Instituto Tecnológico y de Estudios Superiores de Occidente

Reconocimiento de validez oficial de estudios de nivel superior según acuerdo secretarial 15018, publicado en el Diario Oficial de la Federación del 29 de noviembre de 1976.

Departamento de Electrónica, Sistemas e Informática

Maestría en Sistemas Computacionales

GOVERNMENT FILE MANAGEMENT SYSTEM MAKING USE OF A LOCAL ETHEREUM BLOCKCHAIN

TRABAJO RECEPCIONAL que para obtener el **GRADO** de
MAESTRO EN SISTEMAS COMPUTACIONALES

Presenta: **EMANUEL ROBLEDO MORÁN**
Director **DOCTOR LUIS JULIÁN DOMÍNGUEZ PÉREZ**

Tlaquepaque, Jalisco. Julio 2024

Table of Contents

1. Introduction

1.1 Background

Blockchain technology, even though it came out at the end of the first decade of this century, has taken a lot of relevance in recent years, transcending its own niche and even being a casual conversation topic among people.

The idea of a blockchain started in the 90s, with the objective of storing files without manipulation risk, in a digital and practical way.

Years later, at the half of the decade of 2000, there was a system called reusable proof of work, in which, you could get a proof of work token based on Hashcash that was RSA signed and could transfer from one person to another.

Based on the Hashcash proof of work algorithm but using a peer-to-peer decentralized protocol instead of RPoW, Bitcoin received its protection against double expenses. In summary, the individual miners “mine” Bitcoin to get a reward by using the proof of work mechanism, and then the other decentralized nodes in the network verify it.

Mining consists of solving certain mathematical problems while the network nodes are collecting and grouping the transactions. Then, the person who can first solve the problem will be the one who validates the new block in the blockchain.

There are multiple schemes for obtaining cryptocurrency, the following ones are the most common:

Proof of work: Using the trial and failure method, it looks for a “digest message” that contains certain characteristics (generally a certain amount of 0’s at the left) until it gets the desired result, which is rewarded with a virtual coin.

Proof of stake: In this scheme, those who possess a larger number of coins are given a higher priority, because it is considered logical that, since they want to increase the value of their coins, it is more reliable when they verify the blockchain.

With the evolution and development of this technology, there have emerged some problems, like hoarding in cryptocurrency mining, because when you are looking to get a “digest message” using hash functions, it is often given more priority to the ones who have a bigger amount of coins, or the ones with more powerful equipment (because they have more possibilities to get a valid digest before)

To solve the above-mentioned mathematical problems, it is necessary a considerable computing power, used to be provided by GPU mining, this caused a wave of hoarding of these hardware devices, making them more expensive for the public in general, despite not being the main objective of this piece of hardware. But nowadays this method is obsolete, and with specialized hardware, it is possible to get better and faster results.

Even though mining seems ideal, there are potential sources of fraud, like double spending, which consists of spending the same coin twice, and since you add another coin that did not exist before, it devalues the cost of the rest of the coins. In the case of Bitcoin, this is avoided by adding each transaction to the blockchain and verifying it subsequently.

In 2013, Vitalik Buterin, software developer and co-founder of the Bitcoin magazine, declared that Bitcoin needed a scripting language to create decentralized applications. Since the community could not reach an agreement, Vitalik started developing a new distributed computation platform based on blockchain, Ethereum, which introduced a scripting functionality, called smart contracts. Smart contracts are programs or scripts that are implemented and executed in the Ethereum blockchain. They can be used, for instance, to make a transaction if certain conditions are met. Smart contracts are written in specific programming languages and are compiled in a byte's code that a completely decentralized Turing virtual machine, called an Ethereum virtual machine (EVM), can read and execute.

The Ethereum cryptocurrency is called Ether, it can be transferred between accounts and is used to pay the calculation power commission used when executing smart contracts.

For this project, we aim to use an Ethereum local blockchain, consisting of multiple nodes as well as several users.

1.2 Justification

The current file management system such as the File National Register, or RNA (Registro Nacional de Archivos) in Mexico is not optimal, is a very bureaucratic process, as well as a very vulnerable one, because there is not a very solid way to verify the integrity of the files. That is why it is going to be used as a blockchain, since with these kinds of functions, it is necessary to corroborate the multiple transactions made throughout time, but it must be made efficiently and keep its integrity.

