Universidad de Córdoba

Escuela Politécnica Superior de Córdoba

máster universitario en inteligencia computacional e internet de las cosas

Cyber Seguridad (CS)

Blockchain mediante Python.

Estudiante:

Mabrouka Salmi
z12salsm@uco.es

17 de febrero de 24

Índice

Índice de figuras

Listings

1 INTRODUCCIÓN

La tecnología de blockchain se erige como una innovación revolucionaria, remodelando el panorama de la gestión de datos y las transacciones descentralizadas. En su núcleo, blockchain es un libro de contabilidad distribuido e inmutable que garantiza transparencia, seguridad y confianza en las interacciones digitales. Esta introducción ofrece una visión concisa del poder transformador que blockchain posee en diversas industrias, prometiendo eficiencia, confianza y nuevos paradigmas de colaboración.

La importancia de la cadena de bloques va más allá de su aplicación inicial en la criptomoneda. Introduce una forma revolucionaria de registrar y verificar transacciones, eliminando la necesidad de intermediarios y proporcionando un marco descentralizado resistente a manipulaciones. A medida que profundizamos en las complejidades de la cadena de bloques, la comprensión teórica y la implementación práctica se convierten en componentes simbióticos de nuestra exploración. El vínculo entre teoría y práctica no es solo conceptual; es el puente que nos permite traducir principios abstractos en aplicaciones tangibles. Este enlace asegura que nuestra exploración no se limite a ámbitos teóricos, sino que se extienda al desarrollo práctico de una cadena de bloques utilizando Python. A través de este proyecto, la intersección entre teoría y práctica promete una exploración integral y enriquecedora de la tecnología de cadena de bloques.

2 COMPRENSIÓN TEÓRICA DE BLOCKCHAIN

2.1 Introducción a la Tecnología Blockchain

La tecnología blockchain ha evolucionado significativamente a lo largo de los años, marcando un viaje transformador desde sus orígenes conceptuales hasta implementaciones prácticas. El contexto histórico nos ofrece una perspectiva a través de la cual podemos rastrear el desarrollo de este sistema de registro descentralizado Gorkhali et al. (2020); Di Pierro (2017). Hitos y avances destacados han dado forma a la trayectoria de la blockchain, convirtiéndola en una fuerza revolucionaria en la era digital.

Más allá de sus raíces históricas, la cadena de bloques se ha convertido en un eje fundamental en diversas industrias, ofreciendo soluciones a desafíos de larga data. Su importancia abarca finanzas, cadena de suministro, atención médica y más. Al explorar casos de uso específicos, obtenemos perspectivas sobre cómo la cadena de bloques ha revolucionado procesos, garantizando transparencia, seguridad y eficiencia.

Uno de los aspectos fundamentales de la cadena de bloques es su papel en garantizar la integridad de los datos. La inmutabilidad de los datos registrados en la cadena de bloques proporciona un medio seguro y confiable para almacenar y verificar información. Esta característica aborda un desafío crítico en la gestión de datos, contribuyendo a la solidez de las aplicaciones de la cadena de bloques.

La descentralización se erige como una característica emblemática de la cadena de bloques, eliminando la necesidad de una autoridad central. Esto no solo fomenta la confianza entre los participantes, sino que también reduce el riesgo de puntos únicos de falla

o manipulación. El potencial disruptivo de la cadena de bloques al desafiar los modelos de negocios tradicionales destaca aún más su importancia en un paisaje tecnológico que cambia rápidamente.

El impacto global de la cadena de bloques se extiende más allá de las industrias, prometiendo inclusión financiera, reducción de fraudes y empoderamiento en regiones con acceso limitado a sistemas financieros tradicionales. Consideraciones éticas, como las preocupaciones sobre la privacidad, subrayan la necesidad de equilibrar la transparencia con los derechos individuales en el ecosistema de la cadena de bloques.

