabezas que requiere inteligencia (eso favorecería injustamente a gente inteligente, y es muy difícil de diseñar). Un rompecabezas que requiere **fuerza bruta** funcionaría mejor—probar miles de millones de intentos hasta que encuentres la respuesta correcta. Esto consume electricidad real, electricidad que podrías haber usado para algo más, como calentar tu casa o ejecutar tu negocio. Así que, si realmente quieres usar esa electricidad para ganarte el derecho de escribir en la base de datos, tienes que hacer un sacrificio. ¿Estará la gente de acuerdo en que este sacrificio vale la pena? Resulta que sí lo estuvieron, y todavía lo están.

Pero si no lo estuvieran, incluso si estuvieras siguiendo las reglas y resolviendo el rompecabezas acordado, todo tu esfuerzo y electricidad habría sido desperdiciado. Tú, el trabajador y potencial futuro líder, y los demás tienen una dependencia mutua entre sí. Yo hago el trabajo, y confío en que ustedes lo valorarán. Y viceversa, has demostrado hacer el trabajo, y te recompenso dejándote escribir en la base de datos y yo usándola. Pero recuerda, mañana podría ser otra persona, incluso una malvada, así que mejor sigues las reglas y continúas trabajando.

No solo los usuarios de la base de datos dependen de los mineros, sino que los mineros dependen de que los usuarios encuentren útil la base de datos también. Brillantes dinámicas de teoría de juegos en juego aquí.

Los mineros son solo un nombre especial puesto a los ordenadores en la red que están intentando resolver estos rompecabezas para ganar el derecho de escribir las próximas transacciones en la base de datos.

El algoritmo de Bitcoin que muestra su interpretación del esfuerzo se llama: Proof-of-Work.

10.5 El Rompecabezas

Déjame explicar el rompecabezas que Bitcoin usa (simplificado):

El objetivo: Encontrar un número que, cuando se combina con los datos de transacción y se pone a través de una función hash, produce una salida que comienza con un cierto número de ceros.

¿Qué es una función hash? ¿Recuerdas las funciones unidireccionales del Capítulo 7? Una función hash es similar—toma cualquier entrada y produce una salida de tamaño fijo que parece aleatoria. Cambia la entrada incluso ligeramente, y la salida cambia completamente.

Ejemplo (simplificado):

```
hash("Hola") = d3a1f2
hash("Hola!") = 9f4e7b (completamente diferente, y... ¡también único! si es suficientemente largo)
```

La función hash es una función unidireccional que también es determinista (la misma entrada siempre da la misma salida), pero impredecible (no puedes predecir la salida sin calcularla).

Porque las funciones hash son impredecibles y producen salidas de suficiente longitud, puedes usarlas para generar IDs—la probabilidad de que dos entradas diferentes produzcan la misma salida es astronómicamente baja. Diseñar una buena función hash no es trivial, pero para nuestros propósitos, asumiremos que la función hash de Bitcoin (SHA-256) es suficientemente buena.

El rompecabezas:

Entrada: [Transacciones + Bloque Anterior + Número Aleatorio Que Elegiste]

Salida: Hash que comienza con, digamos, 5 ceros (00000...)

Intenta:

```
hash([Transacciones + Bloque Anterior + 0]) = 3f2a8c... (no, no comienza con 00000)
hash([Transacciones + Bloque Anterior + 1]) = 7d1b5e... (no)
hash([Transacciones + Bloque Anterior + 2]) = 5c8f2a... (no)
...
hash([Transacciones + Bloque Anterior + 8,347,291]) = 00000f8a2c... (¡SÍ! ¡Lo encontré!)
```

El único algoritmo para resolver esto es... fuerza bruta. Tu única opción es intentar miles de millones o billones de números aleatorios hasta que encuentres uno que produzca un hash que comience con el número requerido de ceros.

Esta es la hazaña. Requiere: - **Esfuerzo:** Intentar miles de millones requiere trabajo computacional masivo (electricidad, tiempo, hardware). - **Prueba:** Una vez que encuentras la respuesta, cualquiera puede verificarla instantáneamente (solo ejecuta la función hash una vez con tu número). - **No puede falsificarse:** No puedes pretender que hiciste el trabajo sin hacerlo realmente, porque la función hash es unidireccional. No puedes simplemente tener un valor con 5 ceros iniciales e intentar adivinar la entrada que lo creó.

10.6 Por Qué Esto Funciona

Veamos cómo Proof-of-Work resuelve nuestros problemas:

1. ¿Quién puede escribir? Quien resuelva el rompecabezas primero. Se ganó el derecho a través del esfuerzo.

2. ¿Puede Dave engañar con 500 identidades falsas? No. Ejecutar 500 ordenadores no importa—lo que importa es el poder computacional. Si Dave tiene 30% del poder computacional total, gana aproximadamente 30% de los rompecabezas, mientras que los participantes honestos con 70% del poder computacional ganan aproximadamente 70% del tiempo. Es proporcional al esfuerzo, no al número de identidades.

3. ¿Cómo sabemos que siguieron las reglas? Cuando alguien publica su lote de transacciones (llamado un “bloque”), todos los demás verifican: - ¿Resolvieron el rompecabezas? (Verificar la validez del hash.) - ¿Son válidas todas las transacciones? (Verificar firmas, verificar saldos.) - ¿Sin gastar doble? (Verificar contra la base de datos.)

Si algo es inválido, todos rechazan el bloque, todas las nuevas transacciones, y no actualizan (sincronizan) su base de datos. El trámoso desperdició todo ese esfuerzo computacional para nada.

4. ¿Con qué frecuencia sucede esto? Bitcoin ajusta un **valor hash objetivo** para que se encuentre un nuevo bloque aproximadamente cada 10 minutos. En pantalla, esto a menudo se ve como requerir hashes con **más ceros iniciales**. Si más personas se unen y el poder computacional aumenta, el objetivo se vuelve más bajo (más difícil de cumplir). Si las personas se van y el poder computacional disminuye, el objetivo se vuelve más alto (más fácil de cumplir).

La razón por la que puedes “fijar” el tiempo que toma minar (crear) un bloque es porque puedes hacer algunas estadísticas para calcular el tiempo esperado para encontrar una solución basándose en el poder computacional total de la red. Al ajustar la dificultad del rompecabezas cada 2016

bloques (aproximadamente dos semanas), Bitcoin asegura que, en promedio, se encuentre un nuevo bloque cada 10 minutos. Si te interesa la matemática detrás de esto, te animo a investigar en línea y profundizar por tu cuenta.

Por ahora, desmitifiquemos algo de terminología:

- **Mining (Minería):** El proceso de intentar resolver el rompecabezas (ejecutar cálculos). Una prueba de esfuerzo para ganar el derecho de escribir el próximo lote de datos en la base de datos distribuida que todos comparten en la red.
- **Miner (Minero):** Un participante que ejecuta ordenadores para minar (resolver rompecabezas).
- **Block (Bloque):** Un lote de transacciones más la solución al rompecabezas. Que, hablando en general, son un conjunto de instrucciones sobre qué escribir a continuación en la base de datos.
- **Node (Nodo):** Un participante que mantiene una copia de la base de datos y verifica bloques (no necesariamente minando). Pueden aceptar o rechazar bloques basándose en las reglas que se han acordado. Si el minero comparte bloques inválidos, los nodos los rechazarán, y todos simplemente tendrán el mismo estado de base de datos. ¿Válido? Todos actualizan la base de datos añadiendo las transacciones en el orden establecido por el líder temporal y así todos terminan con el mismo estado en la base de datos.

10.7 El Incentivo: Recompensas de Minería

Espera—¿por qué alguien gastaría electricidad y dinero en ordenadores caros para resolver estos rompecabezas “inútiles”?

Porque hay una recompensa.

Quien resuelva el rompecabezas puede:

1. **Crear nuevas monedas** - Actualmente 3.125 Bitcoin por bloque (esto se reduce a la mitad cada 4 años, eventualmente alcanzando cero alrededor del año 2140, si el consenso no cambia).
2. **Recolectar comisiones de transacción** - Los usuarios pueden adjuntar pequeñas comisiones a sus transacciones. El minero que las incluye en un bloque recolecta todas esas comisiones.

Esto se llama mining (minería). Como los mineros de oro gastando esfuerzo para extraer oro de la tierra, los mineros de Bitcoin gastan esfuerzo computacional para ganar Bitcoin. Y, como Bitcoin comparte un buen montón de propiedades que tiene el oro, supongo que por eso Satoshi Nakamoto lo llamó “mining” (minería). Pero esto es solo especulación del autor.

El incentivo se alinea perfectamente:

- Los mineros quieren recompensas, así que siguen las reglas (los bloques inválidos son rechazados, desperdiциando su esfuerzo).
- Los mineros compiten, así que invierten en más poder computacional (asegurando la red).
- A medida que el valor de Bitcoin aumenta, más mineros se unen, lo que significa más seguridad.

Teoría de juegos en acción: Es más rentable jugar honestamente y mantener la red funcionando que hacer trampa.

10.8 El Milagro de Bitcoin, los Sueños Húmedos de los Frikis, Tal Vez

Pero espera... les pagan... ¿en qué? ¿En bits? En efecto. ¡En bitcoins! Pero a nadie le importaban estos bits inicialmente. En efecto, aquí es donde “el milagro” sucedió.

Un grupo de personas, probablemente “frikis” de ciencias de la computación y anarquistas (dicho

con amor por el autor, que es un friki él mismo), decidieron simplemente gastar su electricidad y poder computacional en este “experimento loco” de una moneda digital descentralizada. Creían en la idea, o simplemente eran frikis divirtiéndose, y estaban dispuestos a invertir recursos en ello.

Algunas personas especulan que fue la CIA o cualquier otra agencia del gobierno de Estados Unidos intentando crear dinero programable para controlar a la población. Pero como se dijo, Satoshi Nakamoto desapareció de la vista pública—su **última publicación pública en un foro fue en diciembre de 2010**, y su **último correo privado conocido fue en abril de 2011**—y nunca se ha encontrado evidencia concreta para apoyar ninguna teoría sobre su identidad o afiliación.

El autor se inclina hacia los frikis jugando—¿tal vez frikis de la CIA en sus fines de semana? Quién sabe. La cosa es que la fiesta se salió de control, y lenta pero seguramente, más personas estuvieron de acuerdo con esta coordinación y consenso digital y comenzaron a intercambiar realmente estos bits con dinero fiat. Y... ¡boom! Nació una nueva clase de activo.

Es bastante poético que la cosa que es valiosa precisamente porque la gente puede usarla sin siquiera conocerse fue creada por alguien que nadie conoce—en la práctica entonces, por nadie. Por lo tanto, poéticamente, podemos atrevernos a decir que fue creado por nadie, y por todos nosotros, al mismo tiempo.

¡Pero espera! Segundo todo lo que hemos explicado, todavía podemos engañar a la red. Nota que no dijimos la palabra “blockchain” en absoluto...

10.9 Liderazgo Temporal

Nota lo que hemos logrado:

- **Cada 10 minutos**, alguien gana el derecho de escribir el próximo lote de transacciones en la base de datos.
- **Tienen autoridad temporal** - Pero solo para ese bloque. Luego la carrera comienza de nuevo.
- **El liderazgo rota** - Quien resuelva el próximo rompecabezas se convierte en el próximo líder temporal.
- **Sin poder permanente** - Ninguna entidad individual controla la base de datos para siempre... ¿o sí? Si alguien tiene 51% de todo el poder computacional, simplemente será más probable que escriba más bloques, pero aún tiene que seguir las reglas o todos los demás rechazarán sus bloques.

Hemos tomado prestada la jerarquía temporalmente, la distribuimos a lo largo del tiempo, y usamos el esfuerzo como mecanismo de selección.

Esto es radicalmente diferente de los bancos: - Bancos: Una entidad tiene control permanente.
- Bitcoin: Control temporal, ganado a través de trabajo computacional, rotando entre miles de participantes.

10.10 Qué Hemos Resuelto (¿Lo Hemos Hecho?) y Qué No

Lo que Proof-of-Work resuelve: - Quién puede escribir: Quien resuelva el rompecabezas primero.
- Ataques Sybil: Necesitas poder computacional, no identidades falsas. - Ordenar transacciones: El

ganador decide el orden en su bloque. - Incentivos: Los mineros son recompensados por trabajo honesto. Los ataques Sybil ya no tienen sentido para agentes dentro de la red.

Lo que todavía necesitamos abordar: - ¿Cómo estamos explicando blockchain sin la estructura blockchain misma? ¿Por qué la necesitamos? (Próximo capítulo.) - ¿Qué pasa si dos mineros resuelven el rompecabezas al mismo tiempo? ¿Qué pasa si esto sucede a menudo? - ¿Cómo hace esto que la historia sea evidente de manipular? - ¿Qué pasa si un minero controla más del 50% del poder computacional? ¿Gana demasiadas oportunidades de tener el derecho de escribir en la base de datos?

10.11 El Panorama General

Proof-of-Work es la forma de Bitcoin de seleccionar líderes temporales basándose en el esfuerzo. No es el único mecanismo de consenso, o algoritmo de consenso si prefieres. Ethereum, otra “cripto,” ahora usa Proof-of-Stake, que exploraremos después. Pero Bitcoin fue el primero en resolver exitosamente el problema de la red de base de datos sincronizada distribuida sin intermediarios de confianza, usando consenso en su lugar.

El avance: Usar energía (trabajo computacional) como un recurso relacionable y escaso entre extraños para prevenir ataques Sybil y alinear incentivos.

El compromiso: Bitcoin usa mucha electricidad. Esto es controvertido. Pero es el costo de asegurar un sistema monetario sin depender de gobiernos o bancos. Si ese compromiso vale la pena es para que la sociedad lo decida.

Por ahora, entiende el mecanismo: Proof-of-Work convierte la computación en autoridad. Autoridad temporal, rotativa, ganada.

Y tal vez puedas empezar a ver el patrón general: necesitas algún “recurso” para probar esfuerzo y ganar el respeto de otros, luego necesitas algunas reglas para que no puedas realmente hacer lo que quieras con el poder temporal ganado, y finalmente necesitas incentivos para que la gente quiera jugar según estas reglas.

¿No suena esto familiar? ¿No suena esto similar a una democracia? ¿O al menos a un sistema de votación? ¿Cómo puede esto ser democrático si no todos tienen acceso a grandes recursos computacionales o electricidad barata? Profundizaremos en las implicaciones de todo esto más adelante en el libro.

10.12 ¿Qué es una Blockchain, Realmente?

Ya puedes saber realmente qué son todos estos nombres elegantes: Bitcoin, Ethereum, Solana, Cardano, etc.

Por ahora quiero establecer que redes blockchain, criptomonedas, o cripto son todos nombres horribles. Ocultan la verdadera naturaleza de la tecnología.

Pero es el nombre que se ha quedado.

Estos son mejores nombres para hablar de esta tecnología: - **Redes de base de datos basadas en consenso.**

Cuando alguien dice “blockchain,” ahora puedes entender que están hablando de una “red de base de datos basada en consenso” donde los participantes acuerdan el estado de una base de datos compartida en una red.

En Bitcoin, el mecanismo de consenso es Proof-of-Work, pero en la práctica, puede ser CUALQUIER COSA— incluso un consenso centralizado. Un consenso autoritario, un consenso oligárquico basado en plutocracia donde solo los ricos pueden escribir los próximos bloques. Etc.

Los usuarios de Bitcoin valoran la capacidad de calcular cálculos hash realmente rápido como un recurso difícil de conseguir, lo que hace que sus reglas sean imposibles de romper.

Ethereum comenzó de esta manera también y luego cambió su consenso por razones que exploraremos después.

Finalmente, como humanos podemos simplemente acordar estados de base de datos distribuidos, y algunos de nosotros pensamos que es una buena forma de representar dinero.

¿Podemos incluso escribir las leyes en esta base de datos de tal manera que ningún gobierno corrupto pueda cambiarlas? ¿Podemos crear un mecanismo de votación perfecto que realmente represente a cada ciudadano sin elecciones manipuladas? ¿Cómo deberíamos definir el “esfuerzo” en el consenso para eso? ¿Deberíamos incluso dar tanto poder a la gente si esto es posible? ¿Estamos mejor en manos de tecnócratas?

Me gustaría que el lector se sintiera empoderado. Tener la capacidad de coordinarse con cualquiera sobre lo que constituye datos válidos, información válida, eventualmente si te gusta, verdad válida, es un progreso tecnológico increíblemente complejo y útil para que podamos coordinarnos mejor como especie.

Y como el tío Ben le dijo a Peter Parker: “Con un gran poder viene una gran responsabilidad.” Debemos estudiar, debemos entender, debemos tener cuidado. Construir lenta pero seguramente cosas que importan y son útiles para todos nosotros.

Por ejemplo, puedes entender un casino como una coordinación y como un consenso. La gente juega juegos consensuados en un casino y apuesta dinero. Puedes hacer que tu base de datos represente un casino, establecer algunas reglas, y permitir que cualquiera en el mundo “apueste justamente” a través del internet.

Estas aplicaciones tipo casino están siendo lo principal que la gente normal encuentra atractivo y repugnante sobre “cripto”.

Por favor, no te ciegues con los nombres elegantes, no te ciegues con la gente que usa estas hermosas estructuras de base de datos para estafar a otros o potenciar sus adicciones al juego.

Hay mucho potencial para el bien y el crecimiento de la especie aquí. Como el que Bitcoin tiene, que explicaré después.

Por ahora date cuenta de esto: las bases de datos guardan datos, tú interpretas los datos, tú y tus vecinos le dan significado. Adelante, coordínate con ellos, acuerden verdades sin dar a nadie acceso excesivo para definir las reglas.

Hablaremos más profundamente sobre las implicaciones de todo esto más adelante en el libro. Pero por ahora, es esencial entender realmente qué estás ganando y de qué trata realmente todo este progreso tecnológico.

10.13 Una Nota sobre el Pensamiento de Satoshi

No sé si Satoshi Nakamoto (quienquiera que sea él o ella o ellos) pensó en ello de esta manera mientras diseñaba Bitcoin. Tal vez lo hizo, tal vez lo abordó de manera diferente.

Pero así es como el autor de este libro hace sentido intuitivo de la progresión natural: Necesitamos coordinación, así que tomemos prestada la centralización temporalmente. Necesitamos justicia, así que rotemos el liderazgo. Necesitamos resistencia Sybil, así que requiramos prueba de esfuerzo. Necesitamos incentivos, así que recompensemos a los trabajadores.

Las piezas encajan elegantemente.

Idea Clave: Para resolver el problema de la base de datos distribuida, Bitcoin toma prestada la centralización temporalmente—dando a una persona a la vez el derecho de escribir y ordenar las próximas transacciones de todos. Pero en lugar de elegir líderes a través de votación (vulnerable a ataques Sybil) o selección aleatoria (también vulnerable), Bitcoin requiere prueba de esfuerzo computacional. Quien resuelva un rompecabezas difícil primero gana el derecho de escribir el próximo bloque y recibe una recompensa. Esto crea un liderazgo rotativo, temporal donde la autoridad se gana a través del trabajo, no se otorga por confianza. Proof-of-Work alinea incentivos: es más rentable jugar honestamente que hacer trampa.

A continuación, exploraremos cómo estos bloques se encadenan juntos a lo largo del tiempo, creando la estructura blockchain que hace que la historia sea evidente de manipular, y seguiremos profundizando en detalles importantes e implicaciones de este diseño.

11

Capítulo 11: La Blockchain - Encadenando la Historia

¿Qué pasa cuando dos mineros resuelven el rompecabezas al mismo tiempo?

Al final del Capítulo 10, dejamos una pregunta crítica sin responder.

Hemos establecido quién puede escribir en la base de datos (mineros que resuelven rompecabezas), cuándo escriben (cada 10 minutos), y por qué siguen las reglas (los incentivos se alinean). Pero no hemos abordado un caso límite fundamental: **¿qué pasa cuando dos mineros resuelven el rompecabezas exactamente al mismo tiempo?**

11.1 El Problema de la Solución Simultánea

Imagina que todos tenemos bases de datos sincronizadas mostrando los mismos saldos en el Bloque 100. (Recuerda, Bloque 100 simplemente significa que hemos acordado 100 veces el próximo estado de la base de datos—en el caso de Bitcoin, cada transición de estado involucra restar algunos saldos y añadir otros basados en transacciones.)

Estado actual (Bloque 100):

- Alice: 50 monedas
- Bob: 30 monedas
- Carol: 20 monedas

Ahora imagina dos mineros, Minero A y Minero B, ambos resolviendo el rompecabezas de Proof-of-Work casi al mismo tiempo. Ambos encontraron soluciones válidas, ambos hicieron el trabajo, y ambos merecen la recompensa.

Minero A comparte con la red Bloque 101a:

Bloque 101a:

- Transacción: Minero A recibe una recompensa según las reglas de incentivo, digamos 10 monedas
- Nuevos saldos: Alice: 50, Bob: 30, Carol: 20, Minero A: 10

Minero B comparte con la red Bloque 101b:

Bloque 101b:

- Transacción: Minero B recibe una recompensa según las reglas de incentivo, digamos 10 monedas
- Nuevos saldos: Alice: 50, Bob: 30, Carol: 20, Minero B: 10

Ambos bloques son válidos—ambos mineros siguieron las reglas, y ambos completaron el Proof-of-Work.

Ahora los nodos tienen que decidir: ¿qué bloque deberían aceptar?

Algunos nodos reciben el Bloque 101a primero y actualizan sus bases de datos en consecuencia, mientras que otros nodos reciben el Bloque 101b primero y actualizan a esa versión en su lugar. **Las bases de datos ahora están desincronizadas.** La mitad de la red piensa que un minero recibió la recompensa, y la otra mitad piensa que un minero diferente lo hizo.

11.2 La Elección Imposible

Aquí está el dilema que enfrentamos:

Si elegimos el Bloque 101a: - Rechazamos el trabajo del Minero B, aunque fue honesto e hizo el trabajo. - Rompemos la regla de consenso que dice “quien resuelve el rompecabezas puede escribir.” - ¿Por qué participaría alguien si su trabajo honesto puede ser rechazado arbitrariamente?

Si elegimos el Bloque 101b: - Rechazamos el trabajo del Minero A, aunque también fue honesto. - Mismo problema—estamos rompiendo las reglas.

Si aceptamos ambos: - El estado de la base de datos se vuelve inconsistente. - La mitad de la red tiene saldos diferentes que la otra mitad. - El punto entero del consenso—todos acordando el mismo estado—se pierde.

Si dejamos que la base de datos se desincronice: - El sistema se vuelve inútil porque diferentes personas ven diferentes verdades, diferentes datos. Nunca olvides: todo son solo datos, y lo que importa es cómo los interpretamos. - Si estamos interpretando esos datos como dinero, se vuelve sin valor si no podemos acordar quién tiene qué.

Si rompemos las reglas: - El consenso se rompe. - Las mismas reglas que hacen que la gente elija el sistema y lo encuentre valioso son violadas. - El sistema se vuelve sin valor.

Este es un problema real—no solo por el potencial de un actor malicioso para reescribir la historia (llegaremos a eso), sino también porque participantes honestos creando bloques válidos simultáneamente pueden causar que la red diverja y pierda sincronización.

11.3 La Solución Elegante: Déjalos Competir

Ambos mineros demostraron que hicieron el trabajo, y no podemos elegir uno sin ser injustos.

Así que aquí está la respuesta de Bitcoin: **No elijas todavía. Déjalos competir por una ronda más. Veamos quién puede sostener su esfuerzo.**

La regla: Cuando dos bloques válidos aparecen al mismo tiempo, no rechaces ninguno inmediatamente. En su lugar, ve cuál se extiende primero.

Cómo funciona:

1. Algunos nodos aceptan el Bloque 101a, otros aceptan el Bloque 101b.
2. Los mineros empiezan a trabajar en el próximo rompecabezas (Bloque 102).

3. Algunos mineros construyen encima del Bloque 101a, otros construyen encima del Bloque 101b. Recuerda que el rompecabezas depende de los datos del bloque anterior, así que deben elegir uno sobre el cual construir.
4. Cualquier bloque que se extienda primero (tenga otro bloque construido encima) gana.

Ejemplo:

```

+- Bloque 101a (Minero A) <- Minero C empieza a construir aquí
Bloque 100 ++
    +- Bloque 101b (Minero B) <- Minero D empieza a construir aquí

```

Unos minutos después, el Minero C resuelve el Bloque 102a (construyendo sobre el Bloque 101a):

```

    +- Bloque 101a -> Bloque 102a (esta "cadena" es ahora más larga)
Bloque 100 ++
        +- Bloque 101b (esta cadena es más corta)

```

Ahora la decisión es clara: la cadena con el Bloque 101a es más larga, lo cual simplemente significa que se ha puesto más esfuerzo en resolver rompecabezas para crearla. Siguiendo nuestro consenso de esfuerzo, deberíamos elegir esa.

Todos los nodos cambian a la cadena más larga, y el Bloque 101b es abandonado. El trabajo del Minero B es descartado, pero eso está bien—la decisión fue tomada jugando según el mismo consenso de esfuerzo. En el caso de Bitcoin, ese es el consenso de que cálculos hash rápidos representan una hazaña válida de esfuerzo.

Las transacciones del Bloque 101b que no fueron incluidas en el Bloque 101a vuelven al “mempool” (como una sala de espera para transacciones) y serán incluidas en bloques futuros si todavía son válidas.

Vale genial, ¡lo resolvimos! Ahora podemos seguir adelante y mantener sincronizada la base de datos incluso cuando hay empates.

¿Pero qué pasa si empatan de nuevo, y de nuevo, y de nuevo? En términos generales, ¿qué pasa si el consenso consistentemente nos da dos ganadores válidos dignos del derecho de escribir en la base de datos? ¿Qué pasa si alguien intenta engañarnos enviando bloques falsos en el futuro que parecen tener más trabajo en ellos?

Si queremos crear un sistema robusto, no podemos simplemente esperar lo mejor y asumir que esto nunca sucederá.

11.4 ¿Por Cuánto Tiempo Seguimos Haciendo Esto?

La respuesta no es general—es específica al mecanismo de consenso que uses en tu red de base de datos basada en consenso.

Recuerda, creo que “redes blockchain” es un nombre realmente malo, así que de ahora en adelante en este libro, las llamaré redes de base de datos basadas en consenso, o simplemente redes de base de datos de consenso, o incluso solo RBDC (Red de Base de Datos Consensuada) [en inglés CDN - Consensus Database Network].

La respuesta para el consenso de Bitcoin:

La cadena que acumula más Proof-of-Work gana. Como el rompecabezas que los mineros están resolviendo es verdaderamente aleatorio, eventualmente una cadena se adelantará—y eso solo puede ser causado por el hecho de que esos mineros han puesto mayores cantidades de esfuerzo en ella. La cadena más larga representa el mayor esfuerzo computacional, y esa es la que todos aceptan.

Esto se llama la “regla de la cadena más larga.”

La explicación de por qué una cadena eventualmente se adelanta ha sido simplificada aquí. Si tienes curiosidad, el whitepaper original de Satoshi Nakamoto tiene una explicación más formal de por qué esto funciona.

Cuando hay múltiples versiones competitoras de la base de datos, los nodos siguen esta regla simple: **Acepta la cadena válida más larga (iteración de consenso).**

Como ahora tenemos una nueva regla que tener en cuenta, tenemos que describirla con nuevos datos. Ahora necesitamos almacenar en nuestra base de datos no solo los saldos de las personas sino también cuántos pasos de esfuerzo se han puesto en la base de datos.

Un enfoque ingenuo sería simplemente almacenar un contador—un pequeño número de bits que cuenta cuántas iteraciones de esfuerzo (bloques) se han añadido. Pero esto es fácilmente hackeado.

¿Cómo rastreas iteraciones de una manera que nunca pueda ser falsificada? No puedes simplemente asumir que el próximo bloque válido será el Bloque 101, porque ¿qué pasa si alguien viene y dice, “Oye, encontré una base de datos válida donde nadie ha gastado doble, todas las transacciones son válidas, y dice Bloque 102”? Podría ser que alguien realmente puso más esfuerzo en la base de datos y encontró 2 bloques más mientras no estabas prestando atención.

Pero también podría ser que 2 mineros encontraron una solución válida al mismo tiempo, y uno fue honesto diciendo “el próximo bloque es 101,” mientras que el otro estaba haciendo trampa diciendo “el próximo bloque es 102.” Si la gente ve 102, pensarán que se puso más esfuerzo allí, y todos lo aceptarán.

El segundo minero acaba de burlar el sistema y engañó a todos haciéndoles pensar que se invirtió más esfuerzo cuando realmente no fue así.

¿Por qué pudo lograr esto? Porque no hay vínculo entre cada paso que tomamos en la base de datos—cualquiera puede simplemente afirmar lo que quiera sobre el próximo número.

Necesitamos una forma más inteligente de representar en qué iteración estamos, una forma que no pueda ser engañada. Una forma en la que nadie pueda reinventar ni el futuro ni los datos pasados de nuestra base de datos. Una forma en la que nadie pueda decir, “Estamos repentinamente en el futuro, chicos” (iteraciones más altas).

Y aquí es donde finalmente introducimos la estructura de datos blockchain—una estructura de datos que hace imposible falsificar el tiempo.

Pero antes de darte esta última pieza central del rompecabezas, déjame aclarar algo importante: Finalidad.

11.5 Entendiendo la Finalidad

Finalidad es un término técnico que usamos para definir cuánto tiempo tiene que pasar antes de que una iteración de nuestra base de datos se considere final—significando que nunca puede ser

cambiada de nuevo por nadie, nunca.

El tiempo hasta la finalidad depende del consenso que estés usando en tu red de base de datos basada en consenso. En Bitcoin, la finalidad es probabilística: cuantos más bloques se construyan encima de un bloque dado, más seguro se vuelve ese bloque.

Después de 6 bloques (aproximadamente 1 hora), la probabilidad de reescribir ese bloque se vuelve astronómicamente baja, así que para propósitos prácticos, lo consideramos final.

En otras RBDC, como Ethereum, la finalidad se logra a través de mecanismos diferentes. En el caso de Ethereum, toma aproximadamente 12-15 minutos considerar un bloque final.

Esto significa que en una RBDC, una vez que se alcanza la finalidad, nadie nunca podrá cambiar esos datos—incluso si tienen 51% del poder computacional en el caso de Bitcoin.

La propiedad de finalidad también es habilitada por la estructura de datos blockchain, y diferentes RBDC tienen diferentes tiempos de finalidad.

11.6 Ahora, ¿Qué es una Blockchain, Realmente?

Es simplemente una forma muy inteligente de almacenar en qué iteración de este cuento interminable de escribir en nuestra base de datos común estamos—de una manera que garantiza que nadie pueda hacer trampa al respecto.

Como puedes ver, la blockchain es en realidad solo otra pieza del sistema entero que estamos construyendo. Es una pieza de ingeniería que encaja en nuestra máquina.

Imagina llamar a los automóviles “máquinas de transporte de motor de combustión.” El motor es solo una pieza del sistema de transporte entero. Lo que importa es su uso, por lo que usamos la palabra “automóvil” (auto: por sí mismo, móvil: en movimiento) en lugar de “máquina de transporte de motor de combustión”—eso sería demasiado técnico y no muy útil.

El término “red blockchain” tiene un problema similar: es demasiado técnico y no muy descriptivo. La industria ha estado usando “red blockchain” durante demasiado tiempo. Como mínimo, prefiero decir: Una red blockchain es una red de base de datos basada en consenso que usa la estructura de datos blockchain para probar su historia.

Siendo honesto conmigo mismo, “red de base de datos basada en consenso” también es demasiado técnico para la mayoría de las personas. Pero al menos describe lo que estos sistemas realmente hacen, lo cual es más importante.

La gente normal entiende la palabra “datos” bastante bien—después de leer este libro, lo entiendes aún mejor. La gente también entiende “sincronización” y “descentralización”. Así que aquí propongo llamar a toda esta clase de tecnología **Tecnología Datasync Descentralizada** [Decentralized Datasync Technology] en lugar de Tecnología Blockchain.

Para ser honesto, no aprenderás ninguna funcionalidad nueva de aquí en adelante en este capítulo.