This, when it transfers to official files and its registration, assures the integrity of all the files that have been registered previously, adding the most recent files to the whole chain, making it impossible to manipulate old files, because this would mean a very radical change in the final result in the whole set of files, because all of the past files have an impact in the new state when adding a new file, and every node that makes up this blockchain can and have to prove that the chain is still incorrupt.

Also, using a single repository for all the files, makes them accessible under demand in a very practical way, instead of being processed every single time that you need them.

1.3 Problem

The government file system in Mexico is very inefficient and vulnerable to manipulation, for instance, you need to go through a long, tedious process to register your institution to this system, not to mention that it is not centralized and is also very segmented according to the sector, besides not having the most security measures, making it a very impractical and prone to fakes system. Thus, making use of different tools, such as the cloud, a blockchain that registers every movement, among other things, it looks for an alternative that allows to register and save the government files issued in Mexico in a centralized way, using multiple nodes that prove the validity and integrity of the information and making it accessible at any time.

1.4 Objectives

1.4.1 General Objectives

Through an Ethereum blockchain, a cloud-based file repository, and a web portal, it is going to be implemented a system to store and avoid official government file falsification.

1.4.2 Specific Objectives

Have a repository in the cloud in which the official government files are stored, and from which they can be downloaded on demand using a QR code that is generated when saving the file.

1.5 Methodology / Innovation

The methodology used for this project is a local blockchain with several users and multiple nodes interconnected between them. The transactions saved in this blockchain will register information from the files that are desired to be stored.

The reason why it will be used as a local blockchain is to reduce costs when using a global Ethereum blockchain, in which any transaction to it has a cost.

The network nodes will be simulated throughout different machines with different users, which will be mining simultaneously looking to close the next block in the blockchain.

Besides, each file that is going to be registered in the blockchain, will be uploaded to an AWS S3 bucket, so that it is possible to download it at any time using a QR code generated at the time of registration, after saving such a file.

2. Theoretical Framework

2.1 Symmetric Cryptography

Symmetric encryption is the simplest type of encryption, in which the plain text is processed with a specific key, giving the encrypted message or ciphertext as a result, then, the encrypted message is also processed with the same key, obtaining the original message. So, there are two kinds of functions, encryption, which is turning the plaintext into ciphertext,

$$C = E(K, P)$$

and decryption, in which you get the plaintext back from the ciphertext.

$$P = D(K, C)$$

[5]

Most classical ciphers, which are also symmetric cyphers, work by substitutions, which consist of replacing a letter with a different one, this of course is very easy to break even several years ago, but they were the first ciphers to exist.

Modern ciphers are much more complex, and their security relies heavily on the length of the key used to encrypt the message, but it is not simple, using an extremely long key is unpractical, so there must be a balance.

This complexity is achieved thanks to other concepts, such as confusion, which was introduced by Claude Shannon, this concept means that every single bit of the ciphertext originates from several different parts of the key, making it difficult or ideally impossible to find a relationship between the ciphertext and the key. Thus, the objective of this is that, if even one bit of the key changes, the resulting bits in the ciphertext will be modified in almost its entirety. [15]

Another concept that is very necessary for the security of symmetric encryption is diffusion, this is accomplished by making patterns in the plaintext not apparent in the ciphertext and by hiding the statistical relationship between the plaintext and the ciphertext. [15]

Besides all these concepts, one more way to enhance the security of symmetric encryption is with the use of a one-time password which, as its name suggests, is an alphanumeric string that is used only one time. This is more secure than a user-created password, which can be used across multiple accounts, and even be very weak. [15]

2.2 Asymmetric Cryptography

In contrast with symmetric cryptography, in which it is needed the same key to encrypt and decrypt, that both the sender and the receiver must know, asymmetric encryption uses a key pair, one public and one private, these keys need to have a mathematical relationship between them, that make the original message, when applied both operations with each key, obtained with no changes. The public key, as expected, is of public knowledge, and it is necessary to know it in order to communicate with the owner of a set of key pair, this public key is used to encrypt the original message, while the private need to be only known by the owner, after processing the encrypted message with the private key, it is obtained the original message, thus decrypting is possible.

Also, it is possible to compute the public key from the private key, in an easy way, in contrast, but practically impossible to compute private key from the public key.