Reconocemos a través del entendimiento teórico de la cadena de bloques, su conexión directa con los principios de ciberseguridad. La seguridad criptográfica, la autenticación descentralizada y la resistencia contra la manipulación o ataques posicionan a la cadena de bloques como un componente crucial en el panorama más amplio de la ciberseguridad.

2.2 Conceptos fundamentales

2.2.1 Bloques, Transacciones y Estructura de la Cadena de Bloques

Los elementos fundamentales de la tecnología de blockchain consisten en bloques, transacciones y la estructura subyacente de la cadena de bloques, los cuales establecen de manera colaborativa un sistema de registro seguro y transparente.

Bloques: Un bloque es una unidad fundamental en la cadena de bloques, que contiene un conjunto de transacciones. Cada bloque se identifica de manera única mediante un índice y está vinculado al bloque anterior a través de un hash criptográfico. Esta vinculación crea una cadena cronológica, asegurando la integridad de toda la historia de transacciones.

Transacciones: Las transacciones representan las entradas de datos almacenadas dentro de un bloque. Estas pueden abarcar una amplia variedad de información, dependiendo del caso de uso específico de la cadena de bloques. La transparencia e inmutabilidad de las transacciones dentro de los bloques contribuyen a la confiabilidad general del sistema.

Estructura de la Cadena de Bloques: La propia cadena de bloques es un libro de contabilidad distribuido y descentralizado que registra todas las transacciones a lo largo de su red. La estructura asegura que cada participante tenga una copia de toda la historia de transacciones, promoviendo la transparencia y evitando un único punto de fallo. Comprender cómo se forman los bloques y las transacciones en la estructura de la cadena de bloques es fundamental para comprender los principios fundamentales de la tecnología Gorkhali et al. (2020).

2.2.2 Descentralización, Transparencia y Mecanismos de Consenso

Descentralización: La descentralización es una filosofía clave en el diseño de blockchain. A diferencia de los sistemas centralizados tradicionales, blockchain opera en una red de pares donde cada participante (nodo) tiene autoridad igual. Esto no solo elimina la necesidad de una autoridad central, sino que también mejora la seguridad al distribuir el control a lo largo de la red Gorkhali et al. (2020); Di Pierro (2017).

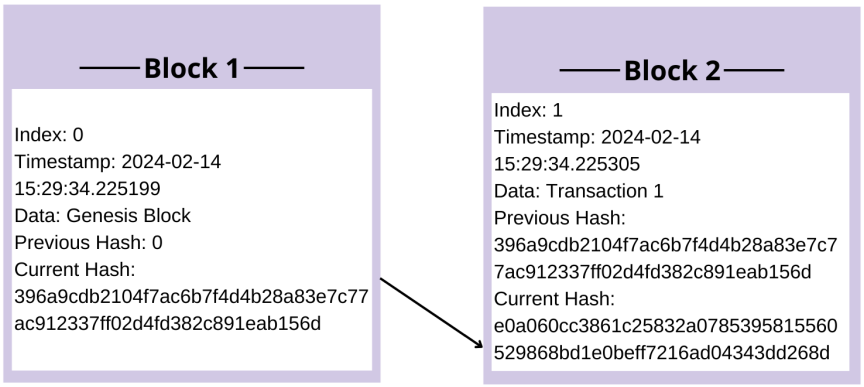

Figura 1: Integridad de la Cadena

Transparencia: La transparencia de la cadena de bloques se logra a través de la naturaleza abierta y pública de su libro de contabilidad. Todos los participantes pueden ver el historial completo de transacciones, garantizando la responsabilidad y la confianza. La transparencia es un aspecto crucial del atractivo de la cadena de bloques, especialmente en industrias donde la responsabilidad es primordial.
Mecanismos de Consenso: Los mecanismos de consenso desempeñan un papel fundamental en mantener un acuerdo entre los participantes sobre el estado de la cadena de bloques. Los mecanismos comunes incluyen Prueba de Trabajo (PoW) y Prueba de Participación (PoS), cada uno con su enfoque para validar transacciones y crear nuevos bloques. Los mecanismos de consenso son fundamentales para la seguridad e integridad de la red de la cadena de bloques.