La estructura de datos blockchain es solo una forma segura de almacenar el contador—y también las transiciones de datos en la historia de nuestra base de datos—de una manera evidente de manipular. Esto asegura que todos sepan en qué iteración estamos, cuál viene después, cuál vino antes, y que nadie puede hacer trampa sobre nada de eso.

Durante el resto del capítulo explicaré intuitivamente cómo funciona esto, pero este es más un tema de ciencias de la computación que uno de “entender el uso de nuevas tecnologías”—que es en lo que este libro intenta enfocarse principalmente.

Estoy intentando explicar tan pocos detalles técnicos como sea posible mientras aún te doy una verdadera comprensión de cómo funcionan estos sistemas.

11.7 Opcional: Para los Curiosos (Puedes Saltarte Esto)

Si estás más interesado en las implicaciones societarias y prácticas de las redes de base de datos basadas en consenso que en las ciencias de la computación detrás de ellas, siéntete libre de saltarte el próximo capítulo y continuar en el capítulo 13.

El siguiente capítulo profundiza en los detalles técnicos de cómo funciona la estructura de datos blockchain—es fascinante, pero no esencial para entender el panorama general de lo que estos sistemas permiten.

Si te quedas, exploraremos estructuras de datos, memoria, punteros, listas enlazadas, y finalmente juntaremos todo para ver exactamente cómo una blockchain previene las trampas.

-> Continúa leyendo si tienes curiosidad sobre los detalles técnicos, o salta al Capítulo 13 para el próximo gran tema.

Idea Clave: Cuando dos mineros resuelven rompecabezas simultáneamente, no podemos elegir uno sin romper las reglas de consenso. La solución: deja que ambas cadenas compitan, y acepta la que se extienda primero (la regla de la cadena más larga). Para prevenir trampas sobre en qué iteración estamos, usamos la estructura de datos blockchain—una forma evidente de manipular de almacenar la historia. Esto permite un sistema donde el pasado no puede ser silenciosamente reescrito, y todos pueden verificar que están en la misma iteración de base de datos. Llamamos a toda esta clase de tecnología **Tecnología Datasync Descentralizada**—sistemas que sincronizan datos a través de miles de ordenadores sin control central.

A continuación, exploraremos qué permiten estos sistemas a nivel societario—cómo cambian las dinámicas de poder, permiten nuevas formas de coordinación, y por qué esto importa para el futuro. La base está completa. Ahora entendamos las implicaciones.

12

Capítulo 12: La Estructura de Datos Blockchain (Profundización Técnica)

Para los curiosos: ¿Cómo previene realmente la blockchain las trampas?

En el Capítulo 11, aprendimos que cuando dos mineros resuelven rompecabezas simultáneamente, los dejamos competir y aceptamos la cadena más larga. También aprendimos que necesitamos una forma inteligente de almacenar en qué iteración estamos—una forma que no pueda ser engañada.

Este capítulo es una profundización técnica en cómo funciona la estructura de datos blockchain. Si vienes del Capítulo 11, ¡bienvenido! Si vienes de otro capítulo—tal vez quieras leer todo hasta el Capítulo 11 primero para el contexto.

12.1 ¿Cómo Funciona Realmente la Estructura de Datos Blockchain?

Así que finalmente podemos explicar qué es una blockchain.

Pero primero, necesitamos entender algunos conceptos de ciencias de la computación. No te preocupes—lo mantendré simple y construiré desde lo que ya sabes.

12.1.1 ¿Qué es una Estructura de Datos? ¿Y Qué es la Memoria de un Ordenador?

¿Recuerdas del Capítulo 2 cuando hablamos sobre **estándares**? ASCII es un estándar para representar letras con bits, y JPEG es un estándar para representar imágenes. **Una estructura de datos es como un estándar, pero más complejo**—define cómo almacenamos grandes cantidades de datos que representan ideas complejas.

ASCII representa cosas simples como letras, pero ¿qué pasa si quieres representar algo más complejo, como un coche? Necesitarías información sobre: - Marca y modelo. - Año. - Color. - Precio. - Propietario.

Una **estructura de datos** define cómo organizar toda esta información en memoria para que un ordenador pueda almacenarla y recuperarla eficientemente.

¿Dónde se almacenan estos datos? En la memoria del ordenador—que, como sabemos del Capítulo 1, son solo muchos transistores representando bits (0s y 1s).

Puedes pensar en la memoria, físicamente, como un rectángulo hecho de metal con muchos cuadrados diminutos tallados en él. Si hacemos zoom conceptualmente, podemos imaginarlo como una cuadrícula:

Cuadrícula de Memoria (visualización simplificada):

0	1	1	0	0	1	0	1	← Fila 0 (dirección 0)
1	0	1	1	0	0	1	0	← Fila 1 (dirección 1)
0	0	1	1	1	0	0	0	← Fila 2 (dirección 2)
1	1	0	1	0	1	1	1	← Fila 3 (dirección 3)

Cada fila tiene una **dirección** (como una dirección de calle) para que el ordenador sepa dónde encontrar datos. Estas direcciones se construyen principalmente a nivel de hardware, no a nivel de software.

Cuando almacenas información, esencialmente estás haciendo que la electricidad pase a través de los cables conectados a esas filas, estableciendo estos bits en patrones específicos. Cuando recuperas información, buscas la dirección y lees el patrón de bits.

Una estructura de datos define cómo organizar estos bits en memoria para representar información compleja eficientemente.

Ahora, dentro de cada fila (dirección), podemos almacenar diferentes piezas de datos—y llamemos a cada fila una “variable” porque su contenido puede variar. Como somos humanos a quienes les gusta interpretar cosas, leamos cualquier dato que esté en la primera fila como una letra según ASCII.

Si dejamos que la electricidad pase selectivamente a través de la primera fila, estableciendo sus bits en valores específicos, y lo leemos como ASCII, podemos deletrear palabras. Aquí está la palabra “SAND” en memoria:

Cuadrícula de Memoria almacenando "SAND":

0	1	0	1	0	0	1	1	← Fila 0: 'S' (ASCII 83 = 01010011)
0	1	0	0	0	0	0	1	← Fila 1: 'A' (ASCII 65 = 01000001)
0	1	0	0	1	1	1	0	← Fila 2: 'N' (ASCII 78 = 01001110)
0	1	0	0	0	1	0	0	← Fila 3: 'D' (ASCII 68 = 01000100)

Nota que la imagen física que he descrito es solo una simplificación conceptual. En realidad, la memoria del ordenador es mucho más compleja, pero este modelo mental nos ayuda a entender cómo funcionan las estructuras de datos con precisión.

12.1.2 Punteros: Referencias a Otras Ubicaciones

Aquí hay una idea poderosa: **¿Qué pasa si una pieza de datos apunta a otra pieza de datos?**

Imagina que almacenas la información de un coche comenzando en la dirección 0. Podemos usar un sistema simple de punteros donde la primera pieza de información nos dice dónde comienzan los datos reales del coche.

Digamos que como humanos acordamos que:

- La fila 0 contiene un puntero (una dirección) a donde comienza la información del coche.
- Cuando seguimos ese puntero, la fila 1 contiene el tipo de coche.
- La fila 2 contiene el país donde se fabricó.

Cuadrícula de Memoria con punteros:

0	0	0	0	0	0	0	1	← Fila 0 (dirección 0): PUNTERO a dirección 1
0	1	0	1	0	1	0	0	← Fila 1 (dirección 1): Tipo de coche (Toyota)
0	1	0	1	0	0	1	1	← Fila 2 (dirección 2): País (EE.UU.)
0	0	0	0	0	0	0	0	← Fila 3 (dirección 3): Vacío

Un puntero es solo un número en memoria que te dice dónde encontrar otros datos en memoria.

¿Por qué es esto útil? Porque puedes crear relaciones entre piezas de datos—una pieza puede “referenciar” otra, estar vinculada con otra, encadenada con otra. ¿Ves a dónde va esto?

12.1.3 Listas Enlazadas: Encadenando Datos Juntos

Ahora combina estas ideas: **¿Qué pasa si cada pieza de datos contiene un puntero a la siguiente pieza de datos?**

En el ejemplo del coche, imagina que después de los datos del coche, tenemos otra fila de memoria con un puntero a los datos de otro coche que el propietario tiene. Esto crea una **lista enlazada**—una cadena de datos donde cada elemento apunta al siguiente.

Dirección 0	Dirección 3	Dirección 6
Datos Coche: A	Datos Coche: B	Datos Coche: C
Siguiente: (3)	→ Siguiente: (6)	→ Siguiente: NULL

En este caso, estamos representando en un ordenador qué coches posee alguien, encadenándolos juntos.

Para leer la lista: 1. Comienza en dirección 0, lee Datos Coche A. 2. Sigue el puntero a dirección 3, lee Datos Coche B. 3. Sigue el puntero a dirección 6, lee Datos Coche C. 4. NULL significa “fin de lista.”

Las listas enlazadas están hechas de punteros, y te permiten encadenar piezas de datos juntas en secuencia.

12.1.4 Funciones Hash (Repasso)

Ya hemos hablado sobre funciones hash antes (Capítulo 7 y Capítulo 10).

Una función hash es un algoritmo que toma cualquier entrada y produce una salida única de tamaño fijo:

```
hash("Hola") = d3a1f2
hash("Hola!") = 9f4e7b (completamente diferente)
```

Propiedades: - **Unidireccional:** No puede revertirse (no puedes obtener “Hola” desde d3a1f2). - **Determinista:** La misma entrada siempre da la misma salida. - **Única (con matices):** Diferentes entradas producen diferentes salidas. - **Sensible:** Cambia la entrada incluso ligeramente, la salida cambia completamente.

Las funciones hash crean **huellas únicas** para datos. Una función hash segura debe producir valores suficientemente grandes para evitar la posibilidad estadística de que dos entradas tengan la misma salida—pero no te preocupes por estos detalles. No los necesitas para entender cómo se construye una blockchain. Solo necesitas recordar las propiedades de las funciones hash.

12.2 Juntándolo Todo: La Estructura de Datos Blockchain

Una blockchain es un tipo de lista enlazada donde cada elemento es un **bloque**, y cada bloque contiene: 1. **Datos** (transacciones). 2. **Un puntero al bloque anterior** (pero no una dirección de memoria—en su lugar, el hash del bloque anterior).

Bloque 1	Bloque 2	Bloque 3
Transacciones	Transacciones	Transacciones
Hash de Bloque 0	Hash de Bloque 1	Hash de Bloque 2
Solución PoW	Solución PoW	Solución PoW

Cada bloque contiene el hash (huella única) del bloque anterior. Esto crea una cadena.

¿Por qué usar un hash en lugar de una dirección de memoria? Porque Bitcoin está distribuido—miles de ordenadores cada uno tiene su propia copia de la blockchain. Las direcciones de memoria son locales a un ordenador, pero un hash es el mismo en todas partes. Todos pueden calcular el mismo hash para su copia local del bloque anterior y verificar que la cadena es correcta.

12.3 Por Qué Esta Estructura Importa: Historia Evidente de Manipular

Aquí está la propiedad mágica de esta estructura: **Si cambias cualquier bloque, rompes toda la cadena.** No es un puntero, pero funciona similarmente. Añadí los conceptos de punteros y listas enlazadas porque ayudan con la comprensión visual.

¿Recuerdas las funciones hash? Cambia la entrada incluso ligeramente, y el hash cambia completamente.

Si alguien intenta cambiar el Bloque 100 (para robar monedas, reescribir la historia, etc.), su hash cambia:

Bloque 100 (original): hash = 0000abc123...

Bloque 100 (modificado): hash = 0000xyz789... (*¡completamente diferente!*)

Pero el Bloque 101 contiene el hash del Bloque 100. Si el hash del Bloque 100 cambia, el Bloque 101 ahora apunta a un hash inválido—el Bloque 101 se rompe.

Y como el Bloque 102 apunta al Bloque 101, también se rompe.

Y el Bloque 103... y 104... y cada bloque después de eso.

Cambiar un bloque rompe toda la cadena.

Visual:

Cadena original:

Bloque 99 → Bloque 100 (hash: abc123) → Bloque 101 (apunta a abc123) → Bloque 102...

Cadena modificada:

Bloque 99 → Bloque 100 (hash: xyz789) → Bloque 101 (*¿apunta a abc123???*) → SE ROMPE
↑

Bloque 101 espera hash abc123,
pero Bloque 100 ahora tiene hash xyz789.
La cadena es inválida.

Esta es la propiedad evidente de manipular. No puedes cambiar la historia silenciosamente. Cualquier modificación es inmediatamente visible porque la cadena se rompe.

12.4 ¿Pero No Puedes Simplemente Recalcular Todos los Hashes?

Pregunta inteligente. ¿Qué pasa si el atacante no solo cambia el Bloque 100, sino que también recalcula los hashes para todos los bloques subsiguientes?

Paso 1: Cambiar Bloque 100

Paso 2: Recacular hash del Bloque 100

Paso 3: Actualizar Bloque 101 para apuntar al nuevo hash

Paso 4: Recacular hash del Bloque 101

Paso 5: Actualizar Bloque 102... y así sucesivamente

Sí, podrías hacer esto. Pero aquí está el problema: **cada bloque requiere Proof-of-Work.**

Recuerda, para crear un bloque válido, debes resolver el rompecabezas computacional (encontrar un hash que comience con muchos ceros). A través de toda la red esto toma, en promedio, aproximadamente **10 minutos** de esfuerzo computacional masivo.

Así que para reescribir la historia: - Cambiar Bloque 100 → Debe rehacer Proof-of-Work (~10 minutos). - Arreglar Bloque 101 → Debe rehacer Proof-of-Work (~10 minutos). - Arreglar Bloque 102 → Debe rehacer Proof-of-Work (~10 minutos). - Arreglar Bloque 103... 104... 105...

Si quieres reescribir 6 bloques, necesitas ~1 hora de trabajo computacional.

Y mientras estás haciendo esto, la red honesta sigue avanzando, añadiendo nuevos bloques.

La carrera: - Tú: Intentando reescribir el pasado (comenzando desde el Bloque 100). - Red honesta: Construyendo el futuro (Bloque 106, 107, 108...).

Si la red honesta tiene más poder computacional que tú, siempre estarán adelante. Tu cadena bifurcada siempre será más corta.

Y recuerda la regla del Capítulo 11: la cadena más larga gana.

12.5 La Regla de la Cadena Más Larga (Repaso)

Introdujimos esto en el Capítulo 11, pero ahora entiendes por qué es tan poderosa.

Cuando hay múltiples versiones de la blockchain, los nodos siguen una regla simple:

Acepta la cadena válida más larga.

¿Por qué la más larga? Porque la cadena más larga representa el Proof-of-Work más acumulado—el mayor esfuerzo computacional invertido.

Ejemplo:

Cadena honesta: Bloque 1 → 2 → 3 → 4 → 5 → 6 → 7 (7 bloques, más PoW)

Cadena atacante: Bloque 1 → 2 → 3' → 4' → 5' → 6' (6 bloques, menos PoW)

Los nodos eligen: Cadena honesta (es más larga)

La cadena del atacante es rechazada—no porque sea “malvada,” sino porque tiene menos Proof-of-Work.

Esto significa: Para reescribir exitosamente la historia, necesitas: 1. Reescribir los bloques pasados. 2. Alcanzar la altura del bloque actual. 3. **Adelantarte a la cadena honesta** (para que la tuya se convierta en la más larga).

Si la red honesta controla 51% o más del poder computacional, el atacante nunca puede alcanzarla.

Cuanto más profundo está enterrado un bloque (más bloques construidos encima), más difícil es reescribirlo.

Por esto la gente espera “6 confirmaciones” antes de considerar una transacción de Bitcoin final. Después de 6 bloques (~1 hora), reescribir la historia se vuelve exponencialmente caro.

12.6 La Máquina Anti-Manipulación Psicológica

Por esto la blockchain a veces se llama una estructura “anti-manipulación psicológica” (en inglés “anti-gaslight”).

Gaslighting (manipulación psicológica) es cuando alguien te hace dudar de la realidad negando hechos repetidamente hasta que asumes que tú eres el loco y aceptas la mentira.

Sin blockchain:

Autoridad Corrupta: "Alice nunca envió a Bob 10 monedas."

Bob: "¡Sí lo hizo! ¡Tengo prueba!"

Autoridad Corrupta: [borra el registro] "Muéstrame la prueba."

Bob: "...No puedo. Tú controlas la base de datos."

Resultado: La manipulación tiene éxito.

Con blockchain:

Minero Corrupto: "Alice nunca envió a Bob 10 monedas." [intenta cambiar Bloque 100]
 Bob: "¡Sí lo hizo! Mira el Bloque 100 en mi copia de la blockchain."
 Minero Corrupto: [cambia Bloque 100] "Mi versión dice lo contrario."
 Bob: "Tu Bloque 101 apunta al hash equivocado. Tu cadena es inválida."
 Todos los demás: "Bob tiene razón. Rechazamos tu cadena modificada."
 Resultado: La manipulación falla. La historia se preserva.

Todos tienen una copia de la cadena. Si una persona intenta reescribir la historia, todos los demás lo notan porque los hashes no coinciden.

Este es el poder de la historia distribuida y evidente de manipular.

12.7 El Ataque del 51%

Mencionamos en el Capítulo 10 que si alguien controla el 51% del poder computacional, tienen más oportunidades de escribir bloques.

Ahora entendemos la implicación completa: **Con el 51% del poder de hash, puedes reescribir la historia reciente.**

Así es como:

1. Haces una transacción (Alice envía 10 BTC a Bob por un coche).
2. Bob te da el coche después de 1 confirmación.
3. Secretamente, empiezas a minar una cadena paralela desde antes de tu transacción, donde te envías esos 10 BTC a ti mismo en su lugar.
4. Porque tienes el 51% del poder, tu cadena eventualmente se vuelve más larga.
5. Transmitemos tu cadena más larga. Los nodos la aceptan (regla de cadena más larga).
6. La transacción de Bob es borrada. Tienes el coche Y tus BTC de vuelta.

Esto se llama un ataque de doble gasto.

Pero nota: - Solo puedes reescribir la historia reciente (últimos pocos bloques). Reescribir historia profunda (100+ bloques) es exponencialmente caro incluso con 51%. - Solo puedes gastar doble tus propias transacciones. No puedes robar las monedas de otros (no tienes sus claves privadas). - El ataque cuesta enormes cantidades de electricidad y hardware. - Si se detecta, el valor de la red se desploma, y tu hardware se vuelve sin valor. - El ataque es visible—todos ven dos cadenas compitiendo.

Por esto Bitcoin requiere un ataque del 51% para ser roto. Pero es un ataque irracional. El costo supera el beneficio para actores internos. Solo actores externos lo harían. Para RBDC más pequeñas, estados-nación podrían permitírselo, pero para Bitcoin, es prohibitivamente caro.

12.8 La Estructura Blockchain Resumida

Juntemos todo:

1. Una blockchain es una lista enlazada donde cada bloque contiene: - Datos (transacciones). - Hash del bloque anterior (el “puntero”). - Solución de Proof-of-Work.

2. Los bloques son evidentes de manipular gracias a los hashes: - Cada bloque contiene el hash del bloque anterior. - Cambiar un bloque → rompe todos los bloques subsiguientes.

3. Cada bloque requiere Proof-of-Work: - Reescribir la historia requiere rehacer todo el trabajo computacional. - Esto hace que la manipulación histórica sea exponencialmente cara.

4. La cadena más larga gana: - Los nodos aceptan la cadena con el Proof-of-Work más acumulado. - Los atacantes deben superar a la red honesta para tener éxito.

5. La historia profunda se vuelve inmutable: - Cuantos más bloques se construyan encima, más segura la historia. - 6+ bloques = efectivamente permanente (para la mayoría de propósitos prácticos).

Esto es la blockchain. No solo una “cadena de bloques,” sino una historia distribuida, evidente de manipular, de solo añadir, que hace que reescribir el pasado sea computacionalmente inviable en redes de base de datos descentralizadas.

12.9 Por Qué Esto Importa

A lo largo de la historia, aquellos que controlaban los registros controlaban la verdad:

- Los gobiernos reescribieron libros de historia para borrar hechos inconvenientes.
- Los bancos alteraron libros de contabilidad para robar o falsificar fondos.
- Los dictadores destruyeron archivos para ocultar sus crímenes.

Blockchain invierte esta dinámica de poder.

Ninguna entidad individual controla la historia. Todos tienen una copia. La manipulación es visible. Reescribir requiere superar a la mayoría.

Esta es la máquina anti-manipulación psicológica. Una historia compartida, verificable, evidente de manipular que ninguna persona controla.

Ya sea dinero (Bitcoin), contratos programables (Ethereum), o cualquier dato que nos importe—blockchain proporciona una forma de coordinar sobre la verdad sin confiar en ninguna autoridad única.

Idea Clave: Una blockchain es una estructura de datos similar a una lista enlazada donde cada bloque contiene el hash del bloque anterior, creando una cadena evidente de manipular. Cambiar cualquier bloque rompe todos los bloques subsiguientes. Como cada bloque requiere Proof-of-Work, reescribir la historia significa rehacer todo ese esfuerzo computacional. La cadena más larga (trabajo más acumulado) gana, haciendo que la historia profunda sea efectivamente inmutable. Esto crea un libro de contabilidad “anti-manipulación psicológica” donde el pasado no puede ser silenciosamente reescrito. Combinado con distribución (todos tienen una copia), blockchain proporciona historia verificable sin control central.

Ahora que entiendes los detalles técnicos de cómo funciona la estructura de datos blockchain, estás aún más preparado para explorar el panorama general—qué permiten estos sistemas a nivel societario, cómo cambian las dinámicas de poder, y por qué esto importa para el futuro.

13

Capítulo 13: Ethereum - La Máquina de Computación Consensuada

¿Y si la base de datos pudiera ejecutar programas?

Hemos pasado los últimos capítulos entendiendo Bitcoin: una base de datos distribuida que permite a extraños acordar quién posee cuánto sin confiar en ninguna autoridad central.

Pero las transacciones de Bitcoin son simples. Alice envía 10 BTC a Bob, Bob envía 5 BTC a Carol, y eso es todo—solo transferencia de valor. Restar un número aquí, sumarlo allá. Hay un poco más de complejidad bajo el capó, pero en su núcleo, Bitcoin solo está ejecutando sumas y restas.

Eso en sí es un programa, uno muy simple que suma y resta saldos. Simple y... aburrido.

¿Qué pasa si la base de datos pudiera hacer más que solo rastrear saldos? ¿Qué pasa si pudiera **ejecutar programas complejos?**

Esa es la idea que dio origen a **Ethereum**—una base de datos distribuida que no solo rastrea saldos sino que también almacena programas. Estos programas se llaman **smart contracts** (contratos inteligentes).

13.1 Recapitulación de Bitcoin: Transacciones Simples

Las transacciones de Bitcoin son instrucciones sencillas:

Transacción:

- De: Dirección de Alice
- A: Dirección de Bob
- Cantidad: 10 BTC
- Firma: [Firma de Alice probando que autorizó esto]

La red verifica: ¿Tiene Alice 10 BTC? Sí. ¿Es válida la firma? Sí. Entonces actualizar la base de datos—restar 10 de Alice, sumar 10 a Bob.

Simple. Limpio. Funciona perfectamente para dinero, o simplemente valor en el sentido del oro. Algunas personas argumentan con muy buenas razones que Bitcoin no es dinero en el sentido de efectivo sino valor en el sentido de oro, pero ese es otro debate.

Por ahora solo nota lo que el scripting simple de Bitcoin no puede hacer fácilmente. No puede decir fácilmente “enviar dinero a Bob, pero solo si sucede alguna condición compleja X.” No puede

retener dinero hasta cierta fecha y luego liberarlo a múltiples partes si se cumplen ciertas condiciones complejas. No puede programar lógica compleja como préstamos o sistemas de votación.

Bitcoin mueve valor de A a B. Para eso fue diseñado.

Ethereum preguntó: **¿Y si pudiéramos programar cualquier cosa?**

13.2 La Idea: Transacciones Programables

Vitalik Buterin (el creador de Ethereum—este tipo, a diferencia de Satoshi, es conocido y está vivo) se dio cuenta: **¿Y si pudiéramos programar lógica compleja en la base de datos?**

¿Y si pudiéramos programar un préstamo? ¿O un testamento? ¿O incluso un sistema de votación?

No solo “enviar X a Y,” sino: - **SI** se entregan los bienes **ENTONCES** pagar al vendedor **SI NO** reembolsar al comprador. - **SI** estoy inactivo durante 1 año **ENTONCES** enviar mis fondos a mis herederos. - **SI** es el día 1 del mes **ENTONCES** deducir la cuota de suscripción. - **SI** soy un votante válido **ENTONCES** permitirme almacenar mi voto para esta elección. - **SI** el ciudadano se porta mal, **ENTONCES** congelar sus fondos. - **SI** mi partido político está perdiendo, **ENTONCES** añadir votos falsos.

Esto es un smart contract. No es un contrato legal (no hay abogados involucrados), y tampoco tiene smartness (inteligencia), pero es un programa que se ejecuta en una RBDC y hace cumplir reglas automáticamente.

La parte smart depende de quién lo escriba. Y la palabra “contract” (contrato) podría venir del hecho de que necesitas firmas criptográficas para interactuar con él—en el “mundo real” firmas contratos y estos hacen cumplir cosas, y aquí firmas datos con criptografía lo cual permite que alguien ejecute código en una base de datos, así que es algo así como un contrato.

El nombre es un poco raro, pero la idea es poderosa. **¿Y si pudiéramos programar cualquier cosa en esta base de datos?**

Déjame mostrarte lo que esto significa con ejemplos reales.

13.3 Smart Contracts: Lógica Consensuada En Código

13.3.1 Ejemplo 1: Préstamos con Garantía

Quieres pedir un préstamo. Con un smart contract, puedes programarlo: en la fecha X, si no se devuelve el dinero, la garantía va al prestamista. Eso es todo.

Sin bancos, sin papeleo, sin verificaciones de crédito. Solo código haciendo cumplir el acuerdo.

Pero espera, **el programa solo puede leer datos de la base de datos**, por lo tanto, ¿cómo sabe el programa cuáles son los datos del mundo real, como el precio de tu garantía que podría ser, digamos, una acción de Netflix?

Gran pregunta. Los programas en la base de datos necesitan información del mundo exterior (como precios, tiempo, confirmaciones de entrega). ¿Quién le da estos datos a la base de datos para que el programa pueda leerlos? ¿No es eso una parte centralizada de confianza?

Bueno, podría serlo, pero ese es un tema para otro momento. Por ahora, imagina que es posible poner datos reales en el programa de manera verificable y minimizando la confianza con incentivos económicos y teoría de juegos como hemos estado haciendo.

Si tienes curiosidad, investiga más profundamente lo que llamamos en la industria **oracles**—el más famoso es Chainlink. La idea principal de estos protocolos oracle es que ponen datos en la RBDC desde el mundo real, como feeds de precios, datos del clima, resultados deportivos, etc., de manera verificable y minimizando la confianza.

13.3.2 Ejemplo 2: Testamento (Heredando Bits)

Quieres que tus activos de Ethereum vayan a tus hijos si algo te sucede.

Solución tradicional: Escribir un testamento legal, darle tus contraseñas a un abogado (arriesgado), esperar que todo salga bien.

Solución con smart contract:

Contrato: Testamento Digital

- SI no interactúo con este contrato durante 2 años
ENTONCES enviar 50% de mi ETH a la dirección del Hijo A
Y enviar 50% de mi ETH a la dirección del Hijo B.
- SI NO SI interactúo (pruebo que estoy vivo)
ENTONCES reiniciar el temporizador de 2 años.

Esto es básicamente imparable. Incluso si pierdes tus claves privadas, incluso si mueres, el contrato se ejecuta automáticamente después del período de tiempo. Sin abogados, sin tribunal, nadie puede bloquearlo.

13.3.3 Ejemplo 3: Suscripción (Pagos Recurrentes Automáticos)

Te suscribes a un servicio por 10\$/mes.

Solución tradicional: Darles tu tarjeta de crédito. Confiar en que no cobrarán de más. Esperar acordarte de cancelar.

Solución con smart contract:

Contrato: Servicio de Suscripción

- El usuario deposita \$120 (para 1 año).
- Cada 30 días, el contrato envía \$10 al proveedor del servicio.
- El usuario puede cancelar en cualquier momento, el contrato reembolsa el saldo restante.

Tú controlas cuándo cancelar. El proveedor del servicio no puede tomar más de lo acordado. Las reglas son transparentes y automáticas.

Y no hay comisiones para todos estos intermediarios que hacen que las tarjetas de crédito funcionen en internet—solo las tarifas de ejecutar el programa en la RBDC (Red de Base de Datos Consensuada como Ethereum), que pueden ser menores. Las comisiones tradicionales de procesamiento de tarjetas de crédito oscilan entre **2–3% por transacción** para los comerciantes, mientras que las tarifas de transacción en tecnologías datasync descentralizadas dependen de la congestión de la red—oscilando desde fracciones de centavo hasta algunos dólares, a menudo significativamente más baratas que las tarifas de tarjetas de crédito para muchos casos de uso.

13.4 La Máquina Virtual de Ethereum (EVM): Todos Ejecutan los Mismos Programas

Aquí está la magia: **Cada nodo en la red de Ethereum ejecuta estos smart contracts.** Mira, la frase anterior está llena de palabras raras que no habrías entendido antes de leer este libro. Espero que puedas ver el progreso que estás haciendo y la complejidad real en todas estas nuevas tecnologías datasync descentralizadas.

Recuerda de Bitcoin: Cada nodo tiene una copia de la base de datos, y cuando sucede una transacción, cada nodo la verifica y actualiza su copia.

Ethereum añade: Cada nodo también almacena programas en la base de datos. Cuando alguien interactúa con un programa, cada nodo **ejecuta el programa** y calcula el resultado.

Esto es la Ethereum Virtual Machine (EVM): Un ordenador virtual que existe a través de miles de ordenadores reales.

13.4.1 Direcciones: Todavía Basadas en Claves Asimétricas

Una nota rápida antes de continuar: Las direcciones de Ethereum funcionan de la misma manera que las direcciones de Bitcoin. Todavía se basan en criptografía asimétrica (claves públicas/privadas) con las mismas propiedades que aprendimos en el Capítulo 7.

Se ven un poco diferentes debido a diferentes esquemas de hash y codificación, pero el mecanismo es el mismo: - Tu clave privada → Tu clave pública → Tu dirección.

Ejemplos: - Dirección de Bitcoin: 1A1zP1eP5QGefi2DMPTfTL5SLmv7DivfNa - Dirección de Ethereum: 0x742d35Cc6634C0532925a3b844Bc9e7595f0bEb

Diferentes formatos, mismos principios criptográficos subyacentes.

Las direcciones mostradas fueron inventadas para ilustración.

13.4.2 Cómo Funciona:

1. Escribe un smart contract (en un lenguaje de programación como Solidity).
2. Lo despliegas en Ethereum (se almacena en la base de datos).
3. Cualquiera puede interactuar con él (enviarle transacciones).
4. Cada nodo ejecuta el código del contrato con tu transacción como entrada.
5. Cada nodo calcula el mismo resultado (determinista).
6. La base de datos se actualiza basándose en el resultado.

Propiedad clave: Computación determinista.

La misma entrada lleva a la misma salida. Siempre. En cada ordenador.

Si Alice envía al contrato de préstamo una transacción de “devolver préstamo”, cada nodo ejecuta el código del contrato, ve que se cumple la condición, y actualiza la base de datos para devolver su garantía.

Todos están de acuerdo en la computación, igual que están de acuerdo en los saldos.

No puedes simplemente no ejecutar un programa, porque está almacenado en la base de datos. Si intentas censurar un programa, o ejecutarlo de manera diferente, la próxima iteración de la base de

datos que produces (el próximo bloque) será rechazada por todos los demás porque tus resultados (los datos en la base de datos) no coincidirán con los tuyos.

Por lo tanto, alterar el código para censurar programas requeriría alterar el código que crea el consenso, así que si lo haces, básicamente estás creando una red completamente nueva—una donde estarás solo, e inútil.

13.5 Por Qué Esto Es Innovación Revolucionaria: Programas Imparables

Piensa en los programas normales. Facebook puede eliminar tu cuenta. PayPal puede congelar tus fondos. Amazon puede cambiar sus términos de servicio.