[5]

2.3 Hash Functions

A hash function is a process that is applied to a certain input, it can be text, a complete file, or any data entry that, after being processed, is going to give the same output.

One of the most important properties of these functions is that no matter the size of the input, the output size is always fixed to a specific number of characters, depending on the particular hash algorithm used.

Also, these functions only work in one way, which means that, with the same input, the result will always be the same, and even a small change of 1 bit on the input will result in a completely different output, but given an output, it must be impossible to know which was the input.

These functions have a lot of applications, from password storage, so it avoids storing the password in plain text since it is possible to just store the digest message, and compare it to the user input every time such user wants to log in. It also works to verify the integrity of files and information, since, as it was mentioned above, a slight change, no matter the complete size of the input, will result in a completely different output, of a fixed length.

Some characteristics that these functions must comply with are:

- Deterministic: The same input always results in the same output.
- Easy to compute: Given an input, it is not expensive to get the output.
- One way: It is not feasible to get the original input from the output or digest message. To get this, it would have to be necessary to try with every possible input until you get the desired output, and even if you get an input that generates the same output, does not confirm that the input is the same.
- Sensitive to changes: Even a small change (Might be a letter, two permuted characters, etc.), as a result a hash completely different to the one generated with the original input, making it look to be generated with a completely different input.
- Almost 1 to 1 mapping: It is not feasible to get a resulting hash equal to another with a different input. Even though there exist the so-called “collisions”, there are some undesired singularities that are a way to exploit the algorithm, but, even if it happens, it will take a large amount of time for this to happen. [1]

This kind of function is also used in some cipher schemes, using a key pair to sign a message, this was originally achieved from a one-time signature scheme, but the problem with that is that if the same key pair was used more than once, an attacker could easily forge signatures.

2.4 Finite fields

It is a finite set that has defined multiplication, addition, subtraction, and division and satisfies the rules of arithmetic.

The finite fields used in cryptography are fields with any power of a prime number of elements. Let p be a prime and F its field.

A finite field of p elements is expressed as F_p, for cryptography, these fields also need to satisfy the multiplication definition and are commonly expressed as F_p^*.

[4]

A finite field must have these properties:

- Associative
- Identity
- Closed
- Commutative
- Inverse

If the number of elements of a finite field is exactly a prime, or in other words, a prime to the power of 1, then it is called a prime field. On the other side, if the power is more than 1, it is called an extension field, and its elements take the form of a polynomial. For instance, the extension field GF(2^m) where m > 1 the polynomials take the form:

$$a_{m-1}X^{m-1} + \cdots + a_1X^1 + a_0$$

[16]

2.5 Discrete Logarithm

Let g be a primitive root for Fp and let h be a nonzero element of Fp. The Discrete Logarithm Problem (DLP) is the problem of finding an exponent x such that

$$g^x \equiv h \ (mod\ p)$$

In this case, x is considered the discrete logarithm of h to the base of g and is denoted as $log_g(h)$.

[4]

2.6 Diffie Hellman Key Exchange

To exchange a secret key for a symmetric cipher, first, the two parties need to agree on a large prime p and a nonzero integer g modulo p

In this case, the values of p and g can be publicly known. It is recommended that the chosen number g has a large prime as its order in F_p^*.

Then, the first party needs to pick a secret integer a that is not revealed to anyone, while the second party picks another integer b that is also kept secret. Each of the parties computes the next values.

$$A \equiv g^a (mod\ p) \qquad B \equiv g^b (mod\ p)$$

Then, each value is shared through the insecure channel to the other party, once received the counterparts, each of the parties use their secret integers to compute the following.

$$A' \equiv B^a (mod\ p) \qquad B' \equiv A^b (mod\ p)$$

These final computed values are the same since

$$A' \equiv B^a \equiv (g^b)^a \equiv g^{ab} \equiv (g^a)^b \equiv A^b \equiv B' \quad (mod\ p)$$

[4]

2.7 Elliptic Curves

An elliptic curve is the set of solutions to any equation of the following form.