2.3 Fundamentos Criptográficos

Dentro de la compleja estructura de la cadena de bloques, el papel esencial de los fundamentos criptográficos es crucial para garantizar la seguridad y la integridad del sistema en su conjunto. Esta sección explora los aspectos fundamentales que constituyen la base de los elementos criptográficos dentro de la tecnología de la cadena de bloques Gorkhali et al. (2020); Di Pierro (2017).

2.3.1 Papel de la Criptografía:

La criptografía constituye la base fundamental para garantizar la seguridad de la cadena de bloques. Cumple una función diversa al asegurar la privacidad, consistencia y legitimidad de los datos. A través de la aplicación de métodos criptográficos, la cadena de bloques alcanza un nivel crucial de confianza y seguridad esencial para su correcto funcionamiento.

2.3.2 Funciones Criptográficas Clave:

Dos funciones criptográficas esenciales, las funciones hash y la criptografía de clave pública, conforman la columna vertebral de la seguridad en blockchain.
Funciones de hash: Las funciones hash son algoritmos criptográficos que transforman datos de entrada en una cadena de caracteres de tamaño fijo. En el contexto de la cadena de bloques, las funciones hash desempeñan un papel crucial en la creación de un identificador único (hash) para cada bloque. Este identificador se determina según el contenido del bloque, lo que hace prácticamente imposible alterar los datos dentro de un bloque sin cambiar el hash. Como resultado, las funciones hash contribuyen de manera significativa a la inmutabilidad e integridad de la cadena de bloques.
Criptografía de clave pública: Criptografía de clave pública, también conocida como criptografía asimétrica, implica un par de claves criptográficas: una clave pública y una clave privada. La clave pública se comparte abiertamente, mientras que la clave privada se mantiene confidencial. En el blockchain, esta técnica criptográfica es fundamental para asegurar transacciones. Cada participante posee un par de claves, lo que les permite recibir información cifrada utilizando su clave pública y descifrarla utilizando su clave privada. Esto garantiza transacciones seguras y verificables, contribuyendo a la confiabilidad general de la red blockchain.

2.4 Casos de Uso y Aplicaciones

La versatilidad de la cadena de bloques va mucho más allá de sus fundamentos teóricos, encontrando aplicaciones prácticas en industrias que requieren transparencia, seguridad y eficiencia. Aquí hay algunos ejemplos ilustrativos Gorkhali et al. (2020); Zheng et al. (2018):

Finanzas: La cadena de bloques revoluciona las transacciones financieras al proporcionar un libro de contabilidad transparente e resistente a manipulaciones. Las criptomonedas como Bitcoin utilizan la cadena de bloques para realizar transacciones descentralizadas de igual a igual, eliminando la necesidad de intermediarios.

Cadena de Suministro: El seguimiento y la verificación de la procedencia de los productos se vuelven más eficientes con la tecnología blockchain. Garantiza la transparencia en la cadena de suministro, reduciendo el riesgo de productos falsificados y mejorando la trazabilidad.

Atención médica: La cadena de bloques asegura los datos de los pacientes y optimiza los procesos de atención médica. Permite la interoperabilidad entre sistemas dispares,

al mismo tiempo que garantiza la confidencialidad e integridad de los registros médicos sensibles.

Contratos Inteligentes: Contratos inteligentes, contratos autoejecutables con los términos del acuerdo escritos directamente en código, aprovechan la cadena de bloques para la transparencia y la automatización. Estos contratos encuentran aplicaciones en procesos legales, seguros y varios otros ámbitos.

Gestión de Identidad: La cadena de bloques proporciona una solución segura y descentralizada para la gestión de identidades. Los usuarios tienen control sobre su información personal, reduciendo el riesgo de robo de identidad.

Bienes raíces: Las transacciones de propiedad se benefician de la transparencia y seguridad de la cadena de bloques. Simplifica la transferencia de la propiedad y garantiza un registro inmutable de la historia de la propiedad.