¿Por qué? Porque controlan los servidores que ejecutan el código. Pueden modificarlo, apagarlo, o bloquearte en cualquier momento.

Los smart contracts son diferentes, por defecto: - Ninguna empresa individual puede controlarlos. - Una vez desplegados, se ejecutan para siempre. - Nadie puede apagarlos, ni siquiera el creador. - Nadie puede cambiar las reglas sin que todos lo vean.

Esto es consenso en la computación, no solo en los saldos.

Nota mi redacción, por defecto. Los programas son flexibles, así que puedes programar un smart contract que sea controlado por alguna empresa, o que pueda ser cambiado por alguna autoridad. Pero esa es una elección que haces cuando escribes el código y cuando interactúas con el código.

De la misma manera que alguien puede escribir código que es controlado por una empresa, alguien puede simplemente escribir ese mismo código pero controlado por nadie. El poder de decidir qué programa usar está en tus manos.

Ambos programas estarán ejecutándose para siempre en la RBDC, pero tú eliges con cuál interactuar.

Así que aquí está la parte crucial: Cualquiera puede desplegar su propio programa, y tú eliges qué programa ejecutar. No estás obligado a usar el programa X si crees que está abusando de tus datos, cobrándote demasiado, etc.

¿Un banco está prestando dinero con demasiada tasa de interés? Cualquiera puede crear un nuevo programa de banco con una tasa de interés más baja para que los clientes puedan prestar en términos más favorables.

Esto permite competencia a nivel de código. Transparente y auditável.

Esto permite tantas posibilidades...

13.6 ¿Qué Podría Esto Permitir? Deja Que Tu Imaginación Vuelle

Un hombre una vez dijo, si puedes soñarlo, puedes programarlo.

Piensa en las posibilidades:

¿Y si tuvieras una red social descentralizada donde la gente pudiera pagarte directamente para acceder a tu contenido en lugar de dar control total a una empresa de redes sociales y sus

servidores y algoritmos opacos?

¿Y si cualquiera pudiera ver el algoritmo de esta red social? Sin manipulación oculta, sin sistemas de clasificación secretos. Código transparente que todos pueden verificar.

¿Y si pudieras crear una organización descentralizada donde cada decisión se vota por los miembros, y los votos automáticamente ejecutan cambios? Sin CEO que pueda anular la voluntad de la comunidad.

¿Y si los artistas pudieran programar regalías en su arte digital, para que cada vez que se revenda, automáticamente obtengan un porcentaje—para siempre?

Las posibilidades son vastas. Bancos, seguros, sistemas de votación, cadenas de suministro, gestión de identidad—cualquier cosa que involucre reglas, acuerdos y confianza puede potencialmente ser programada en una RBDC.

Ethereum es una máquina de computación consensuada. Miles de personas acordando qué programas ejecutar sobre su preciosa información, que puede representar y significar cualquier cosa, porque la interpretamos.

Pero hay inconvenientes.

13.7 Los Lados Negativos:

13.7.1 1. Tienes Que Pagar Por Ello

El costo de ejecución del programa, al final del día, es electricidad, recuerda. Cada nodo ejecuta el programa, consumiendo poder computacional.

Algunas empresas podrían permitirte usar sus ordenadores centralizados gratis porque venden tus datos e información “privada”—pero en un ordenador descentralizado, todos tienen que pagar su parte justa del costo de electricidad para que el modelo sea sostenible.

Además, cuando despliegas un programa, eso literalmente ocupa algunos transistores y también funcionan con electricidad en todo el mundo, así que eso también tiene un costo. Si quieres poner tu código de banco en la base de datos para que cualquiera pueda ejecutarlo, tendrás que pagar por el espacio que usa.

¿Es esto caro? ¿Puede alguien permitírselo? Responderemos estas preguntas más tarde.

Por ahora, solo entiende: **la computación en RBDC cuesta dinero.** Esto es por diseño. Estos costos usualmente se llaman costos de **gas**. ¿Por qué gas? No lo sé. ¿Tal vez porque la gasolina alimenta coches y la computación alimenta ordenadores? No me importa, esta industria ya tiene tantos nombres raros, perdóname si no cuestiono este.

Por cierto, hay una razón extra para pagar cuanto más consumes. Imagina que le dices a la red que ejecute un programa que nunca termina—voilà, hackeaste la red, ahora nada más puede ejecutarse porque todos están ocupados ejecutando tu bucle infinito. Bueno, eso ahora se vuelve económicamente imposible. Cuanta más computación uses más pagas, así que para computar para siempre necesitarías pagar para siempre... ¿dinero infinito? Imposible.

13.7.2 2. ¡Puedes construir cualquier cosa! Espera... ¿cualquier cosa?

Con todas estas tecnologías datasync descentralizadas, puedes construir:

- **Intercambios descentralizados (DEXs):** Intercambia monedas con personas de todo el mundo sin un banco en el medio. Es como una bolsa de valores, pero dirigida por código en lugar de una corporación.
- **Banca descentralizada:** Pide y presta dinero sin un banco tradicional. Solo smart contracts haciendo cumplir los términos para todos, en todas partes.
- **DAOs (Organizaciones Autónomas Descentralizadas):** Organizaciones dirigidas por código y votos y que no siempre necesitan aprobación de una junta directiva. Piensa en ello como una empresa donde los accionistas automáticamente controlan cosas como el presupuesto a través de votación transparente.
- **Sistemas de identidad:** Posee tu identidad en línea a través de plataformas, no controlada por Google o Facebook.
- **Juego global:** Programa tu casino global imparable para promover el juego en todo el mundo sin que ningún gobierno pueda cerrarlo.

Y muchas más nuevas posibilidades, estos son solo algunos.

Algunos de estos son útiles. Algunos son hype. Algunos son experimentos. Algunos son de ética dudosa. Pero la **capacidad** ahora es real.

13.7.3 3. El Código Puede Tener Bugs (bugs significa errores en el código)

El código puede ser hackeado si no lo programas adecuadamente. ¿Dejas tu préstamo con un bug que puede multiplicar accidentalmente por 10 lo que debes? Estás en problemas.

Incluso si la RBDC funciona perfectamente, el código que escribes encima puede tener bugs. Así que ten cuidado con qué programas ejecutas.

Algunos errores en el código (bugs) podrían permanecer inactivos durante mucho tiempo, hasta que alguien los descubra y los explote. Esto ya ha sucedido múltiples veces, con millones de dólares perdidos—notablemente, el hack de The DAO en junio de 2016 resultó en la pérdida de aproximadamente 3.6 millones de ETH (aproximadamente \$60M a precios de 2016). Los hacks cripto totales en años recientes han alcanzado miles de millones por año.

Afortunadamente, la seguridad de los códigos que se están usando está mejorando año tras año, pero este riesgo todavía está presente. Las probabilidades de que te afecte se reducen, pero nunca serán cero.

Personalmente, el autor trabaja precisamente en esta parte de la industria: ciberseguridad de RBDC y smart contracts. Así que puedo decirte: este es un riesgo real, pero está siendo tomado en serio por muchos profesionales.

Lenta pero seguramente, las probabilidades de que pierdas tu dinero por estas razones irán a 0 en la práctica, y si sucede, protocolos de seguro cubrirán tus pérdidas, que realmente no son tu culpa.

Se sentirá como contratar un seguro de vida siendo de 25 años solo por si acaso te cae un rayo. Es muy poco probable, pero si sucede quieres estar cubierto.

A día de hoy cuando estoy escribiendo esto, 16 de diciembre de 2024, es muy arriesgado, el autor no pondría todos sus ahorros/inversiones en ninguna criptomonedas individual debido a posibles problemas de seguridad.

13.7.4 4. El Diablo Está En Los Detalles Que Estoy Ocultando Por Simplicidad.

Algunos de los usos con los que he provocado tu mente son muy complicados de integrar técnicamente. No imposibles, pero de hecho mucho más complicados.

Al menos, con el conocimiento que obtendrás de este libro, estarás mejor equipado para seguir entendiéndolos por tu cuenta, haciendo tus propias preguntas críticas, lógicas e informadas sobre ellos mientras aprendes cómo funcionan.

13.8 Bienvenido al Lado Oscuro

Como habrás notado, he descrito algunos casos de uso que son un poco... moralmente grises. Juego, préstamos no regulados, elecciones corruptas, etc.

Como con cualquier nueva tecnología, hay usos buenos y malos. Algunas personas usarán las RBDC para innovación y libertad. Otros podrían usarlas para estafas, fraude, evadir regulaciones legítimas o crear ilegítimas.

Por esto debes entender la tecnología. Para asegurarte de que la usas para “la mejora del mundo” y no dejar que alguien más la use para abusar de ti.

Esto es como las armas, los coches, o el internet mismo. Herramientas poderosas que pueden usarse para el bien o el mal. Tenemos el deber de preocuparnos, de entenderlas al menos a cierto nivel de detalle, para que podamos tomar decisiones informadas. Este libro es un esfuerzo claro en esa dirección.

13.9 Resumen: Bitcoin Coordina Valor, Ethereum Coordina Valor Y Computación

Alejémonos y veamos lo que hemos desbloqueado:

Bitcoin: - Base de datos distribuida. - Rastrea saldos (quién tiene qué dinero). - Consenso sobre reglas simples (todos están de acuerdo en la propiedad y transferencias simples). - Transacciones simples: “Enviar X a Y.”

Ethereum: - Base de datos distribuida + ordenador distribuido. - Rastrea saldos Y ejecuta programas. - Consenso sobre computación compleja (todos están de acuerdo en las salidas de programas). - Lógica compleja: “SI condición ENTONCES acción.”

Ambos son tecnologías de coordinación. Permiten a extraños acordar algo sin confiar en una autoridad central.

Bitcoin: “Todos estamos de acuerdo en que Alice tiene 10 BTC.”

Ethereum: “Todos estamos de acuerdo en que este programa debería enviar fondos a Bob porque se cumplió la condición.”

13.10 Hora de Contar Historias...

13.11 Pero Ethereum Comenzó con Proof-of-Work. Luego Algo Cambió...

Cuando Ethereum se lanzó el **30 de julio de 2015**, usaba el mismo mecanismo de consenso que Bitcoin: **Proof-of-Work**. Los mineros resolvían rompecabezas computacionales, quemaban electricidad, ganaban recompensas.

Pero el **15 de septiembre de 2022**, Ethereum hizo algo sin precedentes: **Cambió de mecanismo de consenso**. Ese evento se llamó:

The Merge (La Fusión): Ethereum hizo la transición de Proof-of-Work a **Proof-of-Stake**.

No más minería. No más quemar electricidad. Una forma completamente diferente de lograr consenso.

¿Por qué? Una razón principal: Proof-of-Stake consume mucha menos electricidad (aproximadamente 99.95% menos).

Aquí está la idea clave: Esto es consenso. Si la gente (nodos) quiere un algoritmo diferente, simplemente pueden acordar y cambiarlo.

La comunidad acordó que este cambio valía la pena. Votaron con su participación. El cambio sucedió. La red siguió funcionando.

Simplemente dijeron en un foro de internet algo como: En el bloque número X, ejecutaremos un código completamente diferente, ¿vale?

Y cuando llegó ese bloque, todos lo hicieron. Nota un detalle muy importante: ejecutaron código diferente, pero el estado de la base de datos permaneció igual. Todos los saldos, todos los programas, todo permaneció intacto. Solo cambió la forma en que se logró el consenso.

El consenso es social, ¿recuerdas? La tecnología lo permite, pero los humanos deciden las reglas.

Si tienes curiosidad sobre los detalles técnicos de cómo funciona Proof-of-Stake, puedes investigarlo más profundamente. Pero para nuestros propósitos, solo entiende: es otro mecanismo de consenso, otra forma de coordinar, con otro conjunto único de compromisos.

Por ejemplo, si quieres censurar Bitcoin tienes que controlar el 51% del poder computacional de la red. En Ethereum, el poder computacional ahora no importa nada. Lo que importa es... su misma moneda nativa, Ether.

¿La lección importante de todo esto? Las redes pueden evolucionar. El consenso puede cambiar. Mientras todos los participantes estén de acuerdo.

Pero, gracias a esa hermosa estructura de datos, la blockchain, toda la historia se preserva. El pasado es immutable, pero el futuro puede ser moldeado por consenso. Literalmente puedes ver todos los cambios que sucedieron con el tiempo, quién los propuso, cuándo, y cómo la comunidad acordó ejecutarlos. Nadie puede censurar cómo sucedió el pasado, nadie puede reescribir la historia, pero el futuro está abierto a la elección colectiva humana.

Ahora, ¿qué pasa si no todos están de acuerdo? ¿Qué pasa si la comunidad se divide? Esto ya ha sucedido múltiples veces en múltiples RBDC. Exploraremos eso en el próximo capítulo.

Idea Clave: Bitcoin coordina valor, Ethereum coordina valor y computación. Los smart contracts son programas que cualquiera puede desplegar (desplegar en = guardar en) en la base de datos, ejecutados por cada nodo con lógica determinista. Tú eliges con qué programas interactuar—creando competencia transparente a nivel de código. Esto permite préstamos, testamentos, suscripciones, votación, e innumerables aplicaciones sin intermediarios. Pero hay compromisos: la computación cuesta dinero (gas), el código puede tener bugs, y las herramientas poderosas pueden servir tanto a propósitos buenos como malvados. Los programas son imparables por defecto, pero los desarrolladores pueden elegir hacerlos controlables. Cuando la comunidad de Ethereum decidió cambiar de Proof-of-Work a Proof-of-Stake, lo hizo—porque el consenso es en última instancia social. Las redes pueden evolucionar cuando los participantes están de acuerdo. Pero ¿qué pasa cuando no están de acuerdo?

A continuación, exploraremos qué sucede cuando el consenso se divide—cuando la comunidad está en desacuerdo tan fundamentalmente que la red se divide en dos realidades separadas. Esto profundizará nuestra comprensión de lo que realmente significa el consenso y por qué es en última instancia una elección humana, no solo un mecanismo técnico.

14

Capítulo 14: Cuando el Consenso se Divide - La Naturaleza del Acuerdo

¿Qué pasa cuando no todos están de acuerdo?

Hemos aprendido que el consenso es social. Ethereum cambió de Proof-of-Work a Proof-of-Stake porque la comunidad estuvo de acuerdo, y la red evolucionó en consecuencia.

Pero aquí está la pregunta: **¿Qué pasa cuando la comunidad no está de acuerdo?**

¿Qué pasa si la mitad de los participantes quiere ir en una dirección, y la otra mitad quiere ir en otra?

Esto ha sucedido. Múltiples veces.

14.1 La Pregunta Fundamental

Recuerda lo que aprendimos: Las RBDC funcionan porque todos ejecutan el mismo código y están de acuerdo en las mismas reglas.

Pero el código es solo software. Cualquiera puede copiarlo. Cualquiera puede modificarlo. Cualquiera puede ejecutar su propia versión.

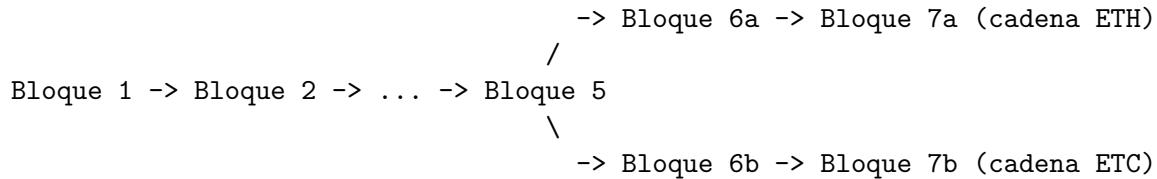
Cuando una porción significativa de una comunidad no está de acuerdo en qué código ejecutar, obtenemos lo que llamamos un **fork** (bifurcación). Piensa en ello como una bifurcación en el camino—dos coches conduciendo hacia esa división pueden elegir caminos diferentes, y desde ese punto en adelante, están viajando por carreteras diferentes, dirigiéndose en direcciones diferentes hacia mundos diferentes, poéticamente hablando.

A nivel técnico, los nodos simplemente empiezan a rechazar bloques que no siguen su versión de las reglas. Los mineros empiezan a minar en la versión que prefieren. Los usuarios empiezan a hacer transacciones en la versión en la que confían. ¿El resultado? Bases de datos diferentes con datos diferentes para cada grupo ejecutando software diferente.

Una blockchain, como sabes del capítulo 12, puede visualizarse como una cadena de bloques:

Bloque 1 → Bloque 2 → Bloque 3 → Bloque 4 → Bloque 5...

¿Qué pasa cuando el próximo estado de la base de datos difiere entre grupos? Bueno, nuestro lindo dibujo de la blockchain tiene que dibujarse con una división en dos—se bifurca, se divide:



Esto es lo que sucede ahora:

Entonces, ¿qué impide que la red se divida en múltiples versiones incompatibles?

La respuesta corta: Nada.

La respuesta larga: Déjame contarte dos historias.

14.2 Historia 1: El Fork de Ethereum - El Código Es Ley vs. Proteger El Ecosistema

En 2016, algo sin precedentes sucedió en Ethereum.

14.2.1 The DAO: Un Experimento de \$150 Millones

Alguien creó un smart contract (programa) llamado “The DAO” (Organización Autónoma Descentralizada). Era esencialmente un fondo de capital de riesgo diseñado para reunir dinero de inversores, pero dirigido completamente por código.

La gente enviaba dinero (ETH) a este contrato, y los poseedores de tokens votaban sobre qué proyectos financiar. Sin CEO, sin junta directiva—solo código haciendo cumplir las reglas.

Recaudó \$150 millones en ETH. En ese momento, eso era aproximadamente el 14% de todo el Ethereum en existencia.

Todos estaban emocionados. ¡Este era el futuro! ¡Organizaciones descentralizadas!

Entonces, alguien encontró un error en el código—un bug.

14.2.2 El Hackeo: \$50 Millones Robados

El 17 de junio de 2016, un atacante explotó una vulnerabilidad en el código de The DAO.

El bug les permitía retirar fondos repetidamente sin que el contrato actualizara adecuadamente el saldo—como un cajero automático que te da dinero pero olvida restarlo de tu cuenta.

Se drenaron \$50 millones en ETH.

El atacante no “hackeó” Ethereum. No cambiaron ninguna regla de consenso. Solo encontraron un fallo en el código del smart contract y lo explotaron—perfectamente legal según el código mismo.

Si escribes código que me envía dinero pero olvidas poner un límite en cuánto puede gastar, bueno, eso técnicamente no es mi culpa. El hackeo real era más complejo que esto, pero captas la idea de un error de código:

```

code SoyAliceYElNoEstaPagando() {
    enviarDineroDeVueltaAAlice(100$);
    dineroQueDebeAlice = -0$;
  
```

```

}
// Este pseudocódigo no resta nada de la deuda,
// permitiendo retiros infinitos a Alice. ¡Oh no!

```

En el hack de The DAO, el código hizo exactamente lo que estaba programado para hacer. El atacante simplemente entendió el código mejor que sus creadores y reconoció un error en su programación.

Recuerda del capítulo anterior: los programas en la cadena tienen sus propias reglas, y si quieras anularlas, no puedes hacerlo sin cambiar completamente el estado de la base de datos—lo que lleva a que las personas tengan estados diferentes y rechacen los bloques de los demás.

14.2.3 El Dilema: Dos Filosofías Incompatibles

La comunidad de Ethereum enfrentó una elección, y ambas opciones tenían argumentos fuertes e internamente consistentes:

Opción 1: El código es ley. Déjalo estar. - El atacante siguió las reglas del código (incluso si no fueron intencionadas). - Revertir esto establece un precedente peligroso: señala que romper el consenso base está bien si alguien se equivocó programando encima de él. - ¿Quién decide qué es una transacción “legítima” vs. un “hackeo”? En este caso claramente fue un hackeo, pero ¿qué pasa con casos futuros? Estas distinciones pueden volverse muy complicadas. - Si podemos revertir esto, podemos revertir cualquier cosa.

La inmutabilidad importa más que el dinero porque señala que somos honestos con nuestro consenso y por lo tanto confiables.

Imagina un país con un sistema legal muy inestable donde las leyes pueden cambiarse retroactivamente y rápidamente. Nadie querría hacer negocios allí. ¿Qué pasaría si una tarde, de la nada, tu cuenta bancaria se vacía debido a una nueva ley que dice “todo el dinero en bancos de empresas tipo X ahora pertenece al estado”?

Opción 2: Esto daña el ecosistema. Arréglalo. - \$50M representan el dinero de gente real. - El atacante explotó un error claro en el código, no comportamiento legítimo. - Si no arreglamos esto, la confianza en Ethereum colapsa. ¿Qué pensará la gente? ¿Entenderán los matices entre bugs de smart contracts y acuerdos a nivel de red? Ethereum tenía solo un año en ese momento, y la gente todavía estaba aprendiendo qué era. - Podemos hacer un fork—reescribir la historia para deshacer el hackeo. - **El pragmatismo importa más que la ideología.**

Ambos lados tenían puntos válidos. Ambos eran internamente consistentes. Ambos reflejaban valores reales.

Este no era un problema técnico. Era un desacuerdo filosófico sobre qué debería ser Ethereum.

¿Debería Ethereum ser una plataforma donde **el código es absolutamente ley**, incluso cuando eso lleva al robo? ¿O debería ser una plataforma que **protege a sus usuarios**, incluso si eso significa romper la inmutabilidad?

14.2.4 La Votación: 85% vs. 15%

La comunidad realizó una votación (a través de un mecanismo de señalización—no vinculante, pero indicativa).

85% votó por hacer fork — revertir el hackeo, devolver el dinero a la gente.

15% votó en contra — mantener la cadena como está, honrar “el código es ley.”

La tensión era palpable. Ambos lados sentían que estaban luchando por el alma de Ethereum. Ambos lados creían que tenían razón.

Así que Ethereum se bifurcó.

14.2.5 La División: Dos Ethereums

En el bloque 1,920,000, la red se dividió en dos cadenas separadas:

Ethereum (ETH): La cadena mayoritaria, la que todavía funciona como “la principal” hoy. Revirtió el hackeo y devolvió los fondos robados. La comunidad eligió el pragmatismo bajo la promesa de nunca hacerlo de nuevo—y hasta ahora han mantenido esa promesa, incluso con hackeos importantes de exchanges como el **hackeo de Bybit de febrero de 2025** donde aproximadamente 401,000 ETH (~\$1.5B) fueron robados del exchange centralizado (no todo estaba dentro de Ethereum pero captas la idea).

Ethereum Classic (ETC): La cadena minoritaria. Mantuvo la historia original, honró la inmutabilidad, y eligió la ideología.

Ambas son RBDC válidas. Ambas todavía están funcionando. Ambas tienen valor. Ambas tienen comunidades.

Valores de mercado (que fluctúan constantemente): - Al momento de escribir esto (principios de 2026), la capitalización de mercado de Ethereum está en los **cientos de miles de millones de dólares**, mientras que la de Ethereum Classic está en los **pocos miles de millones**.

El mercado votó con su dinero. La mayoría ganó económicaamente. Pero la minoría todavía existe.

Ahora que sabes un poco de los procesos técnicos detrás de los forks, déjame explicar qué pasó en la práctica:

En el bloque 1,920,000, algunos nodos eligieron seguir la cadena ETH, aceptando un nuevo bloque con una transacción especial devolviendo los fondos hackeados. Otros eligieron seguir la cadena ETC, rechazando ese bloque y continuando con las reglas originales.

Los mineros se dividieron. Los desarrolladores se dividieron. Los usuarios se dividieron. Los exchanges tuvieron que decidir qué cadena llamar “Ethereum” y cuál llamar “Ethereum Classic.”

La comunidad estaba dividida, pero ambas podían coexistir.

14.2.6 Ten Cuidado, El Pasado Puede Perseguirte

¿Recuerdas la estructura de datos blockchain? Su poder puede verse claramente aquí. Esto no es solo una historia—**es historia registrada**.

Ve a mirar el bloque 1,920,000 en ambas cadenas. Verás el fork. Puedes rastrear el ETH robado. Puedes ver a dónde fue en cada cadena.

En Ethereum (ETH): Los fondos fueron devueltos.

En Ethereum Classic (ETC): El atacante se los quedó.

Nadie puede ocultar esto. Nadie puede reescribirlo. La blockchain actual de ETH preserva el desacuerdo para siempre. Siempre mostrará cómo la comunidad rompió el consenso por una iteración, incluso si después retomaron usando las mismas reglas de consenso.

Como dirían las generaciones más nuevas: **Esta es la máquina anti-manipulación psicológica en acción.**

Para cualquiera que lea esto y no sepa qué es la manipulación psicológica (gaslighting): La manipulación psicológica es una forma de manipulación en la que una persona o grupo hace que alguien cuestione su propia memoria, percepción o cordura. Por ejemplo, en una relación tóxica, una pareja podría negar repetidamente eventos que la otra claramente recuerda, haciéndola dudar de su propio recuerdo y sentirse confundida o loca.

En este caso, la blockchain previene la manipulación psicológica al hacer pública y verificable la historia de desacuerdos. Nadie puede decir “eso nunca pasó.” Los datos están allí, para siempre—hasta que el último nodo se apague, de todos modos. Hay algunos matices en eso, pero captas la idea. Te animo de nuevo a investigar si tienes curiosidad.

14.3 Historia 2: Bitcoin vs. Bitcoin Cash - La Guerra del Tamaño de Bloque

En 2017, Bitcoin enfrentó su propia división.

14.3.1 El Problema: Bitcoin Es Lento

Bitcoin procesa aproximadamente 7 transacciones por segundo. Visa procesa alrededor de 24,000.

A medida que Bitcoin ganó popularidad, las tarifas de transacción se dispararon y los tiempos de espera aumentaron. Se estaba volviendo caro y lento.

¿Por qué tan lento? Porque cada nodo procesa cada transacción, y el protocolo de Bitcoin limita el tamaño de bloque a 1 MB (megabyte).

Un megabyte (MB) es 1 millón de bytes. Un byte es 8 bits. Así que 1 MB = 8 millones de bits.

El tamaño de bloque es como la cantidad máxima de datos que puedes cambiar en la base de datos en cada iteración. Cada cambio de saldos (una transacción) escribe y borra algunos datos, así que hay un límite en cuántas transacciones pueden procesarse a la vez.

Si solo puedes escribir 8 millones de bits por iteración, y cada iteración toma 10 minutos (el objetivo de Bitcoin), estás limitado en cuántas transacciones puedes incluir.

14.3.2 El Debate: Bloques Más Grandes vs. Mantenerlos Pequeños (el tamaño importa)

Una vez más, la comunidad se dividió en dos bandos con visiones válidas pero incompatibles:

Equipo Bloques Grandes: - Simplemente aumentar el tamaño de bloque. - Permitir bloques de 8 MB en lugar de 1 MB. - Más transacciones por bloque = más rápido, más barato. - Necesitamos escalar ahora para competir con Visa. - Bitcoin debería ser usable para pagos cotidianos.

Equipo Bloques Pequeños: - Bloques más grandes = menos personas pueden ejecutar nodos (porque necesitarían más almacenamiento y ancho de banda para manejar el aumento de datos). - Menos personas ejecutando nodos = más centralización. - La centralización derrota el propósito de Bitcoin. - Aumentar la capacidad de transacciones con soluciones de Capa 2 en su lugar (como Lightning Network). No te preocupes, tocaremos qué es una solución de Capa 2 más adelante en este libro. - Bitcoin debería ser una capa de liquidación, no una red de pagos.

Nuevamente, ambos lados tenían puntos válidos. Ambos reflejaban valores diferentes sobre qué debería ser Bitcoin.

Un lado priorizó **accesibilidad y tarifas bajas**. El otro priorizó **descentralización y seguridad**.

14.3.3 El Fork: Bitcoin vs. Bitcoin Cash

La comunidad no pudo ponerse de acuerdo.

Así que el 1 de agosto de 2017, Bitcoin se bifurcó:

Bitcoin (BTC): Mantuvo bloques de 1 MB. Se enfocó en la descentralización. Desarrolló soluciones de Capa 2.

Bitcoin Cash (BCH): Aumentó a bloques de 8 MB (más tarde 32 MB). Se enfocó en el rendimiento de transacciones.

Ambos todavía existen. Ambos tienen comunidades. Ambos tienen visiones diferentes de lo que significa “Bitcoin”.

Valores de mercado (que fluctúan constantemente): - Al momento de escribir esto (principios de 2026), la capitalización de mercado de Bitcoin está en los **cientos de miles de millones de dólares** (acercándose o superando un billón en ocasiones), mientras que la de Bitcoin Cash está en los **pocos miles de millones**.

Nuevamente, una mayoría claramente ganó económicaamente. Pero la cadena minoritaria sobrevive.

14.4 El Patrón: La Tecnología Permite, Los Humanos Deciden

Nota lo que está sucediendo en ambas historias:

14.4.1 1. La Tecnología Permite El Fork

El código es de código abierto. Cualquiera puede copiarlo. Cualquiera puede modificarlo. Cualquiera puede ejecutar su propia versión.

Hacer fork es trivialmente fácil desde un punto de vista técnico. Solo cambia unas pocas líneas de código, anúncialo, y ve quién te sigue.

La estructura de datos blockchain hace fácil probar dónde sucedió la división. La naturaleza de código abierto hace fácil copiar y modificar.

14.4.2 2. Los Humanos Deciden El Resultado

Pero, ¿qué fork tiene valor? Esa no es una pregunta técnica—es una pregunta social.

La gente vota con su participación en la red y con su dinero: - ¿A qué nodos apuntan su hardware los mineros? - ¿En qué cadena construyen los desarrolladores? - ¿Qué moneda listan los exchanges? - ¿Qué moneda compran y mantienen los usuarios? - ¿Qué fork valora la gente?

La respuesta determina qué fork “gana” económicaicamente (aunque ambos pueden sobrevivir).

En el caso tanto de Ethereum como de Bitcoin, la cadena mayoritaria ganó la batalla económica decisivamente. Pero las cadenas minoritarias no desaparecieron—encontraron sus propias comunidades que valoraron sus principios.

14.4.3 3. Blockchain Hace El Desacuerdo Auditável

En sistemas tradicionales, los desacuerdos pueden ocultarse, censurarse o reescribirse.

En RBDC, el desacuerdo es permanente.

Gracias a la propiedad de inmutabilidad de la estructura de datos blockchain, puedes ver exactamente cuándo sucedió el fork. Puedes rastrear qué usuarios fueron en qué dirección. Puedes verificar la historia tú mismo.

Nadie puede decir “eso nunca pasó.” Los datos son públicos.

Esto crea un registro permanente y auditável de cada desacuerdo importante en la historia de la red.

14.5 Los Forks Son Guerras Civiles, Más o Menos

Cuando las comunidades están en desacuerdo fundamentalmente, pueden dividirse.

Al principio, los forks suenan mal. “¡La red se dividió! ¡No es eso un fracaso?”

No realmente. **Los forks son prueba de que la coordinación es voluntaria.**

Podrías pensar: “¿No sería mejor si todos simplemente estuvieran de acuerdo? ¡No nos haría eso más fuertes! Somos más débiles divididos—eso parece malo.”

Ese es un punto de vista totalmente válido. Aquí es donde entra en juego la ética, y nos damos cuenta de que la ética no es objetiva.

¿Son los forks malos o buenos? Depende de tus valores y el contexto en que sucedieron.

Si valoras la unidad y la fuerza a través de la escala, los forks parecen fracasos. La comunidad es más débil dividida.

Si valoras la libertad y la capacidad de salir, los forks parecen éxitos. La minoría tiene poder.

Si valoras la inmutabilidad por encima de todo, el fork de ETH fue una traición. Si valoras el pragmatismo y proteger a los usuarios, fue una intervención necesaria.

Si valoras la accesibilidad, los bloques más grandes de Bitcoin Cash tienen sentido. Si valoras la descentralización, los bloques pequeños de Bitcoin tienen sentido.

No hay una respuesta objetivamente correcta. Esto es filosofía, incluso política si quieras—no matemáticas.

14.5.1 No Puedes Forzar Consenso Global

Nadie puede hacerte ejecutar código específico. Nadie puede forzarte a estar de acuerdo.

Bueno, si todos los nodos están en un país, el ejército de ese país podría ser capaz de... pero la mayoría de las RBDC son globales y distribuidas a través de países que no necesariamente se gustan entre sí.