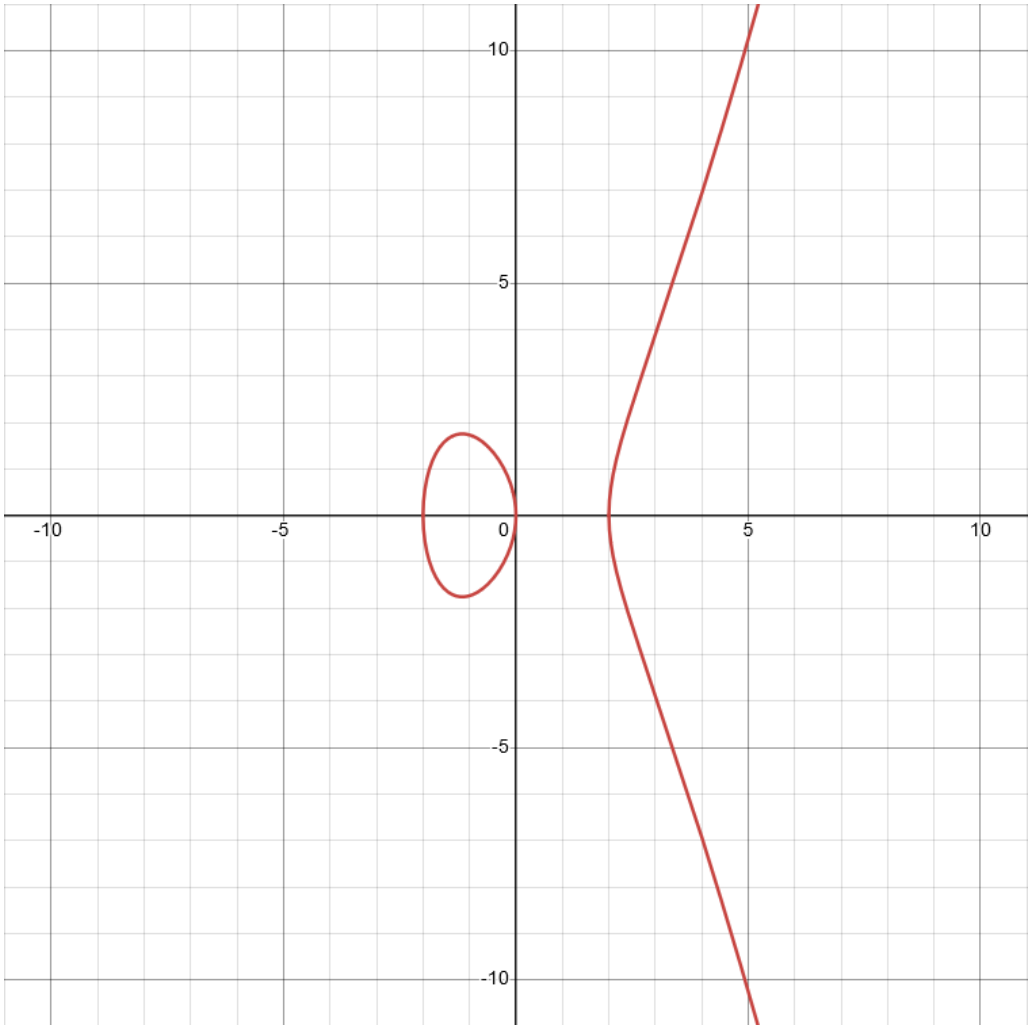
$$Y^2 = X^3 + AX + B$$

The equations of this type are called Weierstrass equations.

What is useful about this kind of equation is that you can produce a point in the curve by “adding” two points.

[4]

All these points have (x, y) coordinates that satisfy the curve’s equation $y^2 = x^3 - 4x$. For example, when x = 0, then $y^2 = x^3 - 4x = 0^3 - 4 * 0 = 0$; hence, y = 0 is a solution, and the point (0, 0) belongs to the curve. Likewise, if x = 2, the solution to the equation is y = 0, meaning that the point (2, 0) belongs to the curve.



It is crucial to distinguish points that belong to the curve from other points, because when using elliptic curves for cryptography, we'll be working with points from the curve, and points off the curve often present a security risk. However, note that the curve's equation doesn't always admit solutions, at least not in the natural number plane. For example, to find points with the horizontal coordinate x = 1, we solve $y^2 = x^3 - 4x$ for $y^2$with $x^3 - 4x = 1^3 - 4 * 1$, giving a result of –3. But $y^2 = -3$ doesn't have a solution because there is no number for which $y^2 = -3$. (There is a solution in the complex numbers, but elliptic curve cryptography will only deal with natural numbers—more precisely, integers modulo a prime.) Because there is no solution to the curve's equation for x = 1, the curve has no point at that position on the x-axis.