2.5 Desafíos y Tendencias Futuras

Reconocer tanto las dificultades actuales como las prometedoras tendencias futuras es crucial para comprender la dirección de este entorno descentralizado Di Pierro (2017); Ghosh et al. (2020); Zheng et al. (2018).

2.5.1 Desafíos actuales y tendencias emergentes

Escalabilidad: La tecnología blockchain enfrenta desafíos en la escalabilidad para dar cabida a un creciente número de transacciones. A medida que la demanda aumenta, las redes existentes pueden experimentar congestión, afectando la velocidad y eficiencia de las transacciones. Están surgiendo soluciones como el "shardingz la escalabilidad de çapa dos"para abordar las preocupaciones de escalabilidad.
Interoperabilidad: La interoperabilidad entre diferentes redes de blockchain sigue siendo un desafío. Establecer una comunicación y transferencia de datos fluida entre sistemas de blockchain dispares es crucial para la adopción generalizada e integración de la tecnología blockchain.
Cumplimiento normativo: Navegar por los marcos regulatorios presenta un desafío para los proyectos de blockchain. Encontrar un equilibrio entre la innovación y el cumplimiento es esencial para el crecimiento sostenido de las tecnologías blockchain.
Consumo de energía: Mecanismos de consenso "Proof of Work"(PoW), empleados por algunas blockchains como Bitcoin, consumen cantidades significativas de energía. Explorar mecanismos de consenso ecológicos, como "Proof of Stake"(PoS), es una solución en tendencia para abordar las preocupaciones ambientales.

2.5.2 Las tendencias emergentes incluyen

DeFi (Finanzas Descentralizadas): El auge de las aplicaciones de finanzas descentralizadas aprovecha la tecnología de blockchain para recrear sistemas financieros tradicionales sin intermediarios. Las finanzas descentralizadas introducen nuevas posibilidades en préstamos, endeudamiento y comercio, desafiando a las instituciones financieras convencionales.

NFTs (Non-Fungible Tokens): "Tokens No Fungibles", que representan activos digitales únicos en la cadena de bloques, han ganado una inmensa popularidad. Desde arte digital hasta bienes raíces virtuales, los NFTs muestran el potencial de la cadena de bloques en establecer la propiedad y procedencia en el ámbito digital.

2.6 Conclusión de la Parte Teórica

En resumen, nuestro examen de la teoría de blockchain ha abarcado sus orígenes históricos, usos prácticos en diversas industrias, fundamentos criptográficos y los desafíos y desarrollos actuales que influyen en su camino. Hemos rastreado cómo blockchain ha evolucionado desde sus inicios conceptuales hasta su presencia impactante en diferentes sectores. Al explorar los principios criptográficos subyacentes, reconocemos la importancia de las funciones hash y la criptografía de clave pública para garantizar la seguridad de la tecnología blockchain.

Examining practical uses in various industries has showcased the adaptability and revolutionary capabilities of blockchain technology. Sectors like finance, supply chain, and healthcare have particularly benefited. Throughout this exploration, challenges such as scalability were confronted, and observations of emerging trends provided valuable insights into the ever-evolving nature of blockchain. These insights now serve as the foundation for our hands-on experience, guiding us as we embark on the implementation of a blockchain system using Python.

3 IMPLEMENTACIÓN PRÁCTICA DE UN BLOCKCHAIN SIMPLE EN PYTHON

3.1 Introducción a la Implementación Práctica

Los fundamentos teóricos establecidos en las secciones anteriores, que abarcan la evolución histórica, los principios criptográficos y las aplicaciones reales de la cadena de bloques, sientan las bases para una exploración práctica. Esta comprensión teórica permite la traducción de conceptos a código tangible, demostrado a través de una implementación en Python.

La implementación sirve como una demostración de cómo la teoría de la cadena de bloques se transforma en un sistema funcional, cerrando la brecha entre la teoría y la práctica. Las secciones próximas detallarán el proceso paso a paso para construir una cadena de bloques simple en Python, facilitando una integración fluida del conocimiento teórico y su aplicación práctica.