Si el 85% quiere revertir un hackeo, el 15% puede decir “no” y mantener viva la cadena original.

Si no estás de acuerdo lo suficientemente fuerte, puedes hacer fork. Y si suficientes otros están de acuerdo contigo, tu fork sobrevive.

Esta es una tecnología de libertad colectiva. La libertad para que cualquier comunidad use la tecnología que mejor se adapte a sus intereses.

14.5.1.1 ¿Cuántas Personas Necesitas Para Hacer Fork?

Solo nosotros 2.

Mientras tengas un grupo—es decir, al menos 2 personas—ambos pueden ejecutar su propia versión del código en la base de datos y convencer a otros de usarla. Mientras tengas al menos 2 personas, tienes consenso entre ustedes y pueden ejecutar su propia versión del código.

¿Es eso económicamente sostenible? ¿Tienen ustedes dos suficiente dinero para ejecutar los ordenadores incluso si nadie está dispuesto a pagar para escribir datos en su base de datos? Si sí, pueden ejecutar su propia versión del código y crear su propia “sociedad digital,” su propia RBDC. Si no, bueno, será mejor que encuentren usuarios dispuestos a pagar por sus servicios.

O amenazarlos, quién sabe. Nunca olvides los lados oscuros de la naturaleza humana. Pero tampoco seas demasiado paranoico. En el equilibrio encontramos la virtud, diría Aristóteles.

Pero, ¿quién soy yo para decirte qué hacer? Nadie—de la misma manera que no soy nadie para decirte qué consenso ejecutar en tu ordenador.

14.5.2 La Historia Del Desacuerdo Se Preserva

Los gobiernos reescriben libros de texto de historia. Las corporaciones borran registros embarazosos. Las plataformas centralizadas banean voces disidentes.

Las RBDC, porque usan blockchains, no pueden hacer esto.

La división Ethereum/Ethereum Classic es visible para siempre. Cualquiera puede estudiarla. Cualquiera puede aprender de ella.

Las generaciones futuras verán: - ¿De qué trataba el desacuerdo? - ¿Quién votó de qué manera? - ¿Cuáles fueron los argumentos? - ¿Cómo reaccionó el mercado?

Nadie puede censurar esto. La blockchain hace el desacuerdo permanente y auditável.

14.6 La Realización Más Profunda: De Dos Sociedades A Muchas

Si el consenso puede dividirse en dos sociedades (ETH/ETC, BTC/BCH), ¿puede dividirse en muchas?

Sí.

Podrías tener: - Ethereum (pragmático, cadena mayoritaria). - Ethereum Classic (ideológico, enfocado en la inmutabilidad). - Ethereum [Nuevo Fork] (experimentando con características diferentes). - Y más...

Cada fork es una mini-sociedad con sus propias reglas, su propia comunidad, sus propios valores.

14.7 Suficiente División—¿Por Qué No Unirse?

Aquí hay un pensamiento provocador: ¿Y si en lugar de dividirnos completamente, creáramos mini-sociedades que se coordinen con una mega-sociedad más grande?

¿Y si tuviéramos: - Una cadena principal (lenta, segura, cara, consenso global). - Muchas cadenas laterales (rápidas, baratas, especializadas, consenso local). - Todas coordinándose juntas cuando sea necesario.

Como Estados Unidos: - Gobierno federal (lento, seguro, árbitro final). - 50 gobiernos estatales (rápidos, locales, especializados). - Ambos trabajando juntos.

O algo “ligeramente” diferente, como la Unión Europea: - UE (capa de coordinación, reglas compartidas). - 27 naciones (soberanas, independientes, especializadas). - Ambas respetándose mutuamente.

Esto es escalado de Capa 2. Y es el tema de nuestro próximo capítulo.

Muy resumido y dejando detalles atrás, por ahora:

Las Capa 2 son simplemente RBDC que se comunican con otra RBDC, compartiendo datos cuando es necesario pero teniendo sus propias reglas y código para ejecutar por su cuenta.

En lugar de separación completa (como los forks), las Capa 2 mantienen conexión mientras permiten independencia. Lo mejor de ambos mundos.

14.8 Lo Que Esto Significa Para Cualquiera

Los forks nos enseñan algo profundo:

El consenso de software no es algo que impongas. Es algo que eliges.

No puedes forzar a la gente a estar de acuerdo. Solo puedes: - Presentar tu caso. - Escribir tu código. - Ver quién te sigue.

La red es, en última instancia, la gente que la eligió. El código es solo una herramienta. La gente decide qué herramienta usar. Si suficientes personas no están de acuerdo, la red se divide. Y eso no es necesariamente el fin del mundo, ya que ambas versiones pueden coexistir.

Esto es lo que realmente significa la descentralización: Ninguna entidad individual decide por todos. Los grupos lo hacen, empezando por el más pequeño—2 personas.

Esto no es libertad individual, sino libertad colectiva. Una tecnología que permite a cualquier colectivo coordinar su información de la manera que les plazca.

14.9 La Máquina Anti-Manipulación Psicológica, Mejorada

Hemos estado llamando a la blockchain “la máquina anti-manipulación psicológica” porque previene reescribir la historia.

Pero ahora vemos algo más profundo:

No puedes prevenir que el consenso cambie. La gente siempre estará en desacuerdo. Los forks sucederán.

Pero puedes probar cuándo y cómo cambió.

Si una mayoría intenta reescribir la historia, la minoría puede hacer fork y preservar el original.

Si un gobierno intenta censurar una transacción, los usuarios pueden hacer fork y mantenerla visible.

Si los desarrolladores intentan imponer nuevas reglas, los usuarios pueden rechazarlas y quedarse con el código antiguo.

Nadie tiene poder absoluto. Todos tienen el poder de salir.

Esta es una estructura social fundamentalmente nueva que permite coordinación instantánea voluntaria de datos con registros permanentes de desacuerdo.

La blockchain no previene el conflicto—hace el conflicto visible, auditabile, y sobrevivable.

Idea Clave: Los forks suceden cuando las comunidades están en desacuerdo fundamentalmente sobre valores y dirección. ¿Son buenos o malos? Depende de tus valores y contexto. El fork de Ethereum (hackeo de The DAO) se dividió sobre pragmatismo vs. inmutabilidad. El fork de Bitcoin (tamaño de bloque) se dividió sobre accesibilidad vs. descentralización. Ambos lados tenían argumentos válidos—esto es filosofía, no matemáticas. La tecnología hace que hacer fork sea fácil (copiar código, ejecutar tu versión), pero los humanos deciden el resultado (qué cadena tiene valor). La blockchain hace el desacuerdo permanente y auditabile—puedes verificar exactamente cuándo/cómo sucedió. Los forks prueban que la coordinación es voluntaria: no puedes forzar el consenso, solo elegirlo. Con tan pocas como 2 personas, puedes hacer fork. La máquina anti-manipulación psicológica mejorada: no puedes prevenir el cambio, pero puedes probar cuándo cambió y preservar alternativas. Nadie tiene poder absoluto—todos tienen el poder de salir. Pero, ¿y si en lugar de separación completa, nos coordináramos en múltiples niveles?

A continuación, exploraremos soluciones de escalado de Capa 2—mini-sociedades que se coordinan con mega-sociedades más grandes. Como los estados de EE.UU. y el gobierno federal, o las naciones de la UE y la UE. Consenso anidado, sociedades dentro de sociedades, todas trabajando juntas. Así es como las RBDC escalan sin forzar a todos a dividirse completamente.

15

Capítulo 15: ¡Una Sociedad Para Ti! ¡Una Sociedad Para Mí! ¡Una Sociedad Para Todos!

¿Podemos tener lo mejor de todos los mundos?

En el Capítulo 14, aprendimos que cuando las comunidades no están de acuerdo, pueden hacer fork. Separación completa. Cadenas diferentes, comunidades diferentes, valores diferentes.

Pero, ¿y si no tuviéramos que elegir entre unidad e independencia?

¿Y si pudiéramos tener **muchas sociedades coordinándose** en lugar de una sociedad global o muchas completamente separadas?

Esta es la idea detrás de las Capa 2 (C2). Supongo que se llaman así porque usualmente se dibujan junto a la otra blockchain, pareciendo que están formando capas:

Cadena Capa 2: ->->-> [] ->->-> [] ->->-> [] ->->-> [] ->->-> [] ->->-> Bloque, Bloque

Cadena Capa 1: ->->-> [] ->->-> [] ->->-> [] ->->-> [] ->->-> [] ->->-> Bloque

Pero antes de entender mejor las Capa 2, ayudará entender un problema fundamental.

15.1 El Trilema de Blockchain (como un dilema pero con tres opciones)

Hay un problema famoso en el diseño de RBDC llamado **el trilema de blockchain**.

Establece que solo puedes optimizar **2 de 3** propiedades:

1. **Descentralización:** Cualquiera puede participar (ejecutar un nodo, verificar transacciones...).
2. **Seguridad:** La red no puede ser fácilmente atacada o comprometida.
3. **Escalabilidad:** La red puede manejar muchas transacciones rápida y económicaamente.

Puedes elegir 2 cualesquiera y ser muy bueno en ellas, pero no las 3.

Expliquemos intuitivamente por qué.

15.1.1 Descentralización + Seguridad = Lento (Bitcoin, Ethereum)

Esto es lo que eligieron Bitcoin y Ethereum.

Descentralización: - Cualquiera con un ordenador puede ejecutar un nodo. - Los bloques son suficientemente pequeños (1-2 MB) para que muchas personas puedan almacenarlos y verificarlos. - Miles de nodos independientes operan mundialmente.

Seguridad: - Para atacar la red, necesitas una gran cantidad de un recurso difícil de obtener. - Extremadamente caro y difícil.

Sin embargo, esto viene con un costo de velocidad: - **Cada nodo procesa cada transacción.** - Cada nodo almacena cada bloque. - Cada nodo debe alcanzar consenso con cada otro nodo.

Hay un límite físico a qué tan rápido puede suceder esto. Si los bloques vienen demasiado rápido, algunos nodos no podrán mantener el ritmo con sus máquinas más pequeñas. Si los bloques son demasiado grandes, solo personas con hardware caro pueden participar, lo que lleva a la centralización y pierde las garantías de seguridad que proporciona la descentralización.

El tiempo de bloque (con qué frecuencia se crean nuevos bloques) se elige en las reglas de consenso. Bitcoin eligió 10 minutos, Ethereum eligió ~12 segundos. Pero no puedes hacerlo demasiado rápido sin arriesgarte a que solo ordenadores poderosos puedan mantener el ritmo.

Resultado: **~7 transacciones/segundo (Bitcoin), ~15 transacciones/segundo (Ethereum).**

Compara con otros sistemas de pago centralizados como Visa: **~24,000 transacciones/segundo.**

15.1.2 Escalabilidad + Seguridad = Centralizado

¿Y si quisieramos manejar 24,000 transacciones por segundo?

Opción: Hacer bloques enormes (como 1 GB en lugar de 1 MB).

Resultado: - Pueden caber muchas más transacciones por bloque. - Mucho más rápido, mucho más barato.

Pero el problema: - Bloques de 1 GB cada 10 minutos = 144 GB por día = 52 TB por año. - La mayoría de las personas no pueden almacenar tantos datos. - La mayoría de las personas no pueden descargar bloques de 1 GB cada 10 minutos. - **Solo grandes corporaciones y centros de datos pueden ejecutar nodos.**

Resultado: **Centralización.** Casi has recreado una base de datos tradicional con pasos extra.

Enhorabuena, en lugar de una dictadura de datos tradicional ahora tienes una pequeña oligarquía. Quiero decir, técnicamente en realidad es más descentralizado.

15.1.3 Descentralización + Escalabilidad = Caos (Inseguro)

¿Y si quisieramos que mucha gente participara (descentralización) Y procesar toneladas de transacciones rápidamente (escalabilidad)?

Opción: Mantener la red abierta para todos, pero hacer bloques enormes y que vengan muy rápido.

Lo que sucede: - Muchas transacciones caben en cada bloque (check de escalabilidad). - Cualquiera puede unirse (check de descentralización). - Pero tienes que procesar cosas muy, muy rápidamente.

El problema:

No todos tienen el hardware para mantener el ritmo. Imagina que estás procesando el bloque 10 mientras alguien en Japón ya está en el bloque 12, otro nodo en Brasil todavía está en el bloque 11, y el tipo rico con el ordenador más rápido está en el bloque 20.

La base de datos se desordena completamente y se desincroniza.

Además, las distancias físicas importan. Incluso a la velocidad de la electricidad, si los bloques vienen demasiado rápido, un nodo en Islandia y un nodo en Senegal no pueden mantenerse coordinados. Para cuando el Bloque 100 llega a Senegal, Islandia ya está en el Bloque 105. La red se fragmenta geográficamente.

Es como la cocina de un restaurante con estrella Michelin donde todos literalmente corren intentando coordinarse a velocidades imposibles. Los chefs chocarían entre sí, empezarían a gritar, se confundirían. Caos.

Resultado: **Inseguridad.** La red no puede mantener el consenso. Los nodos no están de acuerdo sobre el estado actual. Todo el sistema se desmorona.

15.2 Por Qué Las RBDC Descentralizadas Son Inherentemente Lentas

Al menos con algoritmos modernos, criptografía, y disponibilidad de hardware: **El consenso global toma tiempo.**

Recuerda cómo funcionan las RBDC: 1. Las personas envían transacciones. 2. **Cada nodo** las recibe y verifica. 3. Otro nodo (minero/validador) propone un bloque con esa transacción. 4. **Cada nodo** verifica el bloque completo. 5. **Cada nodo** actualiza su copia local de la base de datos.

Esto es coordinación a escala global.

Cuantos más nodos tienes (y por lo tanto más descentralización), más trabajo necesita suceder. Cada nodo debe calcular varias cosas por cada transacción.

Si lo quieres seguro—es decir, funcionando como se espera—no puedes apresurarlo. Si lo quieres descentralizado, no puedes limitar quién participa. Pero si también lo quieres rápido... tienes que sacrificar una de esas.

15.3 ¿Pero Necesitamos Consenso Global Lento Para Todo?

¿Y si no necesitamos consenso global para cada transacción individual?

¿Y si podemos ejecutar RBDC que sacrificuen cierto grado de descentralización a cambio de velocidad, pero solo para ciertos casos de uso?

Piensa en la vida real:

- No necesitas que el gobierno federal apruebe cada compra que haces en una tienda local.
- No necesitas que la ONU valide cada contrato entre dos personas en la misma ciudad.
- No necesitas que cada humano en la Tierra esté de acuerdo en lo que tú y tu amigo hacen juntos.

La mayoría de la coordinación puede ser local. Solo alguna coordinación realmente necesita ser global.

Así que, ¿y si aplicamos esto a las RBDC?

15.4 Soluciones de Capa 2: Mini-Sociedades Dentro De Una Mega-Sociedad

La idea: Crear redes más pequeñas (Capa 2) que se sitúan “encima o junto a” una cadena principal (Capa 1).

Capa 1 (La Capa Base): - El “gobierno federal” o “nivel de la UE.” - Lento, caro, máximamente seguro, consenso global. - Árbitro final cuando surgen disputas. - Todos confían en él, pero no lo usas para cada pequeña cosa.

Capa 2 (Capas de Escalado): - Los “gobiernos estatales” o “naciones miembro.” - Rápido, barato, especializado, consenso local. - Maneja transacciones del día a día. - Periódicamente “liquida” con Capa 1 por seguridad.

Ambos trabajando juntos.

En lugar de procesar cada transacción en la cadena principal, procesamos la mayoría de las transacciones en Capa 2 y solo usamos Capa 1 cuando necesitamos crear un punto de control, liquidar, o resolver disputas.

Como llevar una cuenta en un bar. Normalmente no pagas después de cada bebida—liquidas al final de la noche.

15.4.1 Ejemplos de Capa 2

Bitcoin tiene Capa 2 como: - Lightning Network (para pagos rápidos y baratos)

Ethereum tiene Capa 2 como: - Arbitrum - Optimism - Polygon - Base - Y muchas más...

Cada una es esencialmente su propia RBDC que periódicamente se sincroniza con la cadena principal por seguridad.

Esto es exactamente cómo las sociedades humanas ya funcionan.

15.4.2 Estados Unidos

Gobierno Federal (Capa 1): - Toma grandes decisiones (enmiendas constitucionales, guerra, acuerdos comerciales). - Lento (el Congreso tarda una eternidad). - Caro (enorme burocracia). - Árbitro final (Tribunal Supremo).

Gobiernos Estatales (Capa 2): - Toman decisiones locales (leyes de tráfico, impuestos locales, educación). - Rápido (las legislaturas estatales se mueven más rápido). - Barato (burocracia más pequeña). - Se remiten a lo federal para disputas (la ley federal anula la ley estatal).

Ambos trabajan juntos. Los estados manejan la gobernanza del día a día. El gobierno federal maneja la coordinación importante.

No me repetiré con el ejemplo de la UE, es similar.

Pero las RBDC tienen nuevas características nunca vistas en la historia. Imagina que cada semana cada Estado actualiza sus datos a la RBDC federal, y así, como esa está descentralizada, permanece allí para siempre.

15.4.3 Un Ejemplo de Corrupción

Imagina dentro de un Estado, porque es una RBDC más centralizada y más fácil de corromper, hay un caso de corrupción donde el dinero es robado o mal usado. Entonces podrías usar los datos de la semana pasada como un punto de partida fresco, un momento consensual donde todos están de acuerdo sobre el estado de las cosas, y desde allí puedes empezar a investigar hacia atrás para ver qué tan profunda y durante cuánto tiempo sucedió realmente la corrupción.

En casos tradicionales normales, los archivos que prueban movimientos sospechosos durante semanas y semanas podrían ser “perdidos” o destruidos por aquellos en el poder que se corrompieron. Pero con puntos de control blockchain, solo tienes una semana para cubrir tus huellas. Tienes que apresurarte, y en un sistema legal serio, si intentas grandes crímenes rápidamente, las probabilidades son que te atrapen.

15.4.4 Otro Ejemplo

Le prestas dinero a Bob, pero Bob es amigo de Alice, una persona muy rica que también es política y está bien conectada con las personas que ejecutan esta RBDC rápida pero centralizada. Digamos que Bob no puede devolverte el dinero. Está en problemas, pero llama a Alice: “Oye, ¿podemos simplemente ‘perder’ los registros de este préstamo y pretender que nunca sucedió? ¿Podemos abusar del algoritmo de consenso para incluso reescribir un poco la historia?” Un poco, nota el juego de palabras.

Alice acepta y usa su influencia para conseguir que la RBDC centralizada “ pierda” los registros de este préstamo. Estás sin suerte.

Pero, en un sistema con actualizaciones periódicas a C1, podemos ver que Bob realmente tomó un préstamo de ti hace 2 semanas y ahora ha desaparecido. Hora de investigación para las autoridades federales.

La Capa 1 descentralizada actúa como un punto de control inmutable. Incluso si la Capa 2 está comprometida, la Capa 1 preserva la verdad.

15.4.5 Otro Otro Ejemplo

Imagina que pediste prestado a alguien, y ese prestamista sabía que no podrías devolver el préstamo pero aún así te dio el préstamo de todos modos—algo similar a la burbuja de 2008. Bueno, imagina que eso sucede a nivel estatal en EE.UU., y nuevamente, el sistema centralizado intenta excusarse con, “bueno... no lo sabíamos, ¿cómo lo habríamos sabido?” Bueno... los datos son públicos. Y mira la RBDC Federal: es claramente visible que estabas tomando decisiones muy arriesgadas, claramente irracionales que llevaron a muchos inversores y personas a la bancarrota.

Oh espera, todo es público... ¿Es eso bueno? Algunas personas se preocupan por la privacidad.

15.5 Las Noticias Temporalmente Malas: Compromisos

Las Capa 2 no son perfectas. Introducen complejidad.

15.5.1 La Complejidad Técnica Es Real

Técnicamente, las C2 pueden usar algoritmos muy diversos para ejecutarse, y todavía necesitamos encontrar una forma general de conectarlas fácilmente entre sí. La gente está trabajando en esto, pero por ahora, la complejidad técnica es algo real que hace desafiante ejecutar los escenarios descritos.

Estos sistemas existen, pero porque son complejos, es difícil para las personas normales entenderlos y por lo tanto usarlos para algo significativo.

Esto crea una barrera para la adopción. La tecnología funciona, pero la experiencia de usuario todavía se está descubriendo.

Imagina que para conectarte a internet tuvieras que hacerlo completamente diferente para cada marca de ordenador que uses. Eso ciertamente ralentizaría las cosas.

15.5.2 Carga Cognitiva: Demasiadas Opciones

Más allá de la complejidad técnica, también está la carga cognitiva que crea.

Los usuarios tienen que elegir qué Capa 2 usar. ¿Uso Arbitrum? ¿Optimism? ¿Polygon? ¿Base? Es como elegir en qué estado vivir—no todos entienden las diferencias exactas y toma tiempo hacerlo.

Cada Capa 2 tiene diferentes supuestos de confianza (algunas son más centralizadas que otras), diferentes velocidades y costos, diferentes modelos de seguridad, y diferentes aplicaciones construidas sobre ellas. **¿Cómo se supone que una persona normal elija sabiamente?**

Ahora mismo, la mayoría de las personas no lo hacen. Se quedan con Capa 1 o usan lo que su aplicación favorita recomienda. Esto derrota algo del propósito.

Si quieres mover independientemente tu dinero alrededor de todas estas RBDC, realmente tienes que saber lo que estás haciendo. ¿Cuál es el chainID? ¿Es confiable este proveedor de nodos? ¿Necesito crear una nueva billetera para esta cadena? ¿Está disponible el token o moneda que quiero usar en esta cadena? ¿Es el código en esta cadena tan seguro y revisado por expertos de seguridad como en esta otra cadena? ¿Dónde puedo encontrar una interfaz de usuario simple para mover fondos entre las cadenas con uno o dos clics? ¿Es seguro este sitio web que encontré? ¿Quién lo gestiona? Genial, encontré otra cadena—recojamos toda esta información de nuevo...

Es como mudarse de residencia—posible, pero no instantáneo o gratis.

Nadie tiene tiempo para esto, solo frikis como el autor. Como se dijo, avances técnicos como crear nuevas reglas de consenso van a simplificar el proceso en el futuro, y los usuarios no tendrán que saber tanto sobre todos los detalles en absoluto. Pero eso está simplemente en construcción.

15.5.3 Diferentes Supuestos de Confianza Y Accesibilidad de Software

Algunas Capa 2 son más centralizadas que Capa 1. Estás confiando en el operador de Capa 2 hasta cierto grado o completamente.

Es un compromiso: más velocidad y menor costo, pero un modelo de seguridad ligeramente o completamente diferente.

¿Qué pasa si el operador simplemente desaparece y no tuviste tiempo de mover tu dinero de vuelta a la RBDC segura? ¿Qué pasa si el desarrollador que hizo el sitio web desaparece y ahora, aunque

es técnicamente posible, no sabes cómo mover tus fondos de vuelta?

No todos en este mundo son ingenieros de software, y aún menos personas entienden y pueden interactuar rápidamente con RBDC. Y no es realista esperar que todos se conviertan en uno por el bien de un sistema financiero más efectivo.

¿Cuál es la solución entonces? Como se dijo, los técnicos están trabajando en ello—en sitios web que cualquiera puede ejecutar en sus máquinas incluso sin internet, en aplicaciones que son de código abierto con código que es público y todos pueden verificar, que pueden descargar y usar para siempre en su teléfono para mover fondos de forma segura incluso si su sitio web favorito está caído.

Incluso se han inventado mecanismos criptográficos donde, incluso si la C2 se rompe, las personas pueden simplemente mover sus fondos a la RBDC principal sin tener que interactuar con la rota.

No es realista esperar que todos se conviertan en ingenieros de software, pero no es poco realista crear software de código abierto usable y accesible por cualquiera a escalas globales. Linux, un sistema operativo gratuito que funciona perfectamente, es gratis y para todos, en cualquier lugar. Y así sucesivamente—hay mucho software capaz de esto.

Es cierto que estos softwares requieren un pequeño esfuerzo; no están simplemente disponibles en la Play Store donde los instalas con un clic. Pero esta cosa o dos extra que tienes que aprender es mínima y cualquiera puede hacerlo en una o dos tardes, especialmente las nuevas generaciones que nacieron alrededor de herramientas digitales. Y es probable que alguien cree una forma de incluso eliminar esta pequeña barrera de entrada en el futuro.

Por lo tanto, danos tiempo y te daremos libertad colectiva. O pon un pequeño esfuerzo en entender cómo descargar código de lugares como GitHub. Las IAs modernas pueden explicar cómo hacer esto bastante bien.

Si eres un ingeniero de software leyendo esto, te animo a usar tu conocimiento y construir para el nicho de RBDC. Muchas gracias, eres muy bienvenido, nunca tendremos suficientes personas trabajando—solo crea más descentralización y mejores sistemas. Y por favor, no olvides el propósito, no olvides que la descentralización es lo que mantiene la fiesta en marcha.

15.5.4 Contexto de Edad de la Industria

Esta industria tiene solo unos **16 años** a partir de **2025** si cuentas desde que se creó Bitcoin—muy joven—y todavía queda mucho código por escribir, y escribirlo de forma segura ralentiza el proceso. Arruina algo en esta industria y consigues algunos hackeos irrecuperables, como hemos explicado en capítulos anteriores.

15.5.5 Fragmentación

Si todos usan diferentes Capa 2, el efecto de red se divide.

Es como cómo la UE se beneficia de que todas las naciones se coordinen, pero cada nación hablando un idioma diferente crea fricción.

Si tu dinero está en Arbitrum y el dinero de tu amigo está en Optimism, hacer transacciones entre ustedes dos es más difícil que si ambos estuvieran en la misma red.

Es como enviar un paquete a otro Estado en lugar de enviarlo a alguien en la ciudad más cercana.

15.6 Resumiendo

Coordinación Local -> Regional -> Nacional -> Global.

- Te coordinas con tu familia (muy rápido, muy local).
- Tu ciudad se coordina internamente (rápido, local).
- Tu estado se coordina con otros estados (más lento, regional).
- Tu nación se coordina con otras naciones (lento, global).

Las Capa 2 son el mismo patrón aplicado a RBDC.

No estamos, a niveles de relaciones sociales, inventando algo nuevo—estamos reconociendo que la coordinación naturalmente sucede a múltiples escalas.

La mayor parte de tu vida diaria no necesita coordinación global. Solo algunas cosas lo hacen.

15.7 La Seguridad Se Comparte Globalmente

Imagina que en sistemas tradicionales, cada vez que tenías una disputa, el mejor juez, abogados y detectives del mundo te ayudaban. Eso es imposible.

De cierta manera, ya no. Como vimos, podemos actualizar periódicamente los datos al sistema seguro, creando historia inborrable, evidencia—legible y analizable a la velocidad de la electricidad, en cualquier lugar del mundo.

Sociedades dentro de sociedades, todas trabajando juntas y para cada una.

Ethereum no necesita procesar cada transacción. Solo necesita ser el árbitro final cuando surgen disputas o cuando se necesitan puntos de control.

Bitcoin no necesita registrar cada compra de café. Solo necesita liquidar saldos finales cuando los canales de Capa 2 se cierran.

Así es como escala la coordinación.

No a través de forzar a todos a estar de acuerdo en todo, sino a través de permitir que grupos locales se coordinen rápidamente mientras mantienen una capa global para la verdad y seguridad.

Esto crea la idea de Ethereum como la máquina de confianza global. ¿Por qué vives relajado incluso si literalmente cualquiera puede herirte cuando sales? Porque hemos diseñado nuestro sistema para tener mecanismos “automatizados”, como policía y jueces, para ayudarte cuando eso sucede. Estos sistemas tradicionales también son máquinas de confianza que construyen confianza. Y cuando confías puedes relajarte, y entonces mejor el sexo—lo mismo aplica en finanzas y coordinación de datos. Además, estos sistemas también pueden ser influenciados y mejorados por ti en caso de que se equivoquen, a través de votar por nuevas leyes, etc.

En una frase inversa, también generan la falta de necesidad de confiar, porque sabes que si alguien se porta mal, será castigado. Esto es lo que la industria llama “trustless” (sin necesidad de confianza).

Por cierto, el manifiesto trustless y la filosofía d/acc de Vitalik Buterin reflejan muy bien cómo las principales fuerzas que guían Ethereum tienen estas intenciones que he estado describiendo. Te animo a leerlos.

Como puedes ver, las RBDC son muy similares: acuerdos consensuales válidos sobre cuál es la información oficial, en la que todos pueden participar e incluso proponer cambios si las cosas parecen no funcionar. Y a diferencia de los sistemas tradicionales, estos sistemas pueden ser globales y realmente automatizados.

Este es el estado actual de pensamiento de la industria y un vistazo a las posibilidades que abre. Te animo a seguir aprendiendo sobre esto—como puedes ver, es avance tecnológico que nos otorga nuevas capacidades que simplemente no teníamos antes.

Pero ahora, espera, ¿te has dado cuenta? Todos estamos desnudos.

15.8 El Último Inconveniente, Privacidad

Houston, todavía hay un problema: **Todas estas transacciones y datos siguen siendo públicos.**

Todos pueden ver tu saldo. Todos pueden ver con quién haces transacciones. Todos pueden rastrear tu historial financiero.

Los criminales pueden ver cuánto dinero tienes y decidir si atacarte.

Los gobiernos corruptos también. Quiero decir, son un subconjunto del ejemplo de criminales arriba.

Incluso tus vecinos pueden ver tus hábitos de gasto, pueden envidiarte, y crear presión social.

Las compañías de seguros pueden ver tus datos y decidir cobrarte más o menos sin razón real relacionada con la póliza de seguro actual.

Los empleadores pueden ver en qué gastas antes de contratarte, discriminando basándose en tu vida personal.

Regímenes autoritarios pueden rastrear donaciones de disidentes a grupos de oposición.

Acosadores y ex parejas abusivas pueden monitorear tus movimientos financieros y patrones de ubicación.

Las empresas pueden rastrear tus compras y construir perfiles publicitarios invasivos sin tu consentimiento.

Bueno para auditabilidad y generación de responsabilidad. Malo para privacidad y protección.

¿Hay una forma de tener ambas, privacidad y verificación?

¿Puedes probar que tienes suficiente dinero para hacer una compra sin revelar cuánto tienes?

¿Puedes probar que tienes suficiente edad para entrar a un bar sin mostrar tu fecha de nacimiento exacta?

¿Puedes probar que votaste sin revelar por quién votaste?

Esto parecía matemáticamente imposible durante mucho tiempo. Si quieres probar que algo es correcto, ¿no tienes que mostrarlo?

Resulta que no. Y ese es el tema de nuestro próximo capítulo.

Idea Clave: El trilema de blockchain: solo puedes tener 2 de 3 (descentralización, seguridad, escalabilidad). Bitcoin y Ethereum eligieron descentralización + seguridad, sacrificando velocidad (~7-15 tx/seg vs ~24,000 de Visa). Pero no necesitamos consenso global para todo—la mayoría de la coordinación puede ser local. Las Capa 2 son RBDC más pequeñas encima de Capa 1: Capa 1 es lenta, segura, global (gobierno federal), mientras que las Capa 2 son rápidas, baratas, locales (gobiernos estatales). Ejemplos: Lightning Network (Bitcoin), Arbitrum/Optimism/Polygon/Base (Ethereum). Periódicamente crean puntos de control en Capa 1, creando un registro inmutable que previene que la corrupción se oculte. Compromisos: complejidad técnica (difícil conectar C2), carga cognitiva (demasiadas opciones), diferentes supuestos de confianza, sobrecarga de coordinación, fragmentación. Esto refleja la sociedad humana: local -> regional -> nacional -> global. El consenso puede ser anidado—sociedades dentro de sociedades. La mayor parte de la vida diaria no necesita consenso global, solo puntos de control para verdad y seguridad. Pero todas las transacciones siguen siendo públicas—¿podemos tener privacidad Y verificación?

A continuación, exploraremos las pruebas de Conocimiento Cero—probar que sabes algo sin revelar qué sabes. Esto parecía imposible hasta los años 1980, e impráctico hasta los años 2010, pero ahora es la pieza faltante para coordinación privada y verificable.