[5]

2.8 Secure storage

With the emergence of cloud computing, data storage security in the cloud has received widespread attention. People on the one hand, hope to be able to use large cloud storage services to alleviate the pressure of local storage, on the other hand, worry that cloud service providers may provide confidential information to a third party without authorization, resulting in data information leakage. Data saved in the cloud are required to be encrypted to guarantee a certain security.

Cloud computing is generally thought to include the following several levels of service: infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS). IaaS provides IT infrastructures as a service over the Internet. PaaS provides a computing platform as a service to support cloud applications. SaaS allows users to use Cloud applications without installing and running software on their computers. Data owners have limited control over IT infrastructures. This also means that effective data may be revealed and that the deleted data may still be recoverable. However, the file system is responsible for organizing logical orders of data which is stored among the available blocks on the physical medium, and the file system layer which can call for the device driver interface allows files to be read, written, created, and deleted. These features led people to consider using it to implement secure storage of data file systems. Joel Reardon illustrated some interfaces and layers involved in a physical storage medium in detail.

To make the data owner reserve an element of control for the file that they outsource to the cloud, access control is proposed. Access control is an important means of realizing user data confidentiality and privacy protection.

Data integrity is the basic requirement of data storage. To ensure that the data stored in the cloud has not been tampered with. So, the validation of the data integrity is necessary. CSP may hide a management incident of data loss. In addition, some attackers who consider their interests may compromise several cloud storage servers to modify or delete users' data out of CSP's detection.

Data in the form of cipher text is stored in the cloud, but still, there may be secret keys leaked or a collusion attack. So the study of user traceability is also essential. Here only the security of the users to store files is considered, regardless of the data in a database or other types of data.

[10]

2.8.1 Distributed Systems

Decentralization is a concept that declines the notion of a centralized authority. It defends the distribution of power, control, and responsibilities when making decisions between the multiple nodes or users in the system. By contrast with the traditional centralized systems that government entities now have, the decentralized systems operate through consensus mechanisms, effectively eliminating the necessity of a central control point. This structure offers multiple benefits, such as solidity against attacks, censorship resistance, as well as more user privacy.

Distributed systems, even though they have some parallels with decentralization, emphasize the computational work dissemination throughout multiple nodes in the network. This distribution permits simultaneous processing, increasing efficiency and optimizing resource usage. Distributed

systems can operate with or without a central authority, a key distinguishing factor of decentralized systems.

These kinds of systems stand out in scalability and speed, but they require complex management and coordination protocols. Besides, they can be susceptible to vulnerabilities if they depend on a central authority. [3]

2.8.2 Byzantine consensus

To explain this concept, we begin defining what consensus is, which could be considered as a general agreement, and when we talk about a network such as a blockchain in which there is no authority and the "truth" is decided by the majority of nodes, it must be ensured that even if there are failures, that eventually there will be, the system can overcome this errors, to accomplish this, it is used the byzantine failure tolerance. This method has an asynchronous approach, and must meet the following criteria:

- The nodes are assembled in a certain sequence.
- There must be a "leader" node, the remaining nodes are backed up.
- The "leader" node responds to the client requests and is the moderator between such client and the backup nodes.
- The nodes can communicate between them to verify that such nodes are honest.
- The considered honest nodes come to an agreement, based on what the majority determines, to get to the next change in the network.
- The source of the message is verified to ensure that it has been sent from the right sender.
- It is guaranteed that the message has not been neither modified nor damaged in the medium.