3.2 Script de Python para la Creación de Bloques

Un bloque típicamente consta de varios atributos clave como índice, marca de tiempo, datos, hash anterior y hash actual Bispo (2022). Estos atributos sirven como los bloques de construcción de la estructura de la cadena de bloques 1.

Listing 1: Código en Python para Generar Bloques

```python
class Block:
    def __init__(self, index, timestamp, data, previous_hash):
        """
        Initializes a new block.

        Parameters:
        - index: The index of the block in the blockchain.
        - timestamp: The timestamp of block creation.
        - data: The data or transaction information stored in the block.
        - previous_hash: The hash of the previous block in the blockchain.
        """
        self.index = index
        self.timestamp = timestamp
        self.data = data
        self.previous_hash = previous_hash
        self.current_hash = self.calculate_hash()

    def calculate_hash(self):
        sha = hashlib.sha256()
        sha.update(str(self.index).encode('utf-8') +
                   str(self.timestamp).encode('utf-8') +
                   str(self.data).encode('utf-8') +
                   str(self.previous_hash).encode('utf-8'))
        return sha.hexdigest()
```

En este fragmento de código, establecemos las bases para nuestra clase de bloque. El método *_init_* inicializa el bloque con los atributos proporcionados, y el método *calculate_hash* define la lógica para generar el hash basado en esos atributos. Esta pieza de código concisa pero fundamental sienta las bases para el desarrollo posterior de nuestra cadena de bloques Martinez (2020).

3.3 Estructura básica de Blockchain en Python

Creamos una instancia de una clase Blockchain, que constituye el núcleo de nuestra implementación práctica. Esta clase abarca el conjunto de bloques y coordina sus interacciones. Mediante código en Python, especificamos la arquitectura de la cadena de bloques 2:

Listing 2: Implementación en Python de un código simple de una clase Blockchain

```python
class Blockchain:
    def __init__(self):
        self.chain = [self.create_genesis_block()]

    def create_genesis_block(self):
        return Block(0, datetime.datetime.now(), "Genesis Block",
        "0")
```

```
8
9       def add_block(self, data):
10          previous_block = self.chain[-1]
11          new_index = previous_block.index + 1
12          new_timestamp = datetime.datetime.now()
13          new_block = Block(new_index, new_timestamp, data,
    previous_block.current_hash)
14          self.chain.append(new_block)
15
16      def display_blockchain(self):
17          for block in self.chain:
18              print(f"Index: {block.index}")
19              print(f"Timestamp: {block.timestamp}")
20              print(f"Data: {block.data}")
21              print(f"Previous Hash: {block.previous_hash}")
22              print(f"Current Hash: {block.current_hash}")
23              print("\n")
```

La clase Blockchain comienza con un método de inicialización "(*__init__*)"que configura el estado inicial, incluida la creación del bloque génesis. Se definen métodos adicionales "(*create_genesis_block*, *add_block* y *get_latest_block*)"para facilitar la adición de nuevos bloques y la obtención del bloque más reciente.

3.3.1 Métodos para Bloque Génesis, Agregar Bloques y Obtener el Último Bloque.

- El método *create_genesis_block* inicializa la cadena de bloques con el primer bloque, comúnmente conocido como el bloque génesis. Este bloque sirve como la piedra angular fundamental desde la cual se extenderán los bloques subsiguientes.
- La función *add_block* simplifica el proceso de agregar nuevos bloques a la cadena de bloques. Coordina la generación de un nuevo bloque utilizando los datos proporcionados y, posteriormente, lo añade a la cadena actual.
- Finalmente, el método *get_latest_block* permite la obtención del bloque más reciente en la cadena de bloques. Esta funcionalidad es esencial para mantener la integridad y continuidad de la cadena de bloques.