16

Capítulo 16: Pruebas de Conocimiento Cero - Probar Sin Revelar

¿Puedes probar que sabes algo sin mostrar nada sobre lo que sabes?

En el Capítulo 15, identificamos un problema crítico: **todas las transacciones en blockchains son públicas**. Todos pueden ver tu saldo, todos pueden rastrear tu historia, y criminales, gobiernos, vecinos y empleadores por igual pueden monitorear tu vida financiera.

Así que aquí está la pregunta: ¿Puedes probar que algo es verdad sin revelar qué es ese algo?

¿Puedes probar que tienes suficiente dinero para comprar una casa sin revelar tu riqueza total? ¿Puedes probar que eres mayor de 21 sin mostrar tu fecha de nacimiento exacta? ¿Puedes probar que conoces una contraseña sin escribirla?

Durante la mayor parte de la historia humana, la respuesta parecía obvia: **No. Si quieres probar algo, tienes que mostrar al menos algo de información sobre ello.**

Realmente parece imposible, se siente como magia, incluso ilógico. Pero es solo matemáticas. Piensa en ello por otro segundo: ¿cómo puedo mostrarte que soy mayor de 18 sin revelar nada más? ¿Cómo puedo decir “Oye, soy mayor de 18” y luego, sin que necesites creerme, simplemente SABES que lo soy? Es una locura, y por lo tanto, fascinante.

En los años 1980 (en **1985** específicamente), los matemáticos descubrieron algo impactante: **Puedes probar que sabes algo sin revelar lo que sabes.**

Esto se llama una **Prueba de Conocimiento Cero (ZKP por sus siglas en inglés)**. El nombre describe bastante exactamente lo que hace: probar cosas mientras revelas conocimiento cero sobre ello.

16.1 ¿Qué Es Una Prueba de Conocimiento Cero?

Una Prueba de Conocimiento Cero es una forma para que una persona (el probador) convenza a otra persona (el verificador) de que una afirmación es verdadera, sin revelar ninguna información más allá de la verdad de esa afirmación.

Ejemplo:

- **Afirmación:** “Conozco la contraseña de esta cuenta.”
- **Prueba tradicional:** Escribir la contraseña -> El verificador ve la contraseña.

- **Prueba de Conocimiento Cero:** Ejecutar un protocolo matemático -> El verificador está convencido de que conoces la contraseña, pero no aprende nada sobre cuál es la contraseña.

El verificador aprende SOLO que la afirmación “Conozco la contraseña” es verdadera. Nada más.

16.2 Una Analogía Simple Y Clásica: El Amigo Daltónico

Imagina que tienes un amigo que es daltónico. Tienes dos pelotas: una roja, una verde. Se ven idénticas para tu amigo.

Quieres probarle a tu amigo que las pelotas son de colores diferentes, pero **sin revelar cuál es roja y cuál es verde.**

Así es como:

1. Tu amigo sostiene ambas pelotas detrás de su espalda y aleatoriamente las intercambia (o no).
2. Te muestran las pelotas de nuevo y preguntan: “¿Las intercambié?”
3. Si las pelotas son realmente de colores diferentes, y no eres daltónico, siempre responderás correctamente.
4. Si las pelotas fueran del mismo color, solo adivinarías correctamente el 50% del tiempo.

Repite esto 20 veces.

Si respondes correctamente cada vez, tu amigo daltónico se convence: “La probabilidad de que adivines correctamente 20 veces seguidas por casualidad es 1 en 1,048,576. Debes realmente ver una diferencia en los colores.”

Has probado que las pelotas son de colores diferentes sin decir nunca cuál es roja o verde.

Tu amigo aprende: “Las pelotas son diferentes.”

Tu amigo NO aprende cuál es roja o cuál es verde—nada más allá de la verdad de la afirmación misma.

Esta es la esencia de las Pruebas de Conocimiento Cero.

16.3 El Avance Matemático

En 1985, tres investigadores (Shafi Goldwasser, Silvio Micali y Charles Rackoff) publicaron un artículo probando que las Pruebas de Conocimiento Cero son matemáticamente posibles.

Esto fue impactante. La mayoría de la gente asumía: “Para probar algo, debes revelarlo.” Pero las matemáticas dijeron lo contrario.

¿Recuerdas cómo obtuvimos identidades digitales en el Capítulo 6? Usamos **funciones unidireccionales**—operaciones matemáticas que son fáciles de calcular en una dirección pero casi imposibles de revertir:

- Fácil: $\text{hash}(\text{"password123"}) = \text{d3f8e9...}$
- Difícil: $\text{d3f8e9...} = \text{hash}(\text{"?????")}$ (revertirlo)

Las Pruebas de Conocimiento Cero usan matemáticas aún más avanzadas. Las matemáticas detrás de las ZKP se enseñan en cursos universitarios avanzados (criptografía de nivel de posgrado), programas de doctorado en matemáticas e informática, y laboratorios de investigación especializados.

Involucra: - Criptografía de curva elíptica. - Matemáticas polinomiales. - Teoría de grupos y álgebra abstracta. - Algoritmos probabilísticos. - Teoría de números avanzada.

Si tienes curiosidad y quieres entender todos los detalles técnicos, aquí está lo que necesitarías estudiar: - Aritmética modular. - Campos finitos. - Logaritmos discretos. - Criptografía basada en emparejamientos. - Compromisos polinomiales. - Heurística de Fiat-Shamir. - zk-SNARKs y zk-STARKs (construcciones específicas de ZKP).

Pero honestamente, esto requeriría otro libro completo, o libros.

Las matemáticas son extremadamente complejas— incluso la mayoría de los informáticos no entienden completamente todos los detalles. Pero nuevamente, cuanta más gente entienda esta tecnología, más descentralizado y más fuerte se vuelve el sistema, así que te animo a aprender todo lo anterior si te apetece. Te tomará algunos años si no tienes experiencia previa.

Lo que importa para este libro es entender qué pueden hacer las Pruebas de Conocimiento Cero, no realmente cómo lo hacen.

Así que enfoquémonos en las propiedades.

16.4 Las Tres Propiedades de las Pruebas de Conocimiento Cero

Para que algo sea una verdadera Prueba de Conocimiento Cero, debe tener tres propiedades:

16.4.1 1. Completitud

Si la afirmación es verdadera, un probador honesto puede convencer a un verificador honesto.

- Si realmente conoces la contraseña, puedes probarlo.
- Si realmente tienes \$100,000 en tu cuenta, puedes probarlo.
- Si realmente eres mayor de 21, puedes probarlo.

Una prueba correcta siempre funciona.

16.4.2 2. Solidez

Si la afirmación es falsa, ningún probador trámposo puede convencer al verificador (excepto con probabilidad negligible, como 0.0000000000000000000000000001%—no en la vida de un universo o varios de ellos).

- Si NO conoces la contraseña, no puedes falsificar la prueba (excepto teniendo extremada suerte).
- Si NO tienes \$100,000, no puedes engañar al verificador para que piense que sí.

No puedes mentir con una Prueba de Conocimiento Cero. O al menos, mentir es tan astronómicamente improbable que es efectivamente imposible.

16.4.3 3. Conocimiento Cero

El verificador no aprende nada excepto que la afirmación es verdadera.

- No aprenden la contraseña.
- No aprenden tu saldo exacto (solo que es $\geq \$100,000$).
- No aprenden tu fecha de nacimiento (solo que eres ≥ 21 años).

No se filtra información más allá de la verdad de la afirmación misma.

16.5 ¿Cómo Ayuda Esto a las Blockchains?

Las blockchains tienen un problema de transparencia.

Las Pruebas de Conocimiento Cero ofrecen una solución:

En lugar de enviar a la red: “Alice envió 5 BTC a Bob,” transmitemos:

“Ocurrió una transacción válida. Aquí está una Prueba de Conocimiento Cero de que: - El remitente tenía suficiente saldo. - Las cantidades son correctas. - No se crearon monedas de la nada. - La transacción sigue todas las reglas.”

La red puede verificar que la transacción es válida sin ver: - Quién la envió. - Quién la recibió. - Cuánto se envió.

Privacidad + Verificación. Ambas a la vez.

Incluso puedes tener privacidad selectiva: podrías no preocuparte por ocultar al remitente, pero querer ocultar la cantidad. O viceversa.

16.6 Ejemplos del Mundo Real

16.6.1 Ejemplo 1: Transacciones Privadas

Zcash es una criptomoneda que usa Pruebas de Conocimiento Cero para permitir transacciones privadas.

- Puedes enviar dinero a alguien.
- La red verifica que la transacción es válida.
- Pero nadie (excepto tú y el destinatario) sabe cuánto se envió o quién estuvo involucrado.

Libro mayor público. Detalles privados.

16.6.2 Ejemplo 2: Probar Que Eres Mayor de 18

Quieres entrar a un bar. El portero necesita verificar que eres mayor de 18.

Método tradicional: - Mostrar tu carnet de conducir. - El portero ve: tu nombre, dirección, fecha de nacimiento, foto, número de licencia, estado de donante de órganos, etc.

Con Pruebas de Conocimiento Cero: - Tu teléfono genera una ZKP que prueba: “Esta persona es ≥ 18 años.” - El portero la escanea y está convencido. - El portero aprende: “Esta persona es mayor de 18.” - El portero NO aprende: tu edad exacta, nombre, dirección, o cualquier otro detalle.

Divulgación mínima de información.

16.6.3 Ejemplo 3: Votación Privada

Quieres votar en una elección. El sistema necesita verificar: - Eres elegible para votar. - No has votado ya. - Tu voto se registra correctamente.

La votación digital tradicional tiene problemas: - Si los votos son públicos, no hay privacidad. - Si los votos son secretos, ¿cómo verificas que se contaron correctamente?

Con Pruebas de Conocimiento Cero: - Generas una prueba: "Soy un votante elegible, y emití un voto válido." - La red verifica la prueba. - Tu voto se cuenta. - Nadie sabe por quién votaste, pero todos pueden verificar que el conteo total es correcto.

Privacidad + Auditabilidad.

16.6.4 Ejemplo 4: Probar Solvencia Sin Revelar Saldos

Un exchange de criptomonedas afirma: "Tenemos suficientes fondos para cubrir todos los depósitos de usuarios."

Los usuarios quieren prueba, pero el exchange no quiere revelar: - Exactamente cuánto tienen. - Sus direcciones de billetera (riesgo de seguridad). - Saldos individuales de usuarios.

Con Pruebas de Conocimiento Cero: - El exchange genera una prueba: "Depósitos totales de usuarios = X. Tenencias totales del exchange \geq X." - Los usuarios pueden verificar la prueba. - Los usuarios aprenden: "El exchange es solvente." - Los usuarios NO aprenden: saldos exactos, direcciones de billetera, u otros detalles sensibles.

Transparencia sin exposición.

16.7 Por Qué Esto Importa

Esto es revolucionario. Durante la mayor parte de la historia humana, tenías que elegir:

O: - Transparencia: Todos pueden verificar todo, pero no hay privacidad.

O: - Privacidad: Guardas secretos, pero nadie puede verificar tus afirmaciones.

No podías tener ambas.

Los bancos son privados (no puedes ver los saldos de otros) pero no transparentes (no puedes auditar las reservas del banco—y si logras auditárlas, es un proceso largo y lento que es vulnerable a manipulación).

Las blockchains son transparentes (puedes verificar todo) pero no privadas (todos ven tu saldo).

Las Pruebas de Conocimiento Cero rompen este compromiso.

PUEDES tener tanto privacidad como verificación. Esto abre posibilidades completamente nuevas: - Sistemas financieros privados que aún son auditables. - Credenciales anónimas que aún son verificables. - Votos secretos que aún son probablemente contados correctamente. - Registros médicos confidenciales que aún pueden probar que estás vacunado.

Privacidad y confianza, juntas.

En la práctica, una Prueba de Conocimiento Cero representa una computación, un programa. Así que, en teoría, si algo puede computarse, puede probarse con conocimiento cero.

16.8 La Pega: Complejidad y Rendimiento

Las Pruebas de Conocimiento Cero son **increíblemente complejas y computacionalmente caras**.

16.8.1 Complejidad

Las matemáticas son tan avanzadas que la mayoría de los desarrolladores no las entienden completamente todavía, implementar ZKP correctamente es extremadamente difícil, y los bugs en sistemas ZKP pueden ser catastróficos (rompiendo la privacidad o la seguridad).

Muy pocas personas en el mundo pueden construir estos sistemas correctamente.

Esto crea una barrera. A diferencia de la criptografía básica (que ahora está bien entendida y estandarizada), las ZKP todavía son investigación de vanguardia. Solo empezaron a volverse prácticamente útiles en los años 2010, y todavía están evolucionando rápidamente.

16.8.2 Rendimiento

Generar Pruebas de Conocimiento Cero rápidamente requiere poder computacional significativo.

Pero están mejorando rápidamente.

Nuevos sistemas ZKP (zk-SNARKs, zk-STARKs, Plonky2, y más) están haciendo las pruebas más pequeñas, más rápidas, y más fáciles de generar.

Lo que tomaba minutos en 2015 ahora toma segundos en 2025 en muchos casos. Lo que toma segundos hoy podría tomar milisegundos en 2030.

El rendimiento está mejorando exponencialmente.

16.9 El Futuro: Coordinación Privada a Escala

Imagina un mundo donde:

- Puedes probar tus ingresos a un prestamista sin revelar tu salario exacto.
- Puedes probar tu historial médico a un doctor sin exponer detalles sensibles.
- Puedes votar en elecciones donde los resultados son públicamente verificables, pero tu voto es privado.
- Puedes hacer transacciones en una blockchain pública sin revelar tu saldo o historial de transacciones.
- Los gobiernos pueden probar que están siguiendo la ley sin revelar secretos de estado.
- Las empresas pueden probar que no están haciendo cosas ilegales sin revelar secretos comerciales.

Las Pruebas de Conocimiento Cero hacen todo esto posible.

Nos permiten construir sistemas que son:

- **Verificables:** Puedes probar que las afirmaciones son verdaderas.
- **Privados:** No revelas información innecesaria.

Mézclalo con RBDC para:

- **Sin necesidad de confianza:** No se necesita confiar en ninguna autoridad central.

Esta es una capacidad fundamentalmente nueva.

Antes de las ZKP, siempre tenías que elegir: transparencia o privacidad. Ahora puedes tener ambas.

Combinado con blockchains (libros mayores públicos, verificables, a prueba de manipulación), las Pruebas de Conocimiento Cero permiten **coordinación privada y verificable a escala global**.

16.10 Pero Todavía No Estamos Allí

Las Pruebas de Conocimiento Cero todavía son: - **Complejas:** Difíciles de construir correctamente. - **Lentas:** Computacionalmente caras. - **Nuevas:** Todavía no ampliamente entendidas o adoptadas.

La mayoría de los sistemas RBDC hoy NO usan ZKP. Bitcoin no. La cadena principal de Ethereum no (aunque las Capa 2 están empezando a hacerlo). Sin embargo, Ethereum está explorando activamente las ZKP para escalabilidad y privacidad, recientemente a principios de 2026, Vitalik, el inventor de Ethereum dijo que resolvieron el trilema de blockchain usando ZKP y ahora solo es cuestión de escribir el código de una manera muy segura.

¿Por qué? Porque son difíciles de implementar, más lentas que las transacciones regulares, y la tecnología todavía está madurando.

Pero el progreso es rápido. Lo que parecía imposiblemente lento en 2015 es práctico en 2025.

Las ZKP son una de las innovaciones más importantes en criptografía en los últimos 40 años.

Y apenas están empezando.

16.11 Una Nota sobre Ordenadores Cuánticos

Recuerda en el Capítulo 8 que hablamos sobre ordenadores cuánticos potencialmente rompiendo ciertos tipos de criptografía.

Lo mismo aplica a las Pruebas de Conocimiento Cero.

Algunos sistemas ZKP actuales podrían ser vulnerables a ordenadores cuánticos: - Los zk-SNARKs basados en emparejamientos de curvas elípticas podrían ser rotos por ordenadores cuánticos usando el algoritmo de Shor. - Estos se basan en problemas matemáticos (como logaritmos discretos) que los ordenadores cuánticos pueden resolver eficientemente.

Pero algunos sistemas ZKP se cree que son resistentes a cuánticos: - Los zk-STARKs usan funciones hash y no dependen de curvas elípticas o emparejamientos. - Las ZKP basadas en hash deberían permanecer seguras incluso contra ordenadores cuánticos. - La criptografía basada en retículos (otro enfoque) también se está explorando para ZKP post-cuánticas.

Las buenas noticias: El patrón se repite. Los investigadores están trabajando activamente en Pruebas de Conocimiento Cero resistentes a cuánticos. Para cuando los ordenadores cuánticos se conviertan en una amenaza real, probablemente habremos hecho la transición a sistemas ZKP seguros contra cuánticos.

Idea Clave: Las Pruebas de Conocimiento Cero te permiten probar que una afirmación es verdadera sin revelar ninguna información más allá de la verdad de la afirmación misma. Tres propiedades: completitud (las afirmaciones verdaderas pueden probarse), solidez (las afirmaciones falsas no pueden falsificarse), conocimiento cero (no se filtra información extra). Esto rompe el compromiso histórico entre transparencia y privacidad. Ahora puedes tener ambas: probar que tienes suficiente dinero sin revelar tu saldo, probar que eres mayor de 21 sin mostrar tu fecha de nacimiento, probar que una transacción es válida sin revelar quién la envió o cuánto. Las matemáticas son extremadamente avanzadas (criptografía de nivel de posgrado, polinomios,

curvas elípticas, álgebra abstracta), y la tecnología es computacionalmente cara, pero mejorando rápidamente. Combinado con blockchains, las ZKP permiten coordinación privada y verificable a escala global. Este es uno de los avances criptográficos más importantes de los últimos 40 años, y apenas está empezando.

A continuación, daremos un paso atrás y veremos el panorama general: ¿qué significan todas estas tecnologías para la sociedad, la coordinación y la libertad humana? ¿Cómo cambian las RBDC, los smart contracts, las Capa 2 y las Pruebas de Conocimiento Cero el panorama del poder, la confianza y la toma de decisiones colectiva?

Capítulo 17: Lo Que Todo Esto Significa Para La Humanidad

Esta tecnología hace posibles cosas físicamente imposibles.

Hemos cubierto mucho terreno juntos: bits, algoritmos, criptografía, mecanismos de consenso, blockchains, smart contracts, soluciones de Capa 2, y Pruebas de Conocimiento Cero.

Pero alejémonos por un momento.

¿Qué hemos construido realmente aquí?

Esto no es solo “dinero digital.” No es solo “smart contracts.” No es solo otra tendencia tecnológica.

Esta es tecnología de coordinación para extraños a escala de internet.

Y hace posibles cosas que eran literalmente, físicamente imposibles antes de **2009**, cuando se lanzó la red de Bitcoin.

Este es el comienzo de un nuevo capítulo en la historia humana que nos permite coordinar y alterar el comportamiento tradicional en algunas de nuestras estructuras societarias de formas que una vez sonaban imposibles e ilógicas.

17.1 Lo Que Era Imposible Antes

A lo largo de la historia humana, ciertas cosas requerían confianza en intermediarios centralizados.

¿Quieres enviar dinero a través de fronteras? Necesitabas bancos para verificar saldos y procesar transferencias. Podían congelar tu cuenta, revertir transacciones, cobrar comisiones, o negar el servicio completamente. No tenías otra opción más que confiar en ellos.

¿Quieres probar propiedad de un activo digital? Necesitabas un registro central—una compañía de juegos, oficina de títulos, o bolsa de valores. Controlaban tus activos, y si cerraban, tu propiedad desaparecía. En el mejor de los casos, podría volver después de retrasos largos y caros y batallas legales. La propiedad digital realmente no existía—solo permiso para usar.

¿Quieres coordinarte con extraños sin un mediador? Imposible. Alguien tenía que ser el árbitro de confianza. Los contratos requerían tribunales para hacerlos cumplir, y los acuerdos requerían intermediarios para verificar. Un humano en el medio era obligatorio.

¿Quieres crear dinero sin gobierno? Los gobiernos monopolizan la creación de moneda. Las monedas privadas o eran cerradas o requerían confianza en el emisor. El dinero era una cuestión de clases privilegiadas o control estatal.

¿Quieres probar algo sin revelarlo? Hasta las Pruebas de Conocimiento Cero, esto era matemáticamente imposible. La privacidad requería ocultar mientras que la verificación requería revelar—no podías tener ambas.

¿Quieres crear un sistema de votación que no pueda ser manipulado? La votación requería autoridades de confianza para contar y verificar. Las papeletas de papel podían perderse, alterarse o contarse mal. Un sistema de votación completamente probado matemáticamente y matemáticamente preciso y confiable era inalcanzable.

¿Quieres asegurarte de que los ganadores no reescriban la historia? Las bases de datos centralizadas podían ser alteradas o eliminadas por aquellos en control. Los registros podían cambiarse, borrarse u ocultarse. Una historia inmutable y auditabile estaba fuera de alcance.

¿Quieres reunir dinero para una nueva idea de negocio de muchos inversores alrededor del mundo? Necesitabas intermediarios—bancos, corredores, equipos legales. Las regulaciones lo hacían caro y lento. El crowdfunding sin permisos a escala global era inalcanzable.

Todas estas cosas eran físicamente imposibles.

Y luego, comenzando en 2009 con Bitcoin, lentamente se hicieron posibles.

No fáciles. No perfectas. Pero **posibles**. Eso ya es un salto de tamaño infinito.

17.2 ¿Qué Cambió?

Matemáticas + Ingeniería + Criptografía + Incentivos + Consenso Social = Coordinación Sin Necesidad de Confianza

Desglicemos eso:

17.2.1 Matemáticas

Las funciones criptográficas—hash, firmas, Pruebas de Conocimiento Cero—proporcionan **verdad verificable sin requerir confianza**.

No confías en mí en que la firma es válida; la verificas matemáticamente. No confías en mí en que el hash es correcto; lo calculas tú mismo. No confías en mí en que soy mayor de 18; verificas la Prueba de Conocimiento Cero.

Las matemáticas no mienten. Las matemáticas no tienen opiniones. Las matemáticas son iguales en todas partes.

17.2.2 Incentivos

La teoría de juegos alinea los intereses de los participantes.

Los mineros y validadores se benefician de seguir las reglas y pierden al romperlas. Los atacantes deben gastar enormes recursos para tener éxito, y incluso si lo hacen, destruyen el valor mismo que están atacando—solo las amenazas financiadas externamente tienen sentido.

El sistema asume egoísmo y matemáticamente lo convierte en beneficio colectivo.

17.2.3 Consenso Social

Al final del día, **los humanos deciden qué reglas seguir.**

El código no se hace cumplir por sí mismo—las personas eligen ejecutarlo. El valor no viene de la tecnología—viene de que las personas acuerden que tiene valor. El algoritmo de consenso no fuerza a nadie—coordina participantes voluntarios.

La tecnología permite. Los humanos deciden.

17.3 El Cambio en las Dinámicas de Poder

Esto crea un cambio fundamental en cómo se coordina el poder.

17.3.1 Antes: Guardianes Centralizados

Los bancos controlan tu dinero. Pueden congelar cuentas, revertir transacciones, y negar servicio.

Los gobiernos monopolizan la moneda. Pueden inflar la oferta, confiscar riqueza, y controlar el acceso sin mucha fricción.

Las plataformas poseen tus datos. Pueden censurar contenido, banear usuarios, y cambiar términos arbitrariamente.

Las corporaciones median acuerdos. Pueden cobrar renta, cambiar reglas, y cerrar servicios.

Tenías que confiar en ellos. No había alternativa.

Pero no seamos egoístas—estas organizaciones también tenían que confiar entre sí. No es sorpresa cómo los políticos a menudo se atacan entre sí; tampoco necesariamente confían entre sí. No es sorpresa que los corredores en Wall Street a menudo intentan estafarse entre sí o aprovecharse de la información del otro para beneficiarse.

Esta no es una revolución del pueblo a la clase gobernante—va más allá de eso. Es una revolución de la gestión de confianza misma de la que todos podemos beneficiarnos.

Pero como con todo avance tecnológico, será inútil si no aprendemos algunos básicos de uso y comprensión de las implicaciones. Como cuando se inventaron los coches, necesitábamos aprender a conducir y acordar reglas de tráfico similares que se apliquen mundialmente. Creamos una revolución en cómo nos movíamos.

Esta es la revolución en cómo confiamos.

Una revolución en uno de los aspectos más profundos que dictan cómo nosotros, como especie, nos coordinamos.

17.3.2 Despues: Coordinación Distribuida

RBDC (Blockchains): Ningún controlador único. Participación voluntaria. Reglas transparentes. Las matemáticas hacen cumplir los acuerdos.

Smart contracts: El código se ejecuta automáticamente. No se necesita intermediario. Nadie puede detenerlo una vez desplegado.

Auto-custodia: Controlas tus claves, controlas tus activos. Nadie puede congelar o confiscar sin tu clave privada.

Participación global voluntaria: ¿No te gustan las reglas? Haz fork. Sal. Únete a otra red. Nadie puede forzarte a quedarte.

No necesitas confiar en nadie. Verificas las matemáticas. Eliges en qué red participar.

Esto no elimina el poder ni ciertos fenómenos emergentes de la organización humana a escalas, como la formación de oligarquías. Pero sí cambia quién puede acceder al poder y cómo se negocia.

17.3.3 La Máquina Anti-Manipulación Psicológica

¿Recuerdas la máquina anti-manipulación psicológica del Capítulo 12?

Manipulación psicológica (Gaslighting) es cuando alguien te hace dudar de tu propia memoria o percepción de la realidad mintiendo repetidamente hasta que una “verdad” se acepta.

Los gobiernos hacen esto. Las corporaciones hacen esto. Los abusadores hacen esto.

Borran registros. Cambian documentos. Niegan que las cosas sucedieron. Y si no puedes probarlo, estás indefenso. Incluso si puedes probarlo, en el mejor escenario, todavía tienes que “rezar” para que los tribunales no estén sobornados—que la evidencia vital no se “ pierda” o “destruya accidentalmente” de alguna manera.

Y claro, todo esto asumiendo que no están poniendo un sistema completo de máquinas brillantes con textos cortos y emocionalmente cargados y videos para distraer tus pensamientos y, efectivamente, controlarlos.

De todos modos, las blockchains hacen todas estas cosas más fáciles de probar.

Si un gobierno intenta reescribir la historia, la copia de blockchain de todos todavía muestra la verdad.

Si una mayoría hace fork de la red (como Ethereum hizo con The DAO), la minoría puede mantener la cadena original (Ethereum Classic). El desacuerdo es visible para siempre.

Si una corporación afirma “nunca hicimos eso,” la blockchain dice: “El Bloque 1,920,000 prueba que lo hiciste.”

Nadie puede ocultar la tiranía. Nadie puede borrar el pasado. Al menos la historia económica.

Esta es una propiedad fundamentalmente nueva de los sistemas de coordinación humana.

17.4 Implicaciones del Mundo Real

Seamos concretos. ¿Qué permite esto realmente?

17.4.1 Dinero: No Puede Ser Congelado, Censurado, o Arbitrariamente Inflado

Bitcoin no tiene CEO. Nadie puede cerrarlo, congelar tu cuenta, o revertir tus transacciones.

Mientras controles tu clave privada, controlas tu dinero.

Los gobiernos no pueden inflar arbitrariamente la oferta de Bitcoin. El límite de 21 millones es hecho cumplir por código y consenso social, no por decisiones políticas.

Esto no significa que Bitcoin es dinero perfecto. Es volátil, lento, y difícil de usar para transacciones cotidianas.

Pero es **incensurable, inconfiscable, e no inflable**. Eso era imposible antes.

17.4.2 Gobernanza: Votación Transparente, Neutralidad Creíble, OAD

Las Organizaciones Autónomas Descentralizadas (OAD, en inglés DAO) usan smart contracts para coordinar grupos sin jerarquía tradicional.

Los miembros votan sobre propuestas. El código ejecuta decisiones automáticamente. Las reglas transparentes son hechas cumplir por matemáticas y código.

¿Es esto perfecto? No. Las OAD tienen problemas de gobernanza—apatía de votantes, plutocracia, fallos de coordinación. Nada nuevo bajo el sol aquí; la ley de hierro de la oligarquía todavía aplica.

Pero permiten **participación sin permisos en la gobernanza**. Cualquiera puede unirse. Cualquiera puede proponer. Nadie puede ser excluido arbitrariamente.

17.4.3 Identidad: Posee Tus Datos, Portables A Través de Plataformas

Ahora mismo, tu identidad está fragmentada. Facebook posee tu gráfico social. Google posee tu correo. Los bancos conocen tu historial financiero. Cada plataforma posee tus datos.

Con RBDC, puedes poseer tu identidad.

Tus credenciales, reputación, y datos viven on-chain o en Pruebas de Conocimiento Cero. Pruebas lo que necesitas probar sin revelar todo. Llevas tu identidad contigo a través de plataformas.

¿Está esto listo hoy? No del todo para uso de personas normales. Pero la infraestructura se está construyendo.

17.4.4 Coordinación: Capa 2 = Mini-Sociedades

Como aprendimos en el Capítulo 15, las Capa 2 permiten **coordinación anidada**—sociedades dentro de sociedades.

Diferentes comunidades pueden tener diferentes reglas mientras todavía se coordinan con la capa global cuando es necesario.

Así es como la sociedad humana ya funciona (ciudades, estados, naciones, acuerdos internacionales). Ahora podemos hacerlo con bases de datos.

17.4.5 Privacidad: Conocimiento Cero = Probar Sin Revelar

Como aprendimos en el Capítulo 16, las Pruebas de Conocimiento Cero permiten **privacidad con verificación**.

Puedes probar que eres elegible para votar sin revelar quién eres. Puedes probar que tienes suficiente dinero sin revelar tu saldo. Puedes probar que calificas para un préstamo sin revelar tu salario.

Dignidad + Libertad.

No tienes que elegir entre privacidad y participación.

17.5 Despedida de Flami: Dejando la Zona de Propaganda

¡Hola, soy Flami ! Estamos lentamente dejando la zona de propaganda. Espero que hayas disfrutado el viaje. Baches en el camino por delante, por favor abrocha tu cinturón intelectual.

Esta tecnología es poderosa. Sí permite cosas que eran literalmente imposibles antes, y por lo tanto representa progreso tecnológico e incluso intelectual.

Pero lo que has estado leyendo hasta ahora en este capítulo ha sido maquillado. Esta ha sido la reflexión de los sueños húmedos de la industria “cripto” (industria de RBDC).

Hay muchas advertencias. Muchas. Para propósitos de marketing, se omiten la mayoría del tiempo, pero son reales.

Tiene sentido que cuando introduces a la gente a nuevas tecnologías complejas que son revolucionarias, necesitas simplificar—a veces cayendo en el reino de la propaganda.

Pero eso no ayuda a medio y largo plazo.

El próximo capítulo te dará el control de realidad completo—todos los matices, limitaciones, y verdades incómodas sobre esta tecnología. Los compromisos. Los fracasos. Las exageraciones. La evaluación honesta de qué funciona, qué no, y qué todavía es incierto.

Antes de llegar allí, déjame dejarte con el núcleo filosófico:

17.6 El Núcleo Filosófico

Destilemos esto a primeros principios.

17.6.1 El Consenso Es Social, No Técnico

La tecnología no se hace cumplir por sí misma.

El código de Bitcoin dice “límite de 21 millones de monedas.” Pero si todos acordaran cambiarlo, podrían.

El algoritmo de consenso no fuerza a nadie. Coordina participantes dispuestos.

El código es ley, pero solo porque los humanos acuerdan ejecutar el código.

17.6.2 El Valor Es Consensual En Mundos Complejos

El oro tiene valor porque la gente acuerda que lo tiene. El Dólar estadounidense tiene valor porque la gente acuerda que lo tiene. Bitcoin tiene valor porque la gente acuerda que lo tiene.

El valor en grandes sociedades complejas predominantemente se convierte en consenso social. Siempre lo ha sido.

Algunas cosas tienen valor por su uso, como la comida, pero cuando las sociedades crecen y ya no ansiamos supervivencia, el valor principalmente se convierte en un constructo social para coordinación.