[6]

2.9 Blockchain

Blockchain, as its name indicates, is a sequence of different transactions that are grouped in each of the so-called blocks, every one of them ordered between them. The order is immutable since each one registers several actions done in a determined time in the past. The innovation that it brings is that, unlike any other kind of transaction, these do not need a regulatory authority that validates the veracity of such transactions, because everything is managed in a peer-to-peer decentralized network, in which any one of the nodes has a register of the transactions, making it really difficult to add fake transactions, since any small change in any node would be very easily detected. [2]

The four key concepts behind blockchain are:

- Shared ledger. A shared ledger is an "append-only" distributed system of records shared across a business network. "With a shared ledger, transactions are recorded only once, eliminating the duplication of effort that's typical of traditional business networks."
- Permissions. Permissions ensure that transactions are secure, authenticated, and verifiable. "With the ability to constrain network participation, organizations can more easily

comply with data protection regulations, such as those stipulated in the Health Insurance Portability and Accountability Act (HIPAA)" and the EU General Data Protection Regulation (GDPR).
- Smart contracts. A smart contract is "an agreement or set of rules that govern a business transaction; it's stored on the blockchain and is executed automatically as part of a transaction."
- Consensus. Through consensus, all parties agree to the network-verified transaction. Blockchains have various consensus mechanisms, including proof of stake, multi-signature, and PBFT (practical Byzantine fault tolerance).

Each blockchain network has various participants who play these roles, among others:

- Blockchain users. Participants (typically business users) with permission to join the blockchain network and conduct transactions with other network participants.
- Regulators. Blockchain users with special permissions to oversee the transactions happening within the network.
- Blockchain network operators. Individuals who have special permissions and authority to define, create, manage, and monitor the blockchain network.
- Certificate authorities. Individuals who issue and manage the different types of certificates are required to run a permission blockchain.

[8]

2.9.1 Honest Majority

The honest majority consists in the assumption, in Proof-of-Work based blockchains, that most nodes or participants in a blockchain are honest and comply with the protocol.

There can be attacks on the protocol if the majority of participants are dishonest, causing negative effects.

[12]

2.10 Ethereum

Ethereum is a network around the world that follows a set of rules known as the Ethereum protocol. This network acts as the base of communities, applications, organizations, and digital files that anyone can build and use.

Everyone can build applications on an Ethereum network, making it possible to control your assets and identity.

Ethereum is not controlled by a particular entity. If there is a computer following the Ethereum protocol and adding to the Ethereum blockchain, Ethereum exists. Each computer connected and contributing to the network is known as a node.

Also, anyone can suggest some changes to the code and make it better.

[13]

Ethereum uses a blockchain, which is a distributed ledger. Information is stored in blocks, each containing encoded data from the block before it and the new information. This creates an encoded chain of information that cannot be changed. Throughout the blockchain network, an identical copy of the blockchain is distributed. [22]

Proof-of-stake differs from proof-of-work in that it doesn't require the energy-intensive computing referred to as mining to validate blocks. It uses a finalization protocol called Casper-FFG and the algorithm LMD Ghost, combined into a consensus mechanism called Gasper. Gasper monitors consensus and defines how validators receive rewards for work or are punished for dishonesty or lack of activity. [22]

Validators who act dishonestly are punished under proof-of-stake. Those who attempt to attack the network are identified by Gasper, which flags the blocks to accept and reject based on the validators' votes. [22]

2.11 Smart contracts

Smart contracts, as its name suggests, are contracts capable of being executed by themselves, that is, they do not need the intervention of one or more third parties, and it can be compared to a computing program more than a regular or traditional contract. They can be described as information bits that contain logical information stored in the blockchain and all its nodes.

Many blockchain smart contracts currently use Ethereum, and the most common token standards are ERC-20, ERC-721, ERC-777, and ERC-1155. [21]

Token standards are the set of rules, conditions, and functions that dictate how a crypto token works. Before delving into token standards, it is helpful to understand the mechanisms of a smart contract standard. Fundamentally, smart contract standards are rules that a smart contract must comply with, to function as intended on the underlying blockchain network. [21]

Ethereum Request for Comment (ERC) is essentially a set of technical documents containing guidelines on developing a smart contract. They define a specific set of functions for each token type and facilitate the interaction between applications and smart contracts. [21]

The ERC-20 token standard is a blueprint for creating fungible tokens on the Ethereum network. [21]

Fungible means that each token (or a fraction of a token) is equivalent to and indistinguishable from, another token. For example, fiat currency is fungible. Alice’s US$1 is equal to Bob’s US$1, and both of them can swap their US dollar because it has the same value. Similarly, 1 ETH has the same value as any other ETH. [21]

Like other digital assets, ERC-20 tokens are mainly developed by organizations and tech-focused companies. These tokens allow the entity to customize their utility, such as granting voting rights or rewarding mechanisms. [21]

In contrast with ERC-20, a fungible token standard, ERC-721 is a token standard for non-fungible tokens (NFTs), which are a special type of cryptographic token not mutually interchangeable by their specification. This means that one token cannot be exchanged for another because of its unique specifications. Therefore, NFTs are utilized to represent digital collectibles, game items, digital art, event tickets, domain names, and ownership records for physical assets. [21]

2.12 Blockchain attacks

There are multiple kinds of attacks on a blockchain, but some of the most common ones are:

51% Attack: This attack occurs when certain miner or miners get more than 50% of the processing power of one blockchain, which grants this individual or individuals the majority of consensus.

Sybil Attack: In a Sybil attack, the objective is to create a large number of false identities or pseudonyms to undermine authority or to obtain a considerable large influence. This can also lead to 51% attacks.

Timejacking Attack: This attack is a variant of the Sybil attack, in which the attacker can flood the network with nodes reporting the desired timestamp, causing the network to revert the system time to this since most of the nodes are reporting the fake timestamp.

Selfish Mining Attack: This attack relies on the “longest chain rule”, which dictates that the network will shift to the chain that has the most work. The attacker mines several blocks, but it does not “post” them until there is a fork that is sufficiently ahead of the network in terms of the length of the chain.

Finney Attack: The attacker mines a block stealthily and sends the unconfirmed transaction to the other node. If the other node accepts the transaction, then the attacker can further add a new block to the chain in a short time frame, reversing that transaction and inducing a double spending attack. The attack window in the case of a Finney attack is considerably small, but this can cause a lot of damage if the value of the transaction is large enough.

Race Attack: In this attack, the attacker broadcasts two different transactions, one of them to some node and the other one to the network. If the node that received the different transaction gets the illusion that such transaction is the first one, it accepts it, and then the attacker broadcasts a completely different transaction to the entire network.

[14]

2.13 Communication APIs

An API (Application Programming Interface) is like a bridge that allows different software applications to communicate with each other.

So, a Blockchain API is a set of rules and protocols that enable external software to interact with a blockchain network.

Blockchain API acts as a toolkit for developers. It allows developers to complete tasks using code, such as accessing data, managing transactions, and retrieving information. They can easily integrate blockchain features into their apps without starting from scratch.

[15]

- The main characteristics that an API should have are:
- Simplicity: APIs should be easy to use and maintain.
- Flexibility: APIs should be reusable for different projects.
- Scalability: An API service must adjust to the number of requests without interrupting its availability.
- Security: The data transported on the internet must be secured at all times.
- Speed: An API must give a response with optimum speed, despite the number of requests.

Some full-node providers for blockchain are:

- Alchemy: It is a blockchain node provider known for offering extremely high reliability compared to competitors and a proven track record of data accuracy. In addition to providing blockchain nodes for developers, Alchemy also offers a suite of development tools such as APIs, and a web3 SDK.
- Chainstack: Chainstack offers access to custom node parameters and node customizations; nodes are optimized for low-latency, production-grade workloads. Similar to other providers on this list that support enterprise customers, Chainstack also provides monitoring solutions to customers using their nodes.
- Pokt Network: Pocket Network is a decentralized API with a network of decentralized nodes. Pokt is created and hosted by developers who own POKT tokens in return for supporting a node. Pokt Network's main value propositions are the decentralized nature of the network, and the high number of supported chains, which results in increased flexibility for dApps using Pokt's blockchain nodes.

[23]

3. State of the art

3.1 Cloud Storage

Cloud computing is the next era of Internet technology that provides users with all the services they require over the Internet, such as computer resources, computing infrastructure, implementations, and business processes. With security as the main aspect of choosing what is the best for storing data online, many technologies are being developed to maintain safe and secure cloud storage.

Data storage prototype in which digital data is kept in logical pools, physical storage is spread across multiple servers, and the physical environment is typically owned and managed by a hosting company. Cloud storage companies are in charge of ensuring data availability and accessibility, as well as the security and functionality of the physical environment. These three aspects, availability, accessibility, and functionality determine the level of security of the data storage. People and businesses purchase or rent storage space from providers to store huge amounts of data.

Cloud storage architecture is a great new technological advancement that will allow customers to store their data safely and reliably. Companies can use multiple data servers and multiple copies of information to prevent data loss and server downtime. A distributed file system, a network, and other storage middleware cluster thousands of storage devices together to provide cloud storage services to users. Cloud storage structure includes a huge resource pool, service level agreements (SLA), a distributed file system, and service interfaces, among other things.

Cloud storage is a service model in which data is transmitted and stored over the internet on and can be accessed remotely. Before data is made available to users via the network, it is restored and maintained on these remote storage systems. In cloud computing, there are three basic service types: SaaS is a model in which Cloud Service Providers make it available to the users via the Internet (CSPs). Platform-as-a-Service (PaaS), in which CSPs provide platforms on which Cloud users can develop and deploy their applications, and Infrastructure-as-a-Service (IaaS), in which CSPs provide compute, storage, network, and other computing resources to Cloud users.

[9]

3.2 Estado de la firma electrónica en México

The use of the Electronic Signature arises from the need of organizations to reduce costs and increase the security of their internal processes, through the use of electronic means to streamline processes, reduce time, and avoid the use of paper.

The Model Law was examined and ruled during the 38th session of the Working Group, Electronic Commerce of UNCITRAL, held from March 12 to 23, 2001; having as a target to allow or facilitate the use of electronic signatures and equal treatment to the users of physical documents (in paper) and electronic documents.

On March 30, 2017, the reform of NOM-151-SCFI-2016 (In Mexico), “Requirements to be observed for the preservation of data messages and document Digitalization” was published in the Official Journal of the Federation, through which the necessary elements are established that describe the processes involved in the digitalization of documents in physical support to data messages with the purpose of their conservation.

What is the difference between the existing types of electronic signatures?

Simple electronic signature. It is the basic type of electronic signature. It is a set of electronic data, along with an electronic document used when an issuer sends a message to the receiver, and that message is encrypted so that nobody can modify it or alter it. Its purpose is also to identify the subject that they use.

The electronic signature is based on an asymmetric cryptography system.

The sender of the encrypted message has a public key, assigned by an authorized body for that purpose, and employing this key the message is encrypted, guaranteeing its integrity. The recipient of the encrypted message also has another key, but this one needs to be private, and he is the only one who possesses it, which makes the message impossible to decipher by anyone else besides him.

Advanced Electronic Signature. Like the basic electronic signature, this type of signature is a set of electronic data to identify the sender of a message, as well as the integrity of it. However, the difference with the simple electronic signature is that this model of signature is created under a series of control means that are under the direct control of the signer of the same. That is, it is a more secure method of authentication and identification of the signer.

In other words, it is a public key infrastructure technology (PKI), which allows to exchange of information and performing transactions in an agile and simple, through online systems and the use of a digital certificate, through mechanisms that provide certainty and technical security with them legal effects that an autograph signature.

[11]

3.3 Use of Blockchain in Mexico

The primary current example of blockchain use in education have related to diploma management and achievement assessment. Certification is a primary use of blockchain, and thus it behooves us to understand better the certification process. A certificate often includes a signature, such as a unique symbol, stamp, image, or code, which can only be affixed by the issuer, thus confirming their identity. The certification process does not end when an authority issues a certificate recording the evidence and signature. To the contrary, built into the process is the need for a third party to verify the authenticity of the certificate. Verification ranges from the simple (third party contacts original issuer) to the more complex (security measures like the inclusion of special security paper or signatures on the certificate). Moreover, a certificate is only as good as the people who are aware of it. [19]

Blockchain simplifies this process by keeping a list of issuer and receiver of each certificate, together with the document signature (hash) in a public database (the blockchain) which is identically stored on thousands of computers around the world. Digital certificates that are thus secured on a blockchain hold significant advantages over `regular' digital certificates, in that they cannot be forged or otherwise changed. Moreover, recipients of certificates, authorities, governments, institutions, and employers all have access to this secure material. [19]

The basic education pilot program was a public-private partnership. @prende was the overall lead on the government's side, approving the topics and content and liaising with state governments. The state government invited teachers and principals from a set of schools that had previously worked with @prende. Google provided its education suite and content. UNETE provided the actual training sessions, while Tomate Corp. registered the activities in the Ethereum blockchain. Finally, ITAM, a leading private university in Mexico, designed