3.4 Funciones de hash en Python

La biblioteca hashlib de Python Smith (2010) proporciona una base sólida para implementar funciones de hash. Al aplicar algoritmos de hash como SHA-256, podemos generar hashes seguros y únicos para nuestros bloques. El siguiente fragmento de código ilustra la integración de hashlib para el cálculo del hash:

```
1 import hashlib
2
3 class Block:
```

```
    # ... (previous Block class code)

    def calculate_hash(self):
        hash_string = str(self.index) + str(self.timestamp) + str(self.data) + str(self.previous_hash)
        calculated_hash = hashlib.sha256(hash_string.encode()).hexdigest()
        return calculated_hash
```

En este código, el método *calculate_hash* dentro de la clase Block incorpora *hashlib* para crear un hash basado en los atributos del bloque. Esta integración fortalece la seguridad y la integridad de nuestra cadena de bloques, asegurando que el hash de cada bloque sea único y resistente a manipulaciones.

3.5 Mostrando la Blockchain en Python

Una de las fortalezas de la cadena de bloques radica en su transparencia y la capacidad de que todos los participantes vean el historial completo de transacciones. Para llevar esta transparencia a nuestra implementación en Python, creamos una función 3 que muestra toda la cadena de bloques. Esta función itera a través de cada bloque, presentando su índice, marca de tiempo, datos y hash. El resultado es una representación clara y comprensible de la estructura de la cadena de bloques.

Listing 3: Función de Python para mostrar la cadena de bloques

```
def display_blockchain(blockchain):
    for block in blockchain:
        print(f"Index: {block.index}")
        print(f"Timestamp: {block.timestamp}")
        print(f"Data: {block.data}")
        print(f"Previous Hash: {block.previous_hash}")
        print(f"Current Hash: {block.current_hash}")
        print("\n")
```

En este fragmento de código, la función *display_blockchain* toma la cadena de bloques completa como entrada e imprime los detalles de cada bloque en un formato legible.

3.6 Añadiendo Bloques en Python

Este proceso simula transacciones del mundo real o adiciones de datos, mostrando cómo cada bloque está vinculado de manera segura al anterior. Iniciamos el proceso 4 creando un nuevo bloque y agregándolo a la cadena de bloques existente.

Listing 4: Función para agregar bloques en Python

```
# Assuming 'blockchain' is an existing blockchain instance
new_data = "New transaction or data"
blockchain.add_block(new_data)
```

En este fragmento de código, asumimos que 'blockchain' es una instancia de nuestra clase *blockchain*. El método *add_block* se llama con los nuevos datos como parámetro, y el método se encarga de la creación de un nuevo bloque, asegurando su enlace seguro con la cadena de bloques existente.

Esta implementación práctica refuerza el concepto de un libro de contabilidad descentralizado, donde los participantes en toda la red pueden contribuir de manera independiente al crecimiento continuo de la cadena de bloques.

3.7 Pruebas y Verificación en Python

Para garantizar la solidez de nuestra implementación de blockchain, desarrollamos un script que realiza diversas pruebas, como agregar múltiples bloques, verificar la consistencia de los hashes y comprobar el orden secuencial de los bloques. Este script 5 funciona como una herramienta integral para probar la blockchain en diferentes escenarios.

Listing 5: Verificar la integridad de toda la cadena de bloques.

```
# Assuming 'blockchain' is an existing blockchain instance
# Adding multiple blocks for testing
blockchain.add_block("Test Data 1")
blockchain.add_block("Test Data 2")

# Verifying the integrity of the entire blockchain
is_valid = blockchain.verify_integrity()

if is_valid:
    print("Blockchain is valid and secure.")
else:
    print("Blockchain integrity compromised.")
```

En este fragmento de código, asumimos que "blockchain.es una instancia de nuestra clase *blockchain*. El script añade varios bloques con datos de prueba y luego llama al método *verify_integrity* para comprobar la consistencia general y la seguridad de la cadena de bloques. Este script proporciona un enfoque dinámico para asegurar que nuestra cadena de bloques funcione según lo esperado bajo diversas condiciones.