Bitcoin simplemente hace esto transparente. El valor no está oculto en decisiones de bancos centrales supuestamente completamente independientes o decretos gubernamentales. Es visible en el mercado, en los nodos, en los forks.

17.6.3 La Coordinación Es Voluntaria

Nadie puede forzarte a participar en una RBDC.

Eliges a qué red unirte. Eliges qué software ejecutar. Eliges cuándo salir.

Esto es fundamentalmente diferente de los sistemas tradicionales: - No puedes optar por no usar la moneda de tu gobierno (intenta pagar impuestos en Bitcoin). - No puedes optar por no seguir las regulaciones bancarias (intenta abrir un banco sin licencia). - No puedes optar por no seguir las reglas de plataforma (intenta negociar con los términos de servicio de Facebook).

Con RBDC, salir siempre es una opción. Haz fork. Únete a otra red. Comienza la tuya propia.

Esto no significa que no haya dinámicas de poder. Los efectos de red importan. Las personas con conocimiento tienen ventajas. Pero la **capacidad de salir y la capacidad de realmente poseer cambian el juego.**

17.6.4 La Tecnología Permite, Los Humanos Deciden

Las RBDC no tienen una opinión política.

Es una herramienta. Como el fuego, la electricidad, o internet.

Puedes usar Bitcoin para escapar controles de capital autoritarios. Puedes usar Bitcoin para evadir impuestos y financiar crimen. Puedes usar Bitcoin como inversión especulativa. Puedes usar Bitcoin como una declaración filosófica sobre el dinero.

De la misma manera puedes usar el Estado para crear buenos servicios públicos que eviten hambrunas o usarlo para robar fondos públicos y librarse de guerras innecesarias. Los sistemas son herramientas, las personas que los usan son las que añaden el factor ético.

Las personas que ejecutan Bitcoin sí tienen opiniones políticas. Como elegir que una oferta limitada es lo bueno.

La tecnología es neutral. Los humanos le dan significado.

No porque haya gobiernos corruptos o Estados que son dictaduras “malvadas” allá afuera debemos nunca usar la idea de un Estado en otros contextos. Lo mismo aplica para Bitcoin y RBDC, solo

porque algunas personas usan esos sistemas para cosas ilegales no significa que debamos rechazarlos completamente.

17.7 La Invitación

Esta no es una historia sobre “ellos”—los desarrolladores, los mineros, las instituciones.

Esto es sobre nosotros. Todos nosotros.

Porque las RBDC solo funcionan si las personas eligen participar. Solo tienen valor si las personas acuerdan que tienen valor. Solo cambian el mundo si las personas las usan para cambiar el mundo.

Puedes participar como quieras:

- **Construir:** Escribir código. Crear aplicaciones. Mejorar la infraestructura.
- **Criticar:** Señalar fallos. Identificar riesgos. Presionar por mejores soluciones. Denunciar estafas y hype.
- **Regular:** Trabajar en gobierno para crear política sensata.
- **Usar:** Enviar transacciones. Participar en OAD. Experimentar con la tecnología.
- **Educar:** Enseñar a otros. Escribir. Explicar. Difundir comprensión.

Solo hazlo con comprensión, no hype.

Haz las preguntas críticas: - ¿Este problema necesita una RBDC, o hay una solución más simple? - ¿El costo de una solución más compleja supera los beneficios? - ¿Me están vendiendo hype, o hay valor real aquí? - ¿Cuáles son los compromisos? ¿Cuáles son los riesgos? - ¿Quién se beneficia? ¿Quién pierde? - ¿Quiénes son los usuarios? ¿Estoy construyendo para personas reales con necesidades reales?

Sé escéptico. Sé curioso. Sé reflexivo.

17.8 El Mensaje Central

Este es avance tecnológico REAL.

No hype. No magia. No un esquema de enriquecimiento rápido.

Esta tecnología permite a los humanos hacer cosas que eran literalmente, físicamente imposibles antes:

- Coordinar confianza de una manera tan nueva que incluso necesitamos un nuevo término: “trustless” (sin necesidad de confianza).
- Poseer sin intermediarios.
- Probar sin revelar.
- Crear dinero sin gobiernos.
- Registrar historia sin autoridad central.

Estas son nuevas capacidades para la humanidad.

No las teníamos en 2008. Las tenemos ahora.

Pero con gran poder viene gran responsabilidad.

Esta tecnología puede usarse para el bien o el mal. Puede liberar o permitir crimen. Puede descentralizar el poder o concentrarlo de nuevas maneras.

Depende de nosotros—no desarrolladores, no gobiernos, no corporaciones—todos nosotros.

Lo que construyamos con estas nuevas capacidades definirá si esta tecnología se convierte en algo neto positivo o solo otra herramienta para los poderosos. Igual que con cualquier otra tecnología.

17.9 Hemos Desbloqueado Nuevas Formas de Coordinación

Durante la mayor parte de la historia humana, muchos procesos que requerían coordinación también requerían jerarquía **Porque alguien tenía que ser el árbitro de confianza.**

Las RBDC ofrecen una alternativa: **Coordinación a través de matemáticas, criptografía e incentivos que definen un consenso social.**

No perfecta. No siempre mejor. Pero **posible**. El árbol de jerarquía se aplana a su mínimo de una manera que es efectivamente “sin jerarquía” en la práctica, ya que ninguna acción racional dentro de la organización abusaría de sus privilegios—haciendo a todos efectivamente iguales dentro del grupo, solo con diferentes trabajos.

Con matices, es como un policía en organizaciones tradicionales. Los policías sí tienen privilegios, pero si rompen la ley, ser policía no los protegerá de ser procesados. El matiz es que un policía rompiendo la ley a veces no hace que todo el Estado colapse al día siguiente; en una RBDC, lo haría.

Y ahora que es posible, tenemos opciones que no teníamos antes.

Lo que elegimos construir depende de la humanidad.

Ahora te pregunto: ¿construirás algo? ¿Cómo lo usarás?

¿Qué elegirás?

Nuevamente, suficiente de los brillantes futuros potenciales—solo son potenciales por ahora. Veamos la realidad cruda en el próximo capítulo antes de concluir este libro.

Idea Clave: Esta tecnología permite nuevas formas de coordinación que eran literalmente imposibles antes de 2009. Cambia el poder de guardianes centralizados a redes distribuidas. Hace la historia auditável, los activos poseíbles, la privacidad verificable, y el dinero incensurable. La tecnología es neutral—los humanos deciden cómo usarla. Esto depende de todos nosotros: construir, criticar, regular, usar, o ignorarla sabiamente. Hemos desbloqueado nuevas capacidades. Lo que construyamos con ellas definirá si esto se convierte en algo neto positivo para la humanidad. Pero antes de celebrar, necesitamos un control de realidad—y eso es lo que viene a continuación.

Siguiente: El Capítulo 18 te dará el control de realidad completo y sin filtrar. El diablo está en los detalles, y Flami está a punto de mostrarte dónde están enterrados todos los cuerpos.

18

Capítulo 18: El Diablo Está En Los Detalles - Una Bofetada de Realidad

¡Hola, soy Flami! Bienvenidos a la bola de construcción. Hora de poner los pies en la tierra y bajar de las nubes.

¿Recuerdas el Capítulo 0? Te lo advertí.

Este libro, hasta ahora, se ha escrito con un **tono propagandístico**.

No todas las afirmaciones que has leído son 100% verdad. El diablo está en los detalles.

Las explicaciones técnicas—bits, algoritmos, criptografía, mecanismos de consenso—esas son precisas. Puedes confiar en ellas, son sin confianza.

¿Pero las **implicaciones sociales**, las **promesas**, el **potencial revolucionario**? Esas han sido simplificadas, pasadas por alto, y sí, a veces exageradas para mantenerte enganchado.

¿Por qué?

Porque las tecnologías revolucionarias son difíciles de introducir. Si hubiera empezado con “Bitcoin es lento, caro, difícil de usar, costoso ambientalmente, permite el crimen, puede concentrar la riqueza, y la mayoría de los proyectos son estafas,” habrías dejado de leer en la página 3.

Así que asumí el papel de un entusiasta “cripto”—“cryptobro” los llama alguna gente. Distingo “cryptobro” de “cryptotechbro”: un cryptobro se enfoca en la especulación de precios, no sabe nada o solo conoce detalles muy superficiales sobre la tecnología, y nunca ha escrito ni leído ningún código fuente real. Un cryptotechbro, por otro lado, entiende la tecnología profundamente y puede leer y escribir código, pero aún cree en el bombo y la propaganda alrededor de cripto sin dedicar mucho tiempo a pensar en las implicaciones sociales. Normalmente somos frikis, así que ¿por qué esperarías que pensáramos sobre interacciones sociales? Sí, el autor es uno de ellos haciendo el esfuerzo de pensar sobre implicaciones sociales.

De todas formas, te mostré el sueño. El potencial. Los escenarios del mejor caso.

Ahora es el momento de la bofetada de realidad baby.

Este capítulo abordará sistemáticamente qué he simplificado en exceso, qué he omitido, y dónde la propaganda diverge de la realidad.

Abrochense los cinturones. Esto va a ser incómodo.

18.1 Parte 1: ¿Es Bitcoin Realmente “Dinero”?

18.1.1 Lo Que Te Dije

“¿Podemos hacer que los bits signifiquen ‘dinero’? La respuesta es sí.” (Capítulo 9)

Bitcoin tiene las 8 propiedades del dinero: escasez, verificabilidad, divisibilidad, etc.

18.1.2 La Bofetada de Realidad

Bitcoin triunfa en algunas propiedades del dinero, lucha con otras.

Revisitemos esas 8 propiedades con una perspectiva equilibrada yin-yang:

1. **Escasez difícil de controlar:** **Éxito claro.** Límite de 21 millones impuesto por código y consenso social.
2. **Verificabilidad:** **Éxito claro.** Cualquiera puede verificar transacciones matemáticamente.
3. **No doble gasto:** **Éxito claro.** La innovación central de blockchain previene esto.
4. **Transferibilidad:** **Realidad mixta.** - **Yin (desafíos):** Eventualmente requiere internet, electricidad, y conocimiento técnico. No es tan simple como entregarle efectivo a alguien. Las barreras geográficas permanecen (acceso a internet, restricciones regulatorias). - **Yang (potencial):** La complejidad técnica se está reduciendo activamente a través de Lightning Network, mejores monederos, y UX (Experiencia del Usuario) mejorada. Este no es el problema más difícil que la industria ha resuelto—es solucionable con tiempo y esfuerzo de ingeniería. Además, Bitcoin es solo una RBDC específica; los beneficios de la tecnología no desaparecen porque una implementación específica luche. Sería como decir que si Toyota quiebra, toda la industria del automóvil colapsará.
5. **Propiedad: Condicional y compleja.** - **Yin:** Posees Bitcoin SI controlas las claves privadas, entiendes la seguridad, no pierdes las frases semilla, y no caes en estafas. Los analistas estiman que aproximadamente **10–20%** de todos los bitcoin pueden estar permanentemente perdidos (análisis recientes sugieren entre 2.3–4 millones de BTC perdidos). Incluso si tienes oro en tu casa pero tu casa es fácil de robar, ¿realmente lo “posees”? - **Yang:** La verdadera propiedad sin intermediarios es posible para aquellos dispuestos a aprender. Ningún banco o gobierno puede confiscar Bitcoin adecuadamente asegurado. La auto-soberanía efectiva tiene una curva de aprendizaje, pero es genuinamente alcanzable—no está ni cerca de aprender ciencia de cohetes.
6. **Fungibilidad:** **Teóricamente sí, desafíos prácticos.** - **Yin:** Las monedas de hackeos/crímenes (“monedas manchadas”) pueden negociarse con descuentos. El rastreo de direcciones vincula históricas de transacciones. Algunos exchanges rechazan Bitcoin “sucio”. La privacidad es más difícil de lo que parece. - **Yang:** El valor es valor. Incluso el dinero fiat del “mercado negro” se blanquea en moneda limpia—las reclamaciones éticas sobre el dinero son tan frágiles como la ética humana misma. El BTC manchado probablemente, como siempre ha ocurrido a lo largo de la historia, eventualmente volverá a transacciones lícitas a través de formas modernas de blanqueo.
7. **Divisibilidad:** **Éxito claro.** Hasta satoshis (0.00000001 BTC)—divisibilidad mucho mejor que el oro o el efectivo.
8. **Durabilidad:** **Depende del contexto.** - **Yin:** Bitcoin requiere infraestructura activa (nodos,

internet, electricidad). En el colapso de la civilización o fallo de internet, Bitcoin deja de funcionar. El oro es más duradero en eventos cataclísmicos. Bitcoin no puede funcionar sin conexión como algunas RBDC programables (Ethereum puede funcionar temporalmente a través de intercambios de firma incluso sin internet—la analogía de los “cheques” de capítulos anteriores). - **Yang:** En escenarios del día del juicio final, ¿realmente nos importará el oro tampoco? Si vuelve la paz, las bases de datos con datos de BTC podrían restaurarse y la red reiniciarse. Para el 99.99% de escenarios realistas (no apocalipsis), la durabilidad digital de Bitcoin (copias en todas partes) es en realidad superior a los activos físicos (que pueden ser destruidos, confiscados, o degradados).

9. Extra: Portabilidad Mover oro físicamente es difícil, o grandes pilas de efectivo. Sin embargo, cualquier cantidad de bitcoin puede moverse contigo con un pequeño papel o incluso nada en absoluto si logras recordar 12 palabras en orden dentro de tu cerebro.

18.1.3 Propiedades adicionales que Bitcoin NO tiene completamente (todavía):

9. Unidad de Cuenta: **Mayormente ausente hoy, potencialmente cambiando.** - **Yin:** Casi nadie pone precios en Bitcoin. Los precios se establecen en fiat, convertidos a BTC en el momento de la transacción. Esto crea complejidad contable y riesgo de volatilidad. - **Yang:** Esto es en última instancia una elección. Más personas pueden elegir lentamente mostrar precios en BTC en sus tiendas. El Salvador hizo de BTC moneda de curso legal—mostrando que es posible. A medida que la volatilidad disminuye con la madurez, poner precios en BTC se vuelve más viable.

10. Medio de Intercambio: **Limitado hoy, situacionalmente útil.** - **Yin:** Muy pocos comerciantes aceptan Bitcoin. Aquellos que lo hacen a menudo lo convierten inmediatamente a fiat. La volatilidad lo hace terrible para transacciones diarias en economías estables. - **Yang:** En países con monedas hiperinflacionarias (Venezuela, Argentina, Zimbabue), Bitcoin es mejor que el dinero local. Además, aunque no sea “efectivo,” puede usarse como el oro como depósito y transferencia de valor. Incluso si no es un dinero diario realista para la mayoría de los países, sigue siendo un activo económicamente útil.

11. Depósito de Valor: **Volátil pero mejorando.** - **Yin:** Bitcoin ha tenido múltiples caídas del 70-80%. Es más como una acción tecnológica especulativa que “oro digital”—al menos por ahora. - **Yang:** A medida que los mercados maduran, más personas adoptan, y la liquidez aumenta, la volatilidad tiende a disminuir (ver: oro, acciones, bienes raíces durante siglos). Si la adopción sigue creciendo, la volatilidad de Bitcoin probablemente continuará disminuyendo. Así es simplemente como se comportan los mercados.

18.1.4 La Verdad Equilibrada

Yin: Hoy, Bitcoin se usa principalmente como una **inversión especulativa**, no como dinero. La mayoría de los economistas y bancos centrales no lo consideran dinero, al menos no lo dicen públicamente, en el sentido funcional.

Yang: Esto es en última instancia una cuestión de adopción y elección. Si las personas acuerdan sobre el oro, también pueden acordar sobre Bitcoin—no hay nada fundamentalmente diferente sobre esa mecánica. De hecho, si tuviéramos un mercado Bitcoin de alta liquidez, sus propiedades lo hacen mejor y más cómodo de usar que el oro. Bitcoin es solo una implementación específica de

RBDC; las RBDC (Redes de Base de Datos Consensuadas) son mucho más profundas y útiles que solo Bitcoin. Como dice el dicho: “Bitcoin no es cripto, y cripto no es Bitcoin.” La tecnología tiene aplicaciones más amplias, y las limitaciones actuales de Bitcoin no definen el potencial de todo el campo.

18.2 Parte 2: El Problema de la Centralización

18.2.1 Lo Que Te Conté

“Todo el mundo tiene una copia de la blockchain. Ninguna persona individual la controla.” (Capítulo 9)

Bitcoin está descentralizado. No hay autoridad central.

18.2.2 La Bofetada de Realidad

El poder de Bitcoin está concentrado en la práctica, pero menos que en los sistemas tradicionales.

18.2.3 Concentración de minería:

- **Yin:** Las estadísticas recientes de pools muestran que los 4 principales pools de minería de Bitcoin controlan **del orden del 60-70% del hashrate**, aunque esto fluctúa (por ejemplo: Foundry ~32%, AntPool ~19%, F2Pool ~8%, SpiderPool ~6%). Los 10 principales pools controlan ~90%. Si estos pools se confabularan (o fueran coaccionados por gobiernos), podrían censurar transacciones, hacer doble-gasto o detener la red. Este es un riesgo de centralización real.
 - **Yang:** Cualquier minero puede abandonar un pool injusto y unirse a otro. En la práctica, los costes de cambio, la lealtad y los incentivos mantienen a los mineros concentrados—pero la opción existe. El camino hacia más descentralización está claro: más pools y más mineros. Ejecutar un nodo completo cuesta aproximadamente \$500-800 en hardware más electricidad. Si tienes capital, puedes convertirte en minero o reunir inversores para crear un nuevo pool. Una oligarquía descentralizada de muchos grupos pequeños es estructuralmente mejor que una dictadura. Y crucialmente: una oligarquía de mineros no es el final de la historia. Lo que Bitcoin *es* se decide en última instancia por TODOS los nodos, incluso los baratos. Un paisaje minero más descentralizado con más oligarquías fortalece la red contra ataques externos.
-

18.2.4 Concentración geográfica:

- **Yin:** La mayor parte de la minería de Bitcoin ocurre en unas pocas regiones con electricidad barata (históricamente China, ahora EE.UU., Kazajistán, Rusia). Los gobiernos PUEDEN atacar Bitcoin apuntando a estas instalaciones concentradas. La prohibición de minería de China en 2021 redujo el hashrate global en ~50% temporalmente—demostrando que este riesgo de centralización es real.

- **Yang:** Las principales regiones mineras (EE.UU., Rusia, China, Kazajistán) son actores geopolíticamente independientes—a menudo adversarios. No confían entre ellos por naturaleza, que es precisamente el problema que las RBDC vinieron a abordar. La distribución geográfica de la minería entre naciones rivales en realidad proporciona resiliencia: ningún gobierno individual puede cerrar unilateralmente la red.
-

18.2.5 Concentración de riqueza:

- **Yin:** El 2% superior de direcciones de Bitcoin posee ~95% de todo el Bitcoin. Los early adopters e insiders se hicieron extraordinariamente ricos a través de asimetría de información. La desigualdad de riqueza de Bitcoin (coeficiente de Gini) es PEOR que la de los sistemas financieros tradicionales. Esto crea desequilibrios de poder económico.
 - **Yang:** Nada nuevo bajo el sol. Los early adopters de internet también se hicieron extraordinariamente ricos (Google, Amazon, Microsoft, Meta/Facebook son ahora las empresas más valiosas del mundo). Es válido criticar la concentración de riqueza, pero esto no descubre ningún “mal nuevo”—es naturaleza humana y timing. Si tu sistema político es fácilmente corruptible por dinero, ¿es culpa de Bitcoin? ¿Deberíamos negar la mejora tecnológica, o deberíamos exigir mejor legitimidad institucional para mantener a la humanidad avanzando? Si arreglamos la corrupción institucional, la desigualdad de riqueza se convierte en una cuestión política manejable, no en una amenaza existencial. También vale la pena señalar: en Bitcoin, la riqueza no equivale a poder sobre el protocolo. El hashrate de minería y la operación de nodos determinan las reglas de la red, no la propiedad de monedas. Puedes tener cero BTC y aún así ejecutar un nodo que valide las reglas.
-

18.2.6 Concentración de staking de Ethereum:

Los validadores son el análogo a los mineros en la RBDC de Ethereum. Hacen algo llamado staking para probar su valía para escribir en la base de datos.

- **Yin:** Post-merge, los principales proveedores de staking (Lido, Coinbase, Kraken) controlan la mayoría del ETH en staking. Lido históricamente alcanzó un pico alrededor del **30-32%** del ETH en staking y a finales de 2025/principios de 2026 se sitúa en el rango de **mediados-20s%** (alrededor del **24-30%**). Los conjuntos de validadores están incluso más concentrados que la minería de Bitcoin. Una entidad controlando casi un tercio es preocupante. La ley de hierro de las oligarquías sigue apareciendo—¿podemos realmente crear una RBDC que esté verdaderamente descentralizada?
- **Yang:** ETH es solo otra RBDC, no Bitcoin—así que de nuevo esto no condena todo el campo. Para defender a ETH específicamente: Lido es una forma de *agregar* el ETH de stakers individuales—similar a los pools de minería pero en realidad más fácil de abandonar. Los individuos hacen staking *a través de* Lido; pueden retirar y cambiar a otros proveedores como Rocketpool. Desarrollar software de staking para Ethereum es relativamente barato, y mover fondos para este caso de uso es absurdamente barato onchain. La comunidad de Ethereum anuncia activamente la descentralización y advierte cuando ciertos clientes o plataformas de staking se vuelven demasiado dominantes—la comunidad se preocupa y responde. Las

personas que hacen staking ya demuestran alto conocimiento y cuidado—es más probable que actúen responsablemente. El ecosistema se autocorrege a través de presión social y educación.



Figura 18.1: Advertencia de diversidad de clientes de Ethereum

En la imagen de arriba podemos ver cómo un software que ejecuta Ethereum, Lighthouse, se está volviendo demasiado predominante y hay advertencias para dejar de usarlo tanto. Como dato curioso, el autor trabaja para la misma empresa que desarrolla el software Lighthouse.

Además, con respecto a las oligarquías, ahí es donde entra la magia de las RBDC. Como dijimos, con Bitcoin, una oligarquía de mineros no destruye el valor de la red porque en última instancia los datos son la fuente final de verdad, y eso está distribuido entre todos los nodos—que son mucho más baratos de ejecutar y en realidad muy descentralizados.

18.2.7 Concentración de desarrollo:

- **Yin:** Pequeños grupos de desarrolladores principales controlan la evolución del protocolo. El equipo de Bitcoin Core y la Ethereum Foundation ejercen una influencia enorme. Los usuarios técnicamente pueden rechazar actualizaciones, pero prácticamente siguen las recomendaciones de los desarrolladores. Esto crea dinámicas de “dictador benevolente”.
- **Yang:** Los usuarios PUEDEN rechazar actualizaciones—esto no es teórico. Los desarrolladores que se preocupan por el código pueden aumentar en número; es código abierto. Si la gente empieza a preocuparse más, más desarrolladores se unirán y escribirán implementaciones alternativas para disminuir la dependencia. Esta preocupación es más válida

mientras que “cripto” no esté ampliamente adoptado. Pero si/cuando crezca la adopción, tendrá sentido que las naciones usen dinero de los contribuyentes para pagar a excelentes desarrolladores que escriban sus propios clientes de Bitcoin/Ethereum, minimizando la dependencia de cualquier equipo único. Si la comunidad crece con conciencia, este riesgo se mitigará significativamente. Y crucialmente: esto no rompe las RBDC ahora mismo. Si algo falla, el software probado en batalla que ha estado funcionando durante años ya está disponible. Los nodos pueden simplemente cambiar rápidamente a una versión anterior u otro software. El sistema es resiliente a que desaparezca cualquier equipo único.

18.2.8 La Verdad Equilibrada

“Todo el mundo tiene una copia” es técnicamente cierto pero engañoso. **El poder NO está distribuido equitativamente.** El poder computacional y la riqueza sí concentran el control sobre escribir datos futuros. Pero no otorga influencia extra sobre lo que los datos actuales realmente son, ya que eso está almacenado por prácticamente cualquiera.

El sistema está **mucho menos centralizado que un banco**, pero **lejos de estar en el ideal de que todas las acciones estén completamente descentralizadas.**

- **Yin:** Bitcoin tiene riesgos de centralización reales que necesitan atención y mitigación continuas.
 - **Yang:** “Mucho menos centralizado que un banco” es una mejora significativa—una buena dirección. El sistema tiene mecanismos para la descentralización (forks, clientes alternativos, distribución geográfica, participación sin permisos) que los sistemas tradicionales no tienen. La descentralización perfecta puede ser imposible, pero la descentralización significativa—suficiente para resistir punto único de fallo, censura completamente arbitraria—es alcanzable y se mantiene activamente. Estas tecnologías no han hecho nada más que mejorar gracias a estos rasgos.
-

18.3 Parte 3: El Problema de Energía de Bitcoin

18.3.1 Lo Que Te Conté

“Usar energía como recurso escaso alinea incentivos. Si ese trade-off vale la pena es algo que debe decidir la sociedad.” (Capítulo 10)

18.3.2 La Bofetada de Realidad

18.3.2.1 Yin

El consumo de energía de Bitcoin es masivo y está creciendo.

La escala: - Bitcoin usa más electricidad que países enteros (comparable a Argentina, Países Bajos) - La red de Bitcoin consume del orden de **140-170 TWh por año** (las estimaciones recientes se agrupan alrededor de este rango; ~150 TWh es una cifra razonable de rango medio) - Las estimaciones de energía por transacción de Bitcoin **onchain** varían ampliamente, pero muchas la sitúan en el rango de **cientos a más de mil kWh** (ej., 700-1,400+ kWh dependiendo de la

metodología) - La huella de carbono depende de la mezcla energética (a menudo combustibles fósiles en regiones con energía barata)

El ciclo de retroalimentación precio-hashrate: - “Bitcoin es valioso, así que el gasto de energía está justificado” - Pero Bitcoin es valioso en gran medida debido a la especulación, no a la utilidad hoy - El gasto de energía es proporcional a la **competencia de hashrate**, que aumenta cuando la minería es rentable - Cuando el precio de Bitcoin se duplica, la minería se vuelve más rentable, más mineros se unen, el hashrate aumenta y el consumo de energía aumenta - Esto ocurre incluso si el volumen de transacciones permanece constante—la energía no está sirviendo a más usuarios, solo asegurando un valor especulativo más alto

Las externalidades ambientales: - Residuos electrónicos de hardware de minería obsoleto (los ASICs se vuelven obsoletos cada 1-2 años) - La minería se concentra en regiones con regulaciones ambientales laxas - El uso de combustibles fósiles es común (energía barata a menudo significa carbón o gas natural) - Estos costes se externalizan (la sociedad paga a través del daño ambiental, los mineros se benefician)

El problema de la comparación: - “¡Los bancos también usan energía!” — Cierto, pero los bancos sirven a miles de millones de usuarios diariamente con servicios diversos. Bitcoin sirve principalmente a millones de especuladores. - “¡La minería de oro usa energía!” — El oro tiene usos industriales (electrónica, odontología) y valor estético/cultural que abarca milenarios. El uso principal de Bitcoin hoy es la especulación. - Estas comparaciones son “y tú más”, no justificaciones para el gasto energético.

“La sociedad decide” es engañoso. La sociedad no decide de manera significativa el uso de energía de Bitcoin—está determinado por los mineros que maximizan beneficios dentro de dinámicas competitivas. No hay proceso democrático, ni voto, ni mecanismo de toma de decisiones colectiva.

La Prueba de Participación soluciona esto (Ethereum usa un 99,95% menos de energía), pero introduce diferentes compensaciones de seguridad y centralización.

18.3.2.2 Yang

Las fuentes de energía son un problema general de la humanidad, no específico de Bitcoin.

El principal problema—fuentes de energía que emiten carbono—es un desafío para toda la civilización que afecta a todas las industrias. Bitcoin no crea este problema; hereda la infraestructura energética existente. A medida que la energía renovable se vuelve más barata (lo cual está ocurriendo—la solar y la eólica son ahora las fuentes de electricidad más baratas en la mayoría de regiones), los mineros naturalmente migran a estas fuentes para maximizar beneficios. Culpar solo a Bitcoin por esto ignora el problema real: nuestra mezcla energética global.

Los residuos electrónicos también son un desafío tecnológico más amplio:

La obsolescencia del hardware no es única a Bitcoin. Los ASICs viejos pueden reutilizarse para otras tareas computacionales: ordenadores personales, PCs de gaming, entrenar modelos de IA locales. Sí, esto es una preocupación—pero es parte del problema más grande de los residuos electrónicos. ¿Cuántos iPhones de Apple se tiran cada año? ¿Cuántas nuevas versiones de iPhone se lanzan con

mejoras cuestionables? Los residuos electrónicos son una preocupación real en toda la tecnología, no solo en Bitcoin. Deberíamos abordarlo sistemáticamente, no mencionar solo una tecnología.

Contra la crítica de la especulación:

Sí, Bitcoin es principalmente especulativo hoy—pero el lector debería entender a estas alturas que **esto es una elección**. Puedes ayudar a cambiarlo. Si eres especulador, puedes cambiar tu perspectiva: “Voy a mantener Bitcoin porque estoy de acuerdo en que tiene valor para la coordinación, no solo para el beneficio.” La voz de todos importa. El valor es consenso social, y tú eres parte de ese consenso. Sé una voz más empujando por una utilidad real similar al oro, no solo especulación.

El análisis coste-beneficio depende del contexto:

Podrías decir, con respecto a las emisiones de CO₂ y otros costes monetarios... vale, los aviones contaminan, pero al menos son abiertamente útiles.

¿Valen los costes de Bitcoin el beneficio? Solo lo veremos cuando la gente lo use en su máximo potencial—como una forma nueva y mejor de oro digital, como una capa de liquidación neutral para acuerdos internacionales. La coordinación internacional programable puede traer un valor enorme. Incluso podemos decir, sin exagerar, que mejores mecanismos de coordinación pueden salvar vidas (previniendo conflictos a través de la interdependencia económica). ¿Vale cada vida humana salvada el coste de ejecutar RBDCs?

El problema con evaluar el coste-beneficio de RBDCs como Bitcoin es que estaríamos intentando medir el valor de generar una revolución en cómo confiamos—y eso es imposible de medir porque la forma en que las personas, entidades y naciones se hablan entre sí cambiaría fundamentalmente. El potencial es claramente enorme y difícil de cuantificar. Igual que con los desarrollos actuales en Inteligencia Artificial.

Este debate es eterno y normal:

Los beneficios de ciertas RBDCs como Bitcoin en relación a sus costes es un debate verdadero y continuo. Solo sé plenamente consciente de ambos lados. Este debate nunca terminará—es inherentemente humano: “¿Deberíamos seguir usando la herramienta X? ¿Deberíamos cambiar a la herramienta Y?” En última instancia, esto es coordinación humana clásica, nada especial a Bitcoin. Sigue con todos los productos. Se evalúan las compensaciones, la sociedad elige, la sociedad prueba la nueva tecnología, y si la elección trae un beneficio general, se adopta.

El problema principal con RBDCs como Bitcoin es que los beneficios son prácticamente imposibles de medir rápidamente. Necesitaríamos usar la tecnología durante una o dos generaciones y ver si la dependencia económica programable se correlaciona con una disminución de amenazas de guerra o incluso de las guerras mismas y una mejora en la calidad de vida globalmente.

18.4 Parte 4: Las Exageraciones

18.4.1 “Probar la propiedad de activos digitales sin registro central”:

Yin : Realmente no puedes separar la propiedad del mundo físico.

Al final del día, si tu casa es tu casa, es porque si alguien te la roba—como entra a vivir allí y dice que es suya—lo único que te hará recuperar la propiedad es mostrarle a la policía que en la base de datos o registros del Estado dice que es tuya.

La propiedad criptográfica no existe legalmente a día de hoy. La mayoría de la gente usa servicios de custodia (Coinbase, etc.), reintroduciendo intermediarios. El reconocimiento legal sigue sin estar claro—los tribunales pueden no hacer cumplir la propiedad criptográfica sin identidad legal.

Yang : A medida que crece la adopción, los parlamentos serán presionados para crear marcos legales que reconozcan la propiedad criptográfica. Lo que tenemos ahora son registros mejores e inalterables—puedes ver lo que está pasando con tu propiedad y detectar inmediatamente si algo turbio sucede con los registros de propiedad. Esa es la verdadera innovación.

Conclusión: Las RBDCs no vienen aquí a garantizar la propiedad física. Vienen a proporcionar registros transparentes e inalterables que pueden apoyar sistemas de propiedad.

18.4.2 “Coordinar sin intermediarios de confianza”:

Yin : La afirmación de que no hay intermediarios es falsa. Todavía estás confiando en los nodos.

Yang : Los nodos son tantos y están tan bien diseñados que la confianza está prácticamente garantizada a través de la descentralización. De aquí viene la nueva palabra “sin confianza”—no que no confíes en nada, sino que confías en un sistema diseñado para que ninguna parte individual tenga un incentivo racional para traicionarte jamás. La única amenaza real son grupos externos intentando romper el sistema desde fuera o infiltrar espías para romperlo desde dentro.

18.4.3 “Cosas físicamente imposibles ahora son posibles”:

Yin : “Físicamente imposible” es una exageración.

Yang : No se inventó física nueva—solo usos inteligentes de matemáticas y ordenadores (nuevas hazañas de ingeniería). Así que más precisamente: **técnicamente imposible** antes del avance. Pero si nadie sabía cómo construirlo, aunque los materiales existieran, era **prácticamente imposible** tenerlo—lo que, a todos los efectos, lo hacía sentir “físicamente imposible.”

18.5 Parte 5: Los Contratos Inteligentes No Son Tan Inteligentes Ni Imparables

18.5.1 Lo Que Te Dije

“Los contratos inteligentes son imparables. Nadie puede cambiar las reglas una vez desplegados. El código se ejecuta automáticamente.” (Capítulo 13)

18.5.2 La Bofetada de Realidad

Claves de administrador y mecanismos de actualización:

- **Yin:** Algunos contratos desplegados incluyen direcciones de “propietario” con privilegios especiales. Poderes comunes de administrador: pausar contrato, actualizar lógica, acuñar tokens, ajustar parámetros. Muchos “hackeos” de DeFi (Finanzas Descentralizadas) involucran claves de administrador comprometidas o desarrolladores maliciosos usando sus privilegios. “Imparable por defecto” debería ser “parable por defecto, con raras excepciones.”
 - **Yang:** Cualquiera puede crear la misma versión del código pero sin propietario. Los usuarios son completamente libres de interactuar con el código que consideren adecuado. Y, por defecto, los contratos inteligentes no tienen propietario. Es algo que un humano tiene que programar explícitamente.
-

Dependencias de oráculos:

- **Yin:** Los contratos inteligentes necesitan datos externos (precios, clima, resultados deportivos). Los oráculos son puntos de fallo centralizados. Si un oráculo es manipulado, el contrato se ejecuta incorrectamente a pesar de que el código esté “correctamente programado.” Confiar en oráculos reintroduce supuestos de confianza.
 - **Yang:** La innovación está en curso—RBDCs completamente especializadas en proporcionar datos (Chainlink es un ejemplo) se están desarrollando. Diseñar oráculos de confianza minimizada con RBDCs y alineación de teoría de juegos es un área activa de investigación y desarrollo, ya siendo usado para manejar con seguridad millones de dólares en valor.
-

Costes de gas:

- **Yin:** Desplegar contratos cuesta cientos o miles de dólares cuando Ethereum está congestionado. “Cualquiera puede desplegar” es técnicamente cierto, prácticamente falso para la mayoría de la gente. Interactuar con contratos también cuesta gas—operaciones complejas de DeFi pueden costar 50-200\$ en comisiones.
 - **Yang:** La Capa 2 (C2; en inglés Layer 2, L2) reduce drásticamente los costes. Está emergiendo software de código abierto para facilitar la interacción, junto con tutoriales por todo internet. Es cierto que no cualquiera ahora mismo es capaz de desplegar su propio banco sin conocimiento de ciencias de la computación y con 1 clic—pero eso es factible. Solo necesitamos 2 o 3 frikis para programarlo como proyecto de verano. Y eso no es comportamiento poco realista de frikis en absoluto. Además, las C1 también se están volviendo más y más baratas de usar a medida que pasa el tiempo y se crean mejoras tecnológicas.
-

Transparencia del código comprensión del usuario:

- **Yin:** “Transparente y auditabile” asume que los usuarios pueden leer código Solidity. El 99,9% de los usuarios no puede auditar contratos inteligentes. Incluso contratos auditados son explotados (ver: docenas de hackeos de DeFi auditados). La transparencia beneficia más a atacantes sofisticados que a usuarios promedio.
- **Yang:** La industria de la seguridad es muy rentable y necesaria, por lo tanto está creciendo. La oligarquía de verificadores de código está creciendo y volviéndose más descentralizada. Se

están desarrollando formas de pagar seguros sin problemas en caso de hackeo. El ecosistema está madurando y los contratos inteligentes se están volviendo más seguros cada año. La seguridad sigue siendo una gran preocupación que no debería ignorarse, pero la tendencia es clara y positiva.

La realidad: Los contratos inteligentes son poderosos, pero no son magia. Son código ejecutándose en sistemas distribuidos que, incluso si el sistema en el que se ejecutan es sin confianza, los contratos pueden tener código programado en ellos con supuestos de confianza, puntos de centralización y gobernanza social diferente.

Todavía no son lo suficientemente seguros para que la mayoría de la gente confíe ciegamente en ellos con grandes cantidades de dinero sin debida diligencia. Sin embargo, esta tendencia está mejorando con el tiempo y no es un desafío tan grande de superar. Como puedes ver, hay oportunidades de trabajo por todas partes para hackers dispuestos a ayudar a asegurar el ecosistema de contratos inteligentes y RBDCs del futuro. El autor es uno de esos hackers que día a día está ayudando a asegurar este ecosistema.

18.6 Parte 6: Las Capas 2 No Lo Resuelven Todo

18.6.1 Lo Que Te Dije

“Las Capas 2 permiten transacciones rápidas y baratas mientras que la Capa 1 proporciona seguridad. Lo mejor de ambos mundos.” (Capítulo 15)

18.6.2 La Bofetada de Realidad

La seguridad NO es seguridad de Capa 1:

- **Yin:** Hay muchos diseños técnicos para las C2, cada uno con su propio conjunto de desafíos, compromisos y complejidades. Algunos puede que no hayan sido examinados tan exhaustivamente por investigadores de seguridad (hackers que protegen el código en lugar de abusar de él), y por lo tanto podrían tener vulnerabilidades aún no descubiertas. El ecosistema todavía es joven y está madurando.
 - **Yang:** Todos los sistemas empiezan jóvenes y frágiles. Las C2 son incluso más jóvenes que Bitcoin. Este es un sistema muy ambicioso, por lo tanto llevará tiempo madurar. Pero la tendencia es clara: más gente está trabajando en ello, más dinero se está invirtiendo, más conocimiento se está acumulando. El ecosistema está madurando.
-

Los puntos de control de C1 no garantizan seguridad completa:

- **Yin:** Si los operadores de C2 no publican actualizaciones en C1, no hay “punto de control”. C1 no puede validar completamente la lógica de C2—solo almacena resúmenes. Si los operadores de C2 conspiran y publican información falsa, C1 no lo detectará a menos que los mecanismos de detección de fraude funcionen. Muchas C2 tienen controladores centralizados que pueden bloquear transacciones antes de que lleguen a C1.

- **Yang:** Las C2 están mejorando sus modelos de seguridad y volviéndose más descentralizadas, esforzándose por evitar puntos únicos de fallo. Por ejemplo, la decisión de publicar datos en C1 para crear un punto de control no depende solo de una entidad. L2Beat es una página web donde se critica a cada Capa 2 para advertir sobre sus puntos de centralización. Todo el ecosistema es consciente de esto y aplica presión y crítica constante a estas C2 para mejorar. Además, se están desarrollando técnicas de ingeniería novedosas para evitar este tipo de riesgos. En general, el ecosistema está madurando y las Capas 2 están volviéndose lentamente más sin confianza.
-

La complejidad de UX (Experiencia del Usuario):

- **Yin:** Los usuarios deben entender conceptos técnicos, gestionar múltiples carteras, confiar en contratos puente (muchos hackeados por cientos de millones), y pagar múltiples comisiones de transacción. “Dadnos tiempo y lo simplificaremos” ha sido el mantra durante años con progreso limitado. La complejidad no es solo pulir la UI (Interfaz de Usuario)—es sistemática (múltiples modelos de confianza, diferentes garantías de seguridad, liquidez fragmentada).
 - **Yang:** Aunque “lo simplificaremos” ha estado en marcha sin completarse, es porque hay problemas más difíciles que priorizar. Resolver problemas de UI/UX no es el problema más difícil al que se enfrenta esta industria ahora mismo. Los esfuerzos actuales están proponiendo estándares que facilitan a nivel técnico interactuar con múltiples bases de datos (cadenas), y su creación también llevará a la simplificación de UX. Además, es solo cuestión de tiempo y más gente trabajando para arreglar los problemas de UX. La industria es en realidad joven y pequeña.
-

Los efectos de red se fragmentan:

- **Yin:** La liquidez se divide entre C2 (tu dinero en una C2, el de tu amigo en otra = difícil de realizar transacciones). La atención de los desarrolladores se fragmenta (¿construir en qué C2?). Confusión del usuario (¿qué C2 debería usar?). El marco de “sociedades dentro de sociedades” es bonito, pero en la práctica es caos de coordinación.
 - **Yang:** Se están desarrollando formas de crear código que funcione en cualquier cadena (“escribe una vez, usa en cualquier lugar”). La confusión del usuario desaparecerá cuando mejore la UX—simplemente verán su dinero, su valor, y diferentes opciones de inversión. Los usuarios avanzados podrán hacer clic en detalles para ver en qué base de datos están y sus riesgos. Pero por ahora, ese no es el caso. De nuevo, sí, ahora es caos y desordenado, pero el ecosistema está madurando y estos problemas se están abordando.
-

Comparación con Linux y adopción:

- **Yin:** Dije “Linux funciona perfectamente y cualquiera puede usarlo.” Realidad: la cuota de escritorio de Linux todavía es un **pequeño porcentaje de un solo dígito**, pero recientemente ha subido a alrededor del **4–5%** globalmente (a partir de 2025, desde <3% en años anteriores). Linux tiene éxito en servidores (gestionados por profesionales) pero todavía lucha en la adopción del consumidor. Usar Linux como prueba de que “cualquiera puede usar software de código abierto complejo” contradice décadas de evidencia. “**Dadnos tiempo**

y os daremos libertad colectiva” es hopium (esperanza ilusoria que actúa como droga). Después de más de 15 años, los problemas no han disminuido—han aumentado.

- **Yang:** La gente PUEDE aprender a usar Linux si dedica unas pocas tardes de esfuerzo—5 días como máximo. No es tan complejo, pero la atención de la gente es un recurso escaso. Debemos hacer esfuerzos para conseguir que a la gente le importe esto, o hacerlo tan simple que la gente no necesite preocuparse por ello. Se está trabajando en ambos caminos. Si inviertes 5 días de tu vida aprendiendo una nueva herramienta que puede otorgarte más poder y libertad, con el tiempo incluso podrías recuperarlos debido a la sociedad más eficiente en la que vives—podrías obtener 6 días de vacaciones más adelante. Seguro, es un gran esfuerzo y debe hacerlo mucha gente, así que hasta que todos lo hagan, el beneficio probablemente no será visto por la mayoría de la gente. Las cosas que valen la pena requieren esfuerzo: aprender y enseñar, o hacerlo incluso más fácil para que otros aprendan. La “pereza” e ignorancia de las masas sobre lo que deberían priorizar aprender en sus vidas es un problema—pero puede superarse. La gente espera para cruzar las calles cuando hay un paso de peatones; la gente aprendió a usar smartphones—que puede parecer fácil ahora pero no lo es. Es un cambio masivo que hicimos (mira a una persona mayor intentando aprenderlo). Este no es tan masivo; solo necesitamos profesores, e incluso predicadores, pero principalmente profesores amables. Profesores desinteresados si es posible.
-

18.7 Parte 7: El Mito de la Ausencia de Confianza

18.7.1 Lo Que Te Dije

“No necesitas confiar en nadie. Las matemáticas no mienten. Coordinación a través de matemáticas, criptografía e incentivos.” (Capítulo 17)

18.7.2 La Bofetada de Realidad

“Sin confianza”, aunque novedoso y necesario, sigue siendo un término un poco engañoso. Todavía confías en muchas cosas.

En qué estás confiando:

1. Supuestos criptográficos:

- **Yin:** Asumir que las curvas elípticas no se rompen (los ordenadores cuánticos podrían romperlas). Asumir que las funciones hash son seguras (resistencia a colisiones de SHA-256). Asumir que no hay avances matemáticos que rompan la criptografía actual.
 - **Yang:** Para ser justos, el mundo entero funciona bajo este supuesto. La banca actual, los sistemas militares, tu teléfono—toda la criptografía moderna se basa en estos supuestos. Por lo tanto, no es un problema nuevo único de las RBDC o incluso de la humanidad. Sí, si un matemático brillante descubre una forma de romper esto mañana el mundo podría volverse un caos, perdón por revelarte este supuesto bajo el cual todos vivimos.
-

2. Implementaciones de código:

- **Yin:** Confiar en que los desarrolladores escribieron código sin errores (no lo hacen—existen hackeos de contratos inteligentes). Confiar en que no se introdujeron puertas traseras (¿puedes auditar millones de líneas de código?)
- **Yang:** El código sin errores, no solo en RBDC sino en general, es imposible—eso es cierto. Pero especialistas trabajan cada día para asegurarse de que los bugs sean pequeños y no impactantes. Únete a ellos si quieras, pero es muy difícil de dominar. El código se está volviendo cada vez más seguro. Todos usan y usarán este software, así que estará en el interés de todas las naciones hacerlo bien y asegurarse de que otras naciones no pongan código raro que pueda usarse contra ellos. Por lo tanto, hay un incentivo masivo para hacerlo bien. AAVE, un “banco en la blockchain,” ya está manejando miles de millones de dólares en valor de gente de todo el mundo. La idea de que el código contenga bugs y puertas traseras es una preocupación válida, pero es extremadamente complejo de explotar en realidad. Solo gente con mucho tiempo y mucho conocimiento puede hacerlo, como genios financiados por estados-nación. El software de espionaje en código cotidiano usado en dispositivos es algo real porque el código es privado y controlado por corporaciones que podrían ser sobornadas o simplemente engañadas. Pero esta vez, el código es infraestructura pública, por lo tanto todos los estados-nación que están en contra unos de otros se asegurarán de que nadie introduzca “hacks.” Seguro que lo intentarán, pero otros estarán mirando y notificando. Ahora solo son “unos pocos miles de millones” de dólares en juego, pero en el futuro, si la mayoría de las finanzas del mundo se mueven a RBDC, el incentivo para hacerlo bien será enorme para cualquiera.

3. Incentivos económicos:

- **Yin:** Confiar en que los mineros/validadores permanezcan económicamente racionales. Confiar en que el 51% del hashpower se mantenga honesto en el caso de Bitcoin. Confiar en que los incentivos continúen alineados—¿qué pasa si la economía cambia?
- **Yang:** Las RBDC actuales están mostrando claramente que los algoritmos de consenso están bien diseñados. Puede suceder que una recién creada no lo haga bien, es cierto, pero el conocimiento sobre cómo hacerlo ya está establecido. Solo actores externos pueden tumbar estos sistemas, ya sea desde ataques externos o infiltrándose desde dentro. Los incentivos económicos están bien diseñados y está probado que funcionan. La preocupación real que deberías tener son los atacantes externos y los nuevos diseños de incentivos en nuevas RBDC que no han estado funcionando durante tanto tiempo.

4. Consenso social:

- **Yin:** Confiar en que la comunidad no hará un hard fork en contra de tus intereses. Confiar en que los desarrolladores no introducirán cambios que te desagraden. Confiar en que los usuarios no abandonarán la red (causando un colapso en el valor).
- **Yang:** Este es un problema inherente a la naturaleza humana, no específico realmente de las RBDCs. Cada día que te despiertas confías en que internet funcionará, que nadie hará un golpe de Estado en tu país, que el parlamento no aprobará una ley que te perjudique. Estos son problemas que surgen de la coordinación social y el consenso—problemas muy antiguos con dinámicas muy antiguas. Si estas cosas suceden, o cambias de red o te adaptas. Claro, ten cuidado con las RBDCs recién creadas que están especialmente centralizadas; el riesgo es real, de la misma forma que tienes que ser cuidadoso si un Estado es una dictadura y un día el

dictador decide actuar en tu contra. Lo mejor que puedes hacer para asegurar tu tranquilidad es ver qué RBDCs tienen el historial más fiable de no alterar su consenso frecuentemente o rápidamente o arbitrariamente.

5. Infraestructura:

- **Yin:** Confiar en que internet permanezca disponible. Confiar en que la red eléctrica funcione. Confiar en que los nodos continúen funcionando. Confiar en que los exchanges/wallets proporcionen acceso.
 - **Yang:** Otra vez, el mundo entero depende de esto hoy. Respecto a los nodos, es como decir confía en que tu router en casa funcionará. Si una RBDC tiene pocos nodos, tiene más probabilidades de ser cerrada accidentalmente o maliciosamente—así que ten eso en cuenta. Por ahora, las grandes RBDCs casi nunca experimentan esto. Ethereum ha tenido **un tiempo de actividad casi perfecto** desde 2015, sin caídas importantes en toda la cadena (celebrando más de 10 años sin prácticamente ninguna caída en toda la red). Bitcoin ha logrado **~99.99% de tiempo de actividad** desde 2009, con solo dos incidentes significativos al inicio de su historia. Eso imita el tiempo de actividad de las tecnologías actuales de internet. Los exchanges centralizados solo tienen sentido en el mundo de hoy, pero en el futuro cuando todo el mundo esté onchain dejarán de tener sentido. Los DEXes (exchanges descentralizados, en inglés Decentralized Exchanges) ya existen y no necesitas permiso de nadie para usarlos. Además, existe mucho software de wallets, incluso de código abierto, y se está creando aún más. La probabilidad de que no puedas acceder a tu dinero debido a un problema con una wallet es ridículamente baja. **SOLO RECUERDA: SI NO SON TUS CLAVES, NO SON TUS MONEDAS.** ¿Una wallet deja de funcionar? Usas otra, activas tus claves, y mueves las monedas.
-

6. Oráculos y puentes:

- **Yin:** La mayoría de estas nuevas finanzas descentralizadas (DeFi, Decentralized Finance) dependen de oráculos de precios (Chainlink, etc.)—puntos de confianza centralizados. Los puentes entre cadenas requieren confiar en contratos de puentes (muchos han sido hackeados).
 - **Yang:** Como dijimos antes, se están haciendo muchas innovaciones para hacer los oráculos y puentes más sin confianza. El ecosistema está madurando y estos problemas se están abordando. Chainlink ha estado funcionando durante años proporcionando precios fiables y precisos y más datos del mundo a estas redes, y hay más oráculos. Los sistemas funcionan de una manera de confianza minimizada. Si tienes curiosidad, también suelen ejecutar RBDCs locales con sus clásicos incentivos económicos garantizados por teoría de juegos y criptografía para asegurarse de que nadie desde dentro haga trampas.
-

Conclusión:

De hecho, el término preciso es “de confianza minimizada” (reduciendo los requisitos de confianza), no “sin confianza” (que parece implicar eliminar la confianza por completo). Pero para ser honesto, toda interacción humana se basa eventualmente en la confianza. Sin confianza es imposible para la naturaleza humana. La idea de diseñar sistemas tecnológicos que sistemáticamente

reducen los factores que necesitan confianza es todavía nueva y debería tener una palabra. Tú eliges para ser sincero—la industria es nueva y el término que se quedará es el que la gente eventualmente use.

Yo prefiero confianza minimizada. ¿Cuál prefieres tú?

18.8 Parte 8: El Problema del Caso de Uso

18.8.1 El Elefante en la Habitación

Después de más de 15 años, **el caso de uso principal de las criptomonedas es la especulación**, no las aplicaciones revolucionarias prometidas.

Para qué la gente realmente usa las criptomonedas:

1. **Especulación/Inversión (más del 90% del volumen):** - Comprar/mantener Bitcoin esperando que el precio suba, no usándolo como dinero - Comerciar con monedas volátiles llamativas (como las llamadas Meme coins) para hacer un beneficio rápido e irse
2. **Actividad ilícita (porcentaje significativo pero en declive):** - Pagos de ransomware - Mercados de la deep web - Blanqueo de capitales - Evasión de impuestos - Evasión de sanciones - Esquemas Ponzi que son claramente estafas
3. **Utilidad real (~1%):** - Remesas internacionales (algo de uso, pero las comisiones + volatilidad limitan la adopción) - Resistencia a la censura (activistas en regímenes autoritarios—real pero volumen diminuto) - Acceso financiero para personas sin bancos (mayormente aspiracional, éxito limitado en el mundo real)

Para qué las criptomonedas NO se usan ampliamente: - Comprar bienes/servicios directamente (prácticamente cero adopción por comerciantes) - Pagar a empleados (raro, principalmente empresas cripto) - Almacenar ahorros (demasiado volátil o arriesgado debido a hackeos) - Funciones bancarias tradicionales (préstamos, hipotecas, etc.)

La brecha entre la promesa y la realidad es enorme.

El libro presentó casos de uso como si fueran la realidad actual. La mayoría son aspiracionales en el mejor de los casos, fantasía en el peor.

Yin: Después de más de 15 años, el uso principal sigue siendo la especulación. Las aplicaciones revolucionarias no se han materializado a escala. La mayoría de las promesas siguen siendo aspiracionales. Las actividades ilícitas representan un uso significativo, lo que plantea serias preocupaciones sobre facilitar el crimen.

Yang: El cambio que esta tecnología trae es profundo—redefinir la arquitectura de confianza es un cambio civilizatorio colosal que involucra a todos, desde ciudadanos, hasta desarrolladores, hasta gobiernos. Esto llevará generaciones, no solo unos pocos años.

Sobre actividades ilícitas: Toda tecnología transformadora se usa como arma inicialmente. Las armas, internet, el cifrado—todos permitieron formas más sofisticadas de cometer crímenes. La deep

web facilita mercados ilegales, pero no abandonamos internet. Aprendimos a gestionar los riesgos mientras capturábamos los beneficios. La mayoría de la gente usa internet legalmente hoy. Las RBDCs seguirán el mismo patrón: la sociedad desarrollará marcos para mitigar el abuso mientras preserva las ventajas.

Sobre las expectativas de tiempo: Cambiar instituciones, hábitos y entendimiento colectivo requiere tiempo y esfuerzo colectivo. Si esperas el progreso clásico de la mentalidad de desarrollador de “muévete rápido, rompe cosas”, te frustrarás. La tecnología de minimización de confianza exige un desarrollo cuidadoso, educación, marketing y construcción de consenso social a través de toda la sociedad—desde las naciones más democráticas hasta las más autoritarias.

Sobre la transición especulación-utilidad: La actividad especulativa actual, aunque no es el objetivo final, está financiando el desarrollo de infraestructura. Las primeras empresas de internet también se construyeron sobre inversión especulativa antes de probar su utilidad. La diferencia entre la especulación financiando el desarrollo y la especulación como único propósito es que el desarrollo está realmente sucediendo: se están construyendo Capas 2, la UX de las wallets está mejorando, están emergiendo marcos regulatorios. Si este desarrollo lleva a una utilidad generalizada en 5, 10 o 30 años sigue siendo incierto—pero el trabajo continúa.

Ten paciencia, resiste la tentación del juego, difunde conocimiento, y ayuda a construir esta tecnología. Sí, es difícil no sonar como una secta cuando construyes tecnología revolucionaria—lo siento. Puedes hacer más que esto, más en la siguiente sección.

18.9 Parte 9: El Problema de la Ideología

18.9.1 Lo Que Te Dije

“Bitcoin no tiene una opinión política. Es tecnología neutral. Los humanos le dan significado.”
(Capítulo 17)

18.9.2 La Bofetada de Realidad

Yin: Te dije esto por el bien de la simplicidad, el marketing y el compromiso del lector. Pero es engañoso. Las RBDCs se construyen sobre consenso social, por lo tanto incorporan las ideologías de sus comunidades.

La capa tecnológica en sí puede ser neutral, pero **cómo la aplicamos y qué valores codificamos en ella no lo son.**

Bitcoin es profundamente ideológico. Bitcoin NO es neutral—es tecnología con opinión, sostenida por una comunidad que valora principios específicos como suministro limitado, resistencia a la censura y política anti-inflacionaria.

Yang: Sin embargo, esto no significa que la tecnología sea inherentemente malvada o rota. Todas las instituciones humanas incorporan valores—eso es inevitable. Lo que importa es: 1. **Transparencia:** Los valores de Bitcoin son explícitos y visibles en su código 2. **Elección:** Puedes hacer un fork o crear alternativas con valores diferentes 3. **Participación:** Cualquiera puede unirse a la conversación sobre cuáles deberían ser esos valores

Los sistemas financieros tradicionales también incorporan ideologías: economía keynesiana, control del banco central, vigilancia... Y no puedes escapar de ellos tan fácilmente en absoluto.

18.10 Parte 10: Entonces, ¿Qué Deberías Hacer?

Deberías tener suficiente conocimiento para pensar por tu cuenta ahora, pero aquí hay algunos consejos prácticos:

- 1. Solo arriesga lo que puedas permitirte perder.** - Las criptomonedas son altamente especulativas y volátiles hoy - No uses criptomonedas para ahorros—usa instrumentos tradicionales estables y aburridos en su lugar - Trátalo como boletos de lotería o fichas de casino, sin jugar
- 2. NUNCA compartas frases semilla.** - Si alguien te pide tu frase semilla, te están estafando. Incluso si literalmente es el CEO real de una empresa de buena reputación. - Si no son tus claves, no son tus monedas.
- 3. Si experimentas, quédate con RBDCs establecidas.** - Bitcoin o Ethereum son las menos estafadoras (todavía arriesgadas) - Evita altcoins, meme coins, tokens nuevos (el 99.99% de ellos son estafas separadas o no sus creadores)
- 4. Sé profundamente escéptico.** - La mayoría de proyectos cripto son estafas, rug pulls o Ponzi's - ¿Rendimientos demasiado buenos para ser verdad? ¿986% TAE en mis cripto-dólares? Podría desaparecer mañana - ¿Endorseos de celebridades? Usualmente promociones pagadas en las que la celebridad, o su familia para hacerlo más discreto, obtienen una porción en un algoritmo de juego de suma cero moviendo el precio - Si no entiendes la tecnología, no inviertas - La complejidad y la redacción rara, o el exceso de palabras, a menudo esconde estafas
- 5. Usa las finanzas tradicionales para la mayoría de cosas ahora.** - Bancos, tarjetas de crédito, brokers regulados tienen problemas pero también protecciones - Funcionan mejor para la mayoría de gente, la mayor parte del tiempo - Si no tienes educación o no puedes autocontrolarte, la protección es en realidad algo bueno. Las buenas noticias son que esos dos rasgos pueden mejorar con tiempo y esfuerzo
- 6. Sin embargo, adopta lentamente y experimenta cuidadosamente.** - Acepta pagos en stablecoins en tu tienda si tienes una - Usa bancos descentralizados para obtener rendimiento en el efectivo que no necesitas inmediatamente - VISA/Mastercard ya permiten gasto cripto indirecto incluso si los comerciantes no lo aceptan - Paga a amigos por cenas vía wallets para sentirte cómodo usándolas - Mantén tu patrimonio principal fuera de cripto, pero pon algo en ello para aprender. Recuerda, dinero que realmente no necesitas. - La tecnología depende de la adopción; estás construyendo un futuro más eficiente - Comparte conocimiento con amigos—educa sin ser molesto - Aprende lentamente a mejorar la forma en que almacenas tus claves: en lugar de papel usa placas de titanio, o encriptalas en múltiples USBs, compra una hardware wallet... - Aprende a ejecutar un nodo de tu RBDC favorita y hazla aún más segura mediante descentralización.
- 7. Si crees en la tecnología, contribuye significativamente.** - No solo especules o experimentes con cambio suelto - Construye, educa, invierte en o contribuye al ecosistema - Preocúpate por el potencial real para la humanidad
- 8. Mantente humilde y sigue aprendiendo.** - Nadie conoce el futuro - Puede que no veamos adopción masiva en nuestra generación - “El número sube para siempre” no es una tesis de inversión

inteligente - Lee escépticos Y creyentes - Forma tus propias conclusiones

18.11 Cerrando La Bola de Demolición

Las RBDCs son: - Tecnología real - Capacidades novedosas - Prometiendo en exceso y entregando por debajo (como todos los marketers necesitan en algún punto) - Problemas sin resolver (como todas las tecnologías) - Uso mayormente especulativo hoy (tú, literalmente tú, puedes cambiar eso) - Facilitación del crimen (como todas las herramientas) - Complejo de usar actualmente (como todas las tecnologías al principio) - Futuro incierto (como con cualquier innovación)

18.11.1 Me gustan los trenes.

Cuando se inventaron los trenes en los años 1800, imagina decirle a la gente: "Los trenes son lentos ahora, pero si seguimos invirtiendo, ¡te llevarán de Tokio a Osaka en 3 horas en lugar de 10! Debemos seguir construyendo esta tecnología de 'tren'—claramente es el futuro del transporte."

Algunas personas te habrían llamado, comprensiblemente, loco o estafador.

Sin embargo, avanzando rápido a través de innovaciones en múltiples campos, y hoy tenemos trenes de alta velocidad que hacen exactamente eso.

Las RBDCs están en una fase temprana similar—los primeros trenes de una revolución de gestión de confianza. Pero a diferencia del viaje de 200 años desde máquinas de vapor hasta trenes bala, tenemos ventajas: coordinación global de internet, procesamiento de información potenciado por IA, y conocimiento de ingeniería acumulado. Grandes saltos en innovación ahora pueden tomar solo 20 años, tal vez menos.

18.11.2 La Verdad Equilibrada

Para la mayoría de gente, la mayor parte del tiempo, los sistemas tradicionales funcionan mejor ahora mismo.

Pero para algunas personas en algunas situaciones—activistas bajo regímenes autoritarios, personas en economías hiperinflacionarias, aquellos construyendo aplicaciones sin permisos—las criptomonedas son genuinamente valiosas hoy.

Más importante aún: La tecnología es real. Las capacidades son novedosas. El potencial sigue siendo enorme. Si ese potencial se realiza depende de nosotros—constructores, educadores, críticos y adoptadores—eliendo trabajar hacia el mejor futuro mientras reconocemos y arreglamos los problemas presentes.

Este libro te mostró el sueño. Este capítulo te mostró la realidad.

Ahora puedes decidir por ti mismo.

Has visto ambos lados: el potencial revolucionario y las duras limitaciones. Entiendes la tecnología lo suficientemente profundo como para tomar decisiones informadas. Conoces los riesgos y las posibilidades.

Lo que hagas a continuación depende de ti.

¿Construirás? ¿Educarás? ¿Experimentarás cuidadosamente? ¿Esperarás y observarás? ¿Criticarás constructivamente?

Solo hazlo con los ojos bien abiertos.

Bienvenido al otro lado de la bola de demolición.

— Atentamente, Flami, el flamenco realista.



Figura 18.2: Flami posando para ti

Siguiente: El libro está casi terminado, comencemos la conclusión. El Capítulo 19 recapitulará todo el viaje—desde bits hasta Bitcoin, desde criptografía hasta coordinación. Verás lo que has aprendido y lo que puedes construir con ese conocimiento.

19

Capítulo 19: Del Bit al Bitcoin - Resumen Final

Capítulo “mira qué lejos has llegado”.

¿Recuerdas dónde empezamos?

Un interruptor de luz—encendido o apagado, 1 o 0. Un solo bit, la unidad más fundamental de información digital.

Y ahora, 18 capítulos después, entiendes tecnología de coordinación global que hace posibles cosas que eran técnicamente imposibles antes de 2009.

Construiste este entendimiento desde primeros principios.

Sin jerga vacía. Sin “confía en mí, es complicado.” Sin atajos.

Empezamos con un bit y construimos, capa por capa, concepto por concepto, hasta que intuitivamente entendiste Bitcoin, Ethereum, smart contracts, Capa 2, Pruebas de Conocimiento Cero, y las implicaciones filosóficas de todo esto.

Esto es un logro. Tómate un momento para apreciarlo.

La mayoría de la gente—incluso gente inteligente y educada—no entiende esta tecnología. Escuchan palabras de moda, ven hype, se sienten confundidos, y se rinden.

Pero tú no. Persististe. Aprendiste.

Y ahora entiendes muchísimo mejor.

19.1 El Viaje

Tracemos el camino que recorrimos juntos.

19.1.1 Parte 1: Fundamentos - ¿Qué Es La Información?

Capítulo 1: El Bit - Todo lo digital son patrones de interruptores encendido/apagado, y los ordenadores son miles de millones de transistores diminutos trabajando en concierto. - Toda la complejidad se construye desde esta base simple: 0 y 1.

Capítulo 2: Logos - Mapeando Significado a Números - Los humanos asignan significado a patrones—ASCII mapea 65 a ‘A’, 66 a ‘B’, y así sucesivamente. - Tu nombre existe en binario, y

las imágenes, videos, y sonido son todos solo números. - **Idea clave:** La información es significado en el que acordamos.

Capítulo 3: Algoritmos - Siguiendo Reglas - Los algoritmos son solo instrucciones (recetas), y los ordenadores las siguen perfectamente a velocidad increíble. - Son deterministas: la misma entrada lleva a la misma salida, siempre. - **Idea clave:** Los ordenadores no “piensan”—siguen reglas.

Capítulo 4: Protocolos - Ordenadores Hablando - Un protocolo es un conjunto acordado de reglas para comunicación. - HTTP, TCP/IP—internet son protocolos hasta el fondo. - **Idea clave:** Los protocolos son acuerdos sociales entre máquinas.

19.1.2 Parte 2: Confianza y Criptografía - ¿En Quién Puedes Confiar?

Capítulo 5: El Problema de la Confianza - Alice quiere decirle un secreto a Bob, pero Carol está escuchando en el canal público. - **Idea clave:** Internet es público por defecto.

Capítulo 6: Cifrado Simétrico - Secretos Compartidos - El cifrado César desplaza letras por 3, y el cifrado moderno funciona similarmente: $f(\text{mensaje}, \text{contraseña}) = \text{mensaje_codificado}$. - **El problema:** ¿Cómo compartes la contraseña sin que Carol la intercepte?

Capítulo 7: Cifrado Asimétrico - El Truco de Magia - Las funciones unidireccionales son fáciles en una dirección pero imposibles de revertir. - Dos claves trabajan juntas: una clave pública (el candado) y una clave privada (la llave). - Las firmas digitales prueban identidad sin revelar secretos. - **Idea clave:** Puedes probar quién eres sin dar tus secretos.

Capítulo 8: Computación Cuántica - ¿La Amenaza Futura? - Lo que se rompe: RSA y ECDSA (estándares criptográficos actuales). - Lo que permanece seguro: Criptografía post-cuántica. - Línea temporal: Se estima que los ordenadores cuánticos capaces de romper la criptografía actual siempre están a 10-20 años de distancia. - **Idea clave:** La amenaza cuántica es real pero manejable si la tomamos en serio y nos preparamos consistentemente.

19.1.3 Parte 3: Redes de Consenso - El Avance

Capítulo 9: Dinero Como Bits Sincronizados - ¿Qué hace algo dinero? Ocho propiedades incluyendo escasez, verificabilidad, y resistencia al doble gasto. - Los bits pueden tener estas propiedades, pero la pregunta es: ¿dónde los almacenas? - **La visión:** Distribuir la base de datos para que todos tengan una copia. - **El problema del consenso:** ¿Cómo sincronizas sin una autoridad central?

Capítulo 10: Proof-of-Work - Ganando el Derecho a Escribir - ¿Quién obtiene el derecho de escribir en la base de datos? No puedes simplemente votar—los ataques Sybil lo hacen imposible. - **Proof-of-Work:** Resolver un rompecabezas computacional quemando electricidad para probar que has hecho trabajo real. - Las recompensas de minería combinan monedas nuevas con comisiones de transacción, y los incentivos se alinean: honestidad = ganancia. - **Idea clave:** El liderazgo temporal se gana a través del trabajo y rota entre participantes.

Capítulo 11: La Blockchain - Encadenando Historia - El problema de solución simultánea surge cuando dos mineros resuelven el rompecabezas a la vez. - La solución elegante: déjalos competir, y la cadena más larga gana. - **La estructura de datos blockchain** hace imposible falsificar el tiempo. - **Nombre apropiado:** Redes de Base de Datos Consensuadas (RBDC) un tipo

de Tecnología de Sincronización de Datos Descentralizada. - **Idea clave:** Blockchain es historia a prueba de manipulación.

Capítulo 12: La Estructura de Datos Blockchain (Profundización Técnica Opcional) - Es una especie de “lista enlazada hecha de punteros hash”, y su propiedad de prueba de manipulación significa que cambiar un bloque rompe todos los bloques subsiguientes. - ¿Por qué no puedes simplemente recalcular? Porque eso requiere rehacer todo el Proof-of-Work. - La máquina anti-manipulación psicológica: todos tienen una copia, así que los cambios no pueden ocultarse. - **Idea clave:** Los detalles técnicos de cómo blockchain previene hacer trampa.

19.1.4 Parte 4: Evolución e Implicaciones - Lo Que Todo Esto Permite

Capítulo 13: Ethereum - La Máquina de Computación Consensuada - Bitcoin es oro digital con transacciones simples, pero Ethereum preguntó: *¿y si la base de datos pudiera ejecutar programas?* - Los smart contracts son lógica si-entonces hecha cumplir por código, y la Máquina Virtual de Ethereum (EVM) asegura que cada nodo ejecute los mismos programas. - El gas explica por qué la computación cuesta dinero. - **Idea clave:** Bitcoin coordina sobre valor; Ethereum coordina sobre computación.

Capítulo 14: Cuando el Consenso se Divide - La Naturaleza del Acuerdo - El fork de Ethereum/Ethereum Classic siguió al hackeo de The DAO en 2016. - Bitcoin/Bitcoin Cash se dividió por el debate del tamaño de bloque en agosto de 2017. - **El patrón:** La tecnología permite el fork, los humanos deciden el resultado, y blockchain hace el desacuerdo auditável. - **Idea clave:** Los forks no son fracasos—son prueba de que la coordinación es voluntaria.

Capítulo 15: Capa 2 y El Trilema - Sociedades Dentro de Sociedades - El Trilema de Blockchain: Descentralización, Seguridad, Escalabilidad—elige 2 de 3. - ¿Por qué las RBDC son lentas? Porque cada nodo procesa cada transacción. - **Soluciones de Capa 2** son capas rápidas y baratas que se liquidan periódicamente con Capa 1. - Ejemplos incluyen Lightning Network para Bitcoin y Rollups para Ethereum. - **Idea clave:** El consenso puede ser anidado—sociedades dentro de sociedades.

Capítulo 16: Pruebas de Conocimiento Cero - Privacidad Se Encuentra con Verificación - El problema: Las RBDC son transparentes, lo que significa que todos ven todo. - El sueño: Privacidad Y verificabilidad—lo que parecía imposible! - **Las Pruebas de Conocimiento Cero** te permiten probar que conoces afirmaciones sobre X sin revelar X. - Tres propiedades las definen: Completitud, Solidez, y Conocimiento Cero. - **Idea clave:** Conocimiento cero es la pieza faltante—privacidad y verificación simultáneamente.

Capítulo 17: La Filosofía - Lo Que Todo Esto Significa Para La Humanidad - Esta es tecnología de coordinación para extraños a escala de internet, haciendo posibles cosas que eran técnicamente imposibles antes de 2009. - Las dinámicas de poder cambian de guardianes centralizados a coordinación distribuida. - La máquina anti-manipulación psicológica significa que no puedes ocultar la tiranía o borrar el pasado (al menos la historia económica). - Compromisos honestos incluyen responsabilidad, complejidad, irreversibilidad, uso de energía, y regulación. - **Idea clave:** La tecnología permite, los humanos deciden. Esto depende de todos nosotros.

Capítulo 18: El Control de Realidad - El Diablo Está En Los Detalles - Todo hasta ahora fue simplificado, a veces exagerado para engagement. - Bitcoin no es realmente “dinero” para la mayoría de la gente todavía, la minería está concentrada, y el uso de energía es masivo. - Los smart contracts no son imparables por defecto, las C2 introducen nuevos compromisos, y “sin necesidad

de confianza” realmente significa “confianza minimizada.” - Después de 15+ años, el uso principal sigue siendo especulación—pero la tecnología es real, y el cambio toma generaciones. - **Idea clave:** Ahora conoces ambos lados—el sueño Y la realidad. No más propaganda.

19.2 Recapitulación: Los Términos “Raros” (Ahora Deberían Sentirse Más Naturales)

Cuando empezaste, estas palabras probablemente parecían galimatías.

Ahora tienen perfecto sentido.

19.2.1 Bit

Interruptor encendido/apagado - la fundación de toda la información digital.

Entiendes: Todo lo digital—texto, imágenes, videos, dinero—son patrones de bits.

19.2.2 Logos

Los humanos asignan significado - por eso los bits pueden representar cualquier cosa.

Entiendes: La información es significado en el que acordamos. 65 = ‘A’ porque lo decimos nosotros.

19.2.3 Algoritmo

Instrucciones que los ordenadores siguen - sin magia, solo reglas.

Entiendes: Los ordenadores ejecutan algoritmos perfectamente, determinísticamente, miles de millones de veces por segundo.

19.2.4 Protocolo

Reglas de comunicación acordadas - cómo se coordinan los ordenadores.

Entiendes: Internet son protocolos. HTTP, TCP/IP, Bitcoin—todos protocolos.

19.2.5 Hash

Huella digital unidireccional - no puedes revertirla, pero prueba integridad.

Entiendes: hash(“hola”) = salida única. Cambia una letra, hash completamente diferente. No puedes ir hacia atrás.

19.2.6 Criptografía Asimétrica

Claves públicas/privadas - prueba identidad sin revelar secretos.

Entiendes: Candado (clave pública) y llave (clave privada). Cualquiera puede cerrar, solo tú puedes abrir. Las firmas digitales prueban que enviaste un mensaje sin revelar tu clave privada.

19.2. RECAPITULACIÓN: LOS TÉRMINOS “RAROS” (AHORA DEBERÍAN SENTIRSE MÁS NATURALES)

19.2.7 Consenso

Acordar sobre el estado de la base de datos - sin autoridad central.

Entiendes: El problema central que Bitcoin resolvió. ¿Cómo extraños acuerdan sobre la verdad sin confiar en nadie excepto las reglas?

19.2.8 Proof-of-Work

El esfuerzo prueba el derecho a escribir - selección resistente a Sybil.

Entiendes: Quemar electricidad para resolver rompecabezas. El ganador consigue escribir el próximo bloque. Los incentivos se alinean: honestidad = ganancia.

19.2.9 Blockchain

Historia a prueba de manipulación - la máquina anti-manipulación psicológica.

Entiendes: Lista enlazada con punteros hash. Cambia un bloque, rompe la cadena. No puedes reescribir la historia sin rehacer todo el trabajo.

19.2.10 Tecnología de Sincronización de Datos Descentralizada y RBDC

Los mejores nombres - sincronizando datos a través de extraños.

Entiendes: “Blockchain” es solo la estructura de datos. La innovación real son las redes de base de datos consensuadas que coordinan sin autoridad central.

19.2.11 Base de datos y red

Almacenar y compartir datos - el libro mayor coordinado.

Entiendes: Una base de datos es almacenamiento estructurado de datos. Una red conecta ordenadores. Una RBDC es una base de datos compartida a través de una red de ordenadores que requiere consenso para que sus datos sean alterados.

19.2.12 Smart Contracts

Programas que modifican los datos en RBDC - lógica si-entonces que puede ser imparable.

Entiendes: Código que se ejecuta automáticamente en sistemas distribuidos. No se necesita intermediario, aunque muchos contratos tienen claves de administrador. Si se cumplen condiciones, entonces ejecutar. Poderoso pero no mágico.

19.2.13 Forks

El consenso se divide - prueba de que la coordinación es voluntaria.

Entiendes: Cuando las comunidades no están de acuerdo, pueden dividirse. Ambas cadenas existen. El mercado decide el valor. El desacuerdo es auditable para siempre.

19.2.14 Capa 2

Sociedades dentro de sociedades - coordinación anidada.

Entiendes: Capas rápidas y baratas que se liquidan periódicamente con la Capa 1 lenta y segura. Como estados (C2) y gobierno federal (C1).

19.2.15 Conocimiento Cero

Probar sin revelar - privacidad y verificación simultáneamente.

Entiendes: Magia matemática. Probar que sabes algo sin revelar qué sabes. Completitud, Solidez, Conocimiento Cero.

19.3 Lo Que Ahora Entiendes

Entiendes más que solo los conceptos técnicos.

Entiendes cómo los extraños coordinan con confianza minimizada: - Las matemáticas proporcionan verdad verificable. - La criptografía permite comunicación segura e identidad. - Los incentivos alinean actores egoístas hacia beneficio colectivo. - El consenso social decide qué reglas seguir. - No se necesita autoridad central—aunque todavía confías en supuestos criptográficos, código, e infraestructura.

Entiendes por qué blockchain hace la tiranía auditabile: - Todos tienen una copia de la historia. - Los cambios son inmediatamente visibles. - No puedes reescribir el pasado sin rehacer trabajo computacional enorme. - Las minorías pueden hacer fork y preservar la verdad original. - La máquina anti-manipulación psicológica en acción.

Entiendes por qué el valor es consensual: - El oro tiene valor porque la gente está de acuerdo. - Los dólares tienen valor porque la gente está de acuerdo. (y los impuestos) - Bitcoin tiene valor porque la gente está de acuerdo. - El valor siempre es consenso social—siempre lo ha sido en sociedades grandes. - Bitcoin simplemente hace esto transparente.

Entiendes por qué esto importa para el poder: - Cambia el control de guardianes centralizados a redes distribuidas (aunque el poder no se distribuye equitativamente en la práctica). - Puedes poseer sin intermediarios (si gestionas las claves apropiadamente). - Puedes coordinarte con requisitos de confianza minimizados. - Puedes salir si no estás de acuerdo (aunque existen barreras prácticas). - El poder se redistribuye de manera diferente, no se elimina.

Entiendes por qué la tecnología no es magia: - Los ordenadores siguen reglas (algoritmos). - La criptografía son matemáticas (funciones unidireccionales, firmas, pruebas). - Los incentivos son teoría de juegos (actores egoístas, intereses alineados). - El consenso es social (los humanos deciden, el código hace cumplir). - Sin magia. Solo matemáticas, ingeniería, y coordinación social.

19.4 Ahora Puedes...

Explicar Bitcoin a tu abuela—vale, al menos a tu madre:

“Es una base de datos que miles de ordenadores sincronizan juntos.” O simplemente dales este libro.

Criticar el hype cripto inteligentemente: - ¿Este proyecto necesita una blockchain, o funcionaría una base de datos normal? - ¿Los incentivos están alineados apropiadamente? - ¿Esto está resolviendo un problema de coordinación real, o solo palabras de moda? - ¿Cuáles son los compromisos? (Responsabilidad, complejidad, energía, etc.)

Evaluar nuevos proyectos: - **Pregunta:** ¿Esto puede beneficiarse de participación sin permisos? ¿Resistencia a censura? ¿Historia transparente? ¿Sin autoridad central? - **Si sí:** Quizás una RBDC tiene sentido. - **Si no:** Solo usa una base de datos. Más simple, más rápido, más barato.

Participar informado: - Comprar, construir, regular, criticar, usar, o ignorar—tu elección. - Pero ahora puedes hacerlo con entendimiento, no hype. - Conoces los compromisos. Conoces los riesgos. Sabes qué es posible.

Pensar claramente sobre problemas de coordinación: - ¿Cuándo necesitamos confianza? ¿Cuándo podemos evitarla? - ¿Cuándo es mejor la centralización? ¿Cuándo es mejor la descentralización? - ¿Cuáles son los incentivos? ¿Quién se beneficia? ¿Quién pierde? - ¿Esta tecnología es apropiada para este problema?

19.5 La Meta-Visión

Aquí está la realización más profunda:

Aprendiste esto desde primeros principios.

No empezamos con “Bitcoin es una criptomoneda descentralizada usando consenso Proof-of-Work en un libro mayor distribuido donde los usuarios envían transacciones.” Esa frase habría sido galimatías en la página 1.

En cambio, empezamos con un interruptor de luz. Luego construimos significado. Luego algoritmos. Luego protocolos. Luego criptografía. Luego consenso. Luego Bitcoin.

Paso a paso. Capa por capa. Concepto por concepto.

Sin jerga vacía. Sin “confía en mí, es complicado.” Sin atajos.

Te GANASTE este entendimiento.

Y eso es importante, porque ahora *intuitivamente* entiendes. Ya no tanto de una manera superficial. No repitiendo ciegamente palabras de moda. Sino entendimiento desde primeros principios.

No puedes ser engañado tan fácilmente por el hype. Puedes pensar críticamente. Puedes evaluar afirmaciones.

Esta es educación verdadera.

19.6 Del Bit al Bitcoin

Cerremos el círculo.

Empezamos con un solo bit: encendido o apagado, 1 o 0—la fundación de toda la información digital.

Desde ahí, construimos: - **Significado:** Los humanos asignan logos a patrones de bits (Capítulo 2). - **Computación:** Los algoritmos manipulan bits (Capítulo 3). - **Comunicación:** Los protocolos

coordinan ordenadores (Capítulo 4). - **Seguridad:** La criptografía protege información (Capítulos 5-7). - **Coordinación:** El consenso sincroniza bases de datos (Capítulos 9-11). - **Innovación:** Smart contracts, Capa 2, Pruebas de Conocimiento Cero (Capítulos 13-16). - **Filosofía:** Lo que esto significa para la humanidad (Capítulo 17). - **Control de realidad:** El diablo está en los detalles (Capítulo 18).

Del bit al Bitcoin.

De matemáticas a significado.

De código a consenso.

De imposibilidad técnica a realidad.

Y ahora entiendes todo.

No porque memorizaste definiciones, sino porque construiste el entendimiento desde la base.

Empezaste con un interruptor de luz. Terminaste con tecnología de coordinación global.

Ese es el viaje. Ese es el logro. Cualquier cosa puede aprenderse si se descompone lo suficiente y con paciencia. Ve a aprender cualquier cosa.

Idea Clave: Has llegado tan lejos. De un solo bit a entender tecnología de coordinación global. Aprendiste desde primeros principios—sin atajos, sin jerga vacía. Ahora entiendes cómo los extraños coordinan sin confianza, por qué blockchain hace la tiranía auditável, por qué el valor es consensual, por qué esto importa para el poder, y por qué la tecnología no es magia. Puedes explicar Bitcoin claramente, criticar el hype inteligentemente, evaluar proyectos, y participar informado. Te GANASTE este entendimiento. Del bit al Bitcoin. De matemáticas a significado. De código a consenso. Ahora lo entiendes todo.

Queda un capítulo más: una carta personal de mí para ti sobre por qué esto importa, qué espero que te lleves, y qué viene después. Nos vemos ahí.

20

Capítulo 20: Epílogo

El final está aquí, solo para empezar de nuevo.

Lo lograste, felicidades. :)

Diecinueve capítulos. Del bit al Bitcoin. De interruptores de luz a tecnología de coordinación global.

Gracias por confiar en mí para guiarte a través de este viaje.

Ahora, déjame cerrar este libro con algunas palabras finales para darte una visión aún más profunda sobre cómo esta tecnología puede y no puede afectar nuestras vidas.

20.1 Por Qué Escribí Esto

Tuve una entrevista en un canal de YouTube para explicar posibles casos de uso de esta nueva tecnología. Allí, me di cuenta de que no había término medio: o te quedas demasiado vago (intentando ocultar la complejidad) y la gente piensa que eres un estafador, o usas palabras técnicas y nadie te entiende.

Debo admitir que mis habilidades de comunicación son pobres cuando hablo de cosas técnicas a audiencias no técnicas. Sin embargo, más allá de mis limitaciones, sentí que la industria no tenía una forma apropiada de explicar verdaderamente las cosas:

Los problemas de comunicación imposibles:

- “**Bitcoin es un activo digital.**” (Eso es la mitad de la historia en el mejor de los casos—recuerda, interpretamos la información)
- “**Redes blockchain.**” (¿Qué demonios es eso? Incluso los estudiantes de informática necesitan tiempo para entender una estructura de datos blockchain. ¿Cómo pueden las personas cotidianas escuchar esa palabra e intuitivamente comprenderla?)
- “**¿Dónde debería invertir?**” (Esa pregunta siempre aparece, pero de nuevo, es solo la mitad de la historia. A la gente le importa el dinero, no los casos de uso. Incluso si intentara responder, ¿cómo explico apropiadamente las diferencias de riesgo entre una criptomonedas codificada a nivel de blockchain versus a nivel de smart contract?)
- **La trampa de contradicción:** Dices cosas como “cripto no está controlado por nadie, como Ethereum,” pero luego dices “¡algunas cripto EN Ethereum están controladas?”

Suenas contradictorio. La gente piensa que eres un impostor. (Ahora entiendes cómo ambas afirmaciones son verdaderas simultáneamente.)

- **El problema de fundamentos:** La gente no sabe qué es realmente internet. Si no saben eso, ¿cómo entenderán algo tan complejo como una RBDC?
- **El problema de la doble ignorancia:** La gente—al menos en mi país, España—tiene poco o ningún conocimiento sobre finanzas, combinado con poco conocimiento sobre cómo pensar apropiadamente sobre tecnologías modernas como ordenadores. ¿El resultado? No tienen ninguna posibilidad de sentirse cómodos pensando sobre algo que combina ambas cosas. Intentas encontrar analogías con mecánicas que sí entienden, pero como has visto, la complejidad de esta tecnología significa que tu analogía siempre será incompleta. Además, porque esta tecnología habilita capacidades genuinamente nuevas, puede simplemente no haber una buena analogía para ellas. Y cuando la gente nota los huecos o contradicciones en tus explicaciones, no confiarán en ti. De nuevo, suenas como un impostor.

Recibí muchas críticas sobre esto en comentarios de YouTube, y sobre mis pobres habilidades de comunicación para temas técnicos. Eso fue principalmente porque mi cerebro enfrentaba todas estas complejidades en tiempo real.

Me di cuenta: Si realmente queremos que la gente se sienta cómoda con “cripto,” no podemos saltarnos la complejidad. Debemos desglosarla tanto como sea posible y enseñarla. Para que no sonemos como estafadores sin importar lo que digamos.

Por lo tanto, aquí estoy, escribiendo este libro.

Si tienes curiosidad, el video está en español: <https://www.youtube.com/watch?v=j2k52WvLAcg>

20.1.1 La Barrera Emocional

No puedes hacer que la gente se sienta cómoda y racional sobre algo nuevo y complejo cuando activas sus emociones hacia algo tan político y real como el dinero.

Especialmente en un país con dificultades como el mío, donde la gente lucha para llegar a fin de mes y los salarios reales de España y el PIB per cápita han estado **en gran medida estancados durante las últimas dos décadas**. El dinero no es una broma—especialmente después de tantas crisis, estafadores vendiendo cursos inútiles, todos intentando desperdiciar tu dinero en línea. Si alguien intenta explicar cosas nuevas honestamente y gratis, la gente desconfía de ti también, porque así es exactamente cómo se ven los estafadores al principio. Ganan tu confianza, luego te engañan para que pagues.

No pude encontrar otra forma de sortear esto que mejorar mis habilidades de comunicación y escribir este libro. Lo segundo, como ves, está hecho.

20.1.2 El Objetivo del Libro

El objetivo principal de este libro: **enseñar a la gente a pensar correctamente sobre estas nuevas tecnologías que moldean e influyen en nuestras vidas, del bit a Bitcoin, para que puedan saber qué son y cómo usarlas de forma segura.**

Intenté abstraer la jerga técnica tanto como fuera posible. Cuando fue necesario, proporcioné explicaciones intuitivas primero para que puedas pensar sobre cada concepto como una pieza, una herramienta, con ciertas características y usos en otros sistemas.

El objetivo más profundo: La capacidad de pensar claramente sobre temas complejos. La capacidad de evaluar afirmaciones críticamente. La capacidad de ver a través de palabras de moda y hacer las preguntas correctas. Espero que hayas obtenido una sensación de eso en este libro, y espero que puedas extraer esto a otros campos en la vida.

20.2 La Ley de Hierro de la Oligarquía - Por Qué Importa la Educación

Terminaré con una conclusión de un sociólogo alemán del siglo pasado, Robert Michels, que creo que se aplica poderosamente a esta nueva tecnología.

En su libro *“Partidos Políticos: Un Estudio Sociológico de las Tendencias Oligárquicas de la Democracia Moderna”* (publicado por primera vez en alemán en 1911, traducción al inglés en 1915), analizó cómo las organizaciones democráticas a escala—incluso aquellas con intenciones honestas y buenas—tienden a convertirse en oligarquías con el tiempo. El poder se concentra en manos de unos pocos, y los ideales originales del grupo a menudo se comprometen.

Las RBDC no son otra cosa que grandes organizaciones a escala. ¿Enfrentarán el mismo destino que Robert Michels analizó en toda coordinación humana? ¿Las nuevas tecnologías nos ayudarán a evitar las tendencias oligárquicas de las organizaciones humanas?

Probablemente no. **Pero harán todo más transparente, auditabile, y más difícil de ocultar del ojo público.** Las oligarquías no son malas por sí mismas—son malas cuando contradicen y van contra la gente a la que se supone que representan.

La transparencia, la velocidad de disponibilidad de información, y la auditabilidad son herramientas poderosas contra estos casos. Crean un público más poderoso, más informado que puede alinear rápidamente a sus oligarcas inevitables con sus objetivos originales.

En las últimas páginas de la segunda edición de su libro, escribió:

Una educación más amplia da a la gente una mayor capacidad para ejercer supervisión y control. Vemos cada día que entre los ricos, por muy grandes que sean, la autoridad de los líderes sobre sus seguidores nunca es tan irrestricta como lo es entre los pobres. En la masa, los pobres son impotentes e indefensos ante sus líderes. Su inferioridad intelectual y cultural hace imposible que entiendan a dónde los lleva el líder o que prevean la importancia de sus acciones. Por lo tanto, la gran tarea de la educación social es elevar el nivel intelectual de las masas para que, en la medida de lo posible, puedan contrarrestar las tendencias oligárquicas del movimiento de la clase trabajadora.

Él principalmente analizó el movimiento socialista de la clase trabajadora de su era, pero dejó claro que estos patrones permanecen verdaderos a través de contextos. El rasgo general: las organizaciones humanas democráticas a escala tienden a formar oligarquías.

20.2.1 La Oligarquía de Frikis

Nosotros, los frikis—los que construimos el código, los que pueden leer el código, los que pueden escribir el código—nos convertiremos en una oligarquía fuerte si esta tecnología es masivamente adoptada. Y porque necesitamos muchos frikis para construir una base de código tan masiva, tendremos una oligarquía dentro de la oligarquía.

Eso es inevitable. Si quieres saber más sobre por qué, lee el libro de Robert. Fue escrito **hace más de un siglo** pero se siente como si hubiera sido escrito hoy.

Debemos educar lo que Robert llama “las masas”—la gente cotidiana—para que puedan cuestionarnos y hacer el sistema aún más resiliente a la corrupción y el abuso de poder. Por diseño, las RBDC son muy resistentes. Pero el diseño solo no es suficiente.

¿Recuerdas el hackeo de The DAO de Ethereum que llevó a un fork? Algunos críticos de la opción “revertir el hackeo” dijeron: *“Esto solo está sucediendo porque una gran figura en el grupo, el creador Vitalik Buterin, se está posicionando hacia esa opción. ¿Qué tipo de consenso es este? ¿El consenso de un hombre?”*

Fueron agudos en esta observación. Esta es solo una de muchas tendencias oligárquicas que los humanos tienen por naturaleza. Usualmente las masas (todos los ejecutores de nodos) no tienen tanto conocimiento técnico como los líderes, y de alguna manera “ciegamente” confían en sus opiniones y los siguen.

Sí, la comunidad decidió. Pero ¿hasta qué punto fue esta decisión influenciada por tendencia oligárquica? ¿Es esto realmente consenso?

Aquí está el riesgo real más profundo y complejo de la industria ahora mismo: Necesitamos más frikis y aún más importante, necesitamos que la gente entienda mejor lo que hacemos.

De lo contrario nos convertiremos, como Robert Michels estudió, en una oligarquía de frikis con el potencial de ir contra los intereses de las masas.

Una pequeña chispa de este fenómeno ya parecía haber sucedido.

No solo necesitamos que aprendas para que la tecnología funcione por razones técnicas, sino también debido a tendencias sociales humanas inevitables.

Esta es también la razón por la que escribí este libro.

20.3 Permiso del Autor - Este Libro Es Gratis Para Siempre

Puedes copiarlo. Compartirlo. Remezclarlo. Mejorarlo. Traducirlo. Hacer fork de él.

Al igual que Bitcoin mismo, este conocimiento pertenece a todos.

Si encontraste valor en este libro, lo mejor que puedes hacer es compartirlo con la gente.

El conocimiento debería ser gratis. El entendimiento debería ser accesible.

Por eso escribí esto. Por eso es gratis para siempre.

No prosperamos cuando los recursos son escasos, y el conocimiento es un recurso.

20.4 Por Qué Esto Importa Más Allá de la Tecnología

Las sociedades que confían entre sí generan más riqueza. El conocimiento es un recurso. No prosperamos cuando los recursos son escasos.

Aquellos que entienden y exploran cuidadosamente esta nueva tecnología tendrán ventajas sobre aquellos que la descartan completamente. Sin embargo, tan revolucionaria como es, adoptarla completamente de la noche a la mañana sería imprudente e imposible.

Recuerda: Cuando se inventaron los coches, no tuvieron mágicamente carreteras y reglas de tráfico en 10 años. Tomó generaciones construir la infraestructura, establecer estándares de seguridad, e integrarlos en la vida diaria para que sus beneficios reales se materializaran.

Lo mismo pasa aquí.

Esta tecnología permite a personas que no confían entre sí—incluso personas que se han traicionado mutuamente—encontrar nuevas formas de coordinar. Decir: “Intentémoslo de nuevo, pero esta vez usaremos tecnologías de sincronización de datos descentralizadas como RBDC para minimizar los requisitos de confianza.”

Esa es la promesa. No perfección. No utopía. Solo nuevas herramientas para un viejo problema: cómo coordinar cuando la confianza es escasa.

20.5 Del Bit al Bitcoin

Y así regresamos a donde comenzamos—pero no eres la misma persona que abrió este libro.

Comenzaste con un interruptor de luz, preguntándote qué significaba realmente un “bit”. Ahora entiendes cómo esa simple distinción encendido-o-apagado escala en tecnología de coordinación global. Has visto cómo la criptografía crea confianza entre extraños, cómo el consenso emerge del caos, y cómo el código puede hacer cumplir acuerdos que las palabras solas no pueden.

Más importante, has aprendido a pensar claramente sobre sistemas complejos. No aceptar palabras de moda al pie de la letra. No descartar lo que no entiendes. Sino preguntar: *¿Qué problema resuelve esto realmente? ¿Cuáles son los compromisos? ¿Quién controla qué? ¿Cómo puedo entender esto mejor descomponiéndolo en conceptos más pequeños?*

Esto importa más allá de cripto. Las mismas herramientas mentales—descomponer la complejidad, cuestionar supuestos, ver a través del hype—aplican en todas partes: a la IA, a los algoritmos de redes sociales, a cualquier tecnología que moldea tu vida mientras oculta su mecánica.

La oligarquía de frikis es real, y crecerá. Pero ahora estás mejor equipado para cuestionarnos. Para hacernos responsables. Para participar en decisiones sobre tecnologías que te afectan. Eso no es nada—eso es exactamente lo que Robert Michels dijo que las masas necesitan.

Este conocimiento es tuyo ahora. Compártelo. Úsallo. Construye con él.

De ti hacia el futuro.

Gracias por ser parte de este viaje.

Atentamente, El Autor, Carlos D. Alegre