Ejecutar un script como este se convierte en un paso esencial en el proceso de desarrollo, infundiendo confianza en la fiabilidad y seguridad de nuestra implementación de blockchain. También se alinea con los principios fundamentales de la blockchain, donde la integridad de toda la cadena es crucial para fomentar la confianza entre los participantes en la red descentralizada.

Al probar y verificar la cadena de bloques en Python, la experiencia práctica refuerza la comprensión teórica adquirida anteriormente, creando una exploración integral y completa de la tecnología de cadena de bloques.

3.8 Conclusión de la Implementación Práctica

El código en Python que creamos encapsula la esencia de la cadena de bloques, ilustrando la naturaleza secuencial y transparente de las transacciones, al tiempo que garantiza la integridad de los datos mediante el uso de funciones criptográficas de hash. El código completo se puede encontrar en este **cuaderno Colab**.

A lo largo de la implementación, priorizamos la documentación del código y los comentarios para lograr claridad y comprensión, sentando las bases para un código base mantenible y comprensible. El script de prueba sirvió como una herramienta crítica para verificar la integridad de la cadena de bloques, confirmando la resistencia a manipulaciones y la conexión segura de los bloques. Este viaje práctico no solo ha desmitificado los mecanismos de la cadena de bloques, sino que también nos ha equipado con habilidades prácticas de codificación, permitiéndonos cerrar la brecha entre el conocimiento teórico y la tecnología aplicada.

4 CONCLUSIÓN GENERAL

Concluyendo nuestro proyecto "Blockchain utilizando Python", reflexionamos sobre un viaje que exploró tanto la teoría como la implementación práctica de esta tecnología transformadora, con un notable énfasis en su papel crucial en el fortalecimiento de la ciberseguridad. La exploración comenzó sumergiéndonos en los fundamentos teóricos, abordando la evolución histórica, las complejidades criptográficas, las aplicaciones del mundo real, los desafíos y las tendencias emergentes de la cadena de bloques. Esto sentó una base sólida para comprender su naturaleza descentralizada y transparente.

Pasando de la teoría a la práctica, implementamos una cadena de bloques simple en Python. Esta experiencia práctica incluyó la creación de bloques, la construcción de una cadena de bloques funcional y la adición de nuevos bloques asegurando la integridad de los datos mediante el uso de funciones hash. Cada línea de código dio vida a conceptos teóricos, resaltando principios como la descentralización, la transparencia y la seguridad criptográfica.

Integral a nuestra discusión es la intersección de la cadena de bloques con la ciberseguridad. El proyecto destaca cómo las características inherentes de la cadena de bloques, como los mecanismos de consenso descentralizado, el hashing criptográfico y la inmutabilidad, contribuyen significativamente a mejorar los protocolos de ciberseguridad. La criptografía de clave pública asegura transacciones seguras e inalterables, convirtiendo a la cadena de bloques en un activo valioso para la seguridad de datos sensibles en diversas industrias.

Referencias

Bispo, N. (2022). Python blockchain. Accessed on February 13, 2024.

Di Pierro, M. (2017). What is the blockchain? *Computing in Science & Engineering*, 19(5):92–95.

Ghosh, A., Gupta, S., Dua, A., and Kumar, N. (2020). Security of cryptocurrencies in blockchain technology: State-of-art, challenges and future prospects. *Journal of Network and Computer Applications*, 163:102635.

Gorkhali, A., Li, L., and Shrestha, A. (2020). Blockchain: A literature review. *Journal of Management Analytics*, 7(3):321–343.

Martinez, J. C. (2020). From zero to blockchain in python — part 1. Accessed on February 14, 2024.

Smith, G. P. (2010). hashlib — secure hashes and message digests. Accessed on February 14, 2024.

Zheng, Z., Xie, S., Dai, H.-N., Chen, X., and Wang, H. (2018). Blockchain challenges and opportunities: A survey. *International journal of web and grid services*, 14(4):352–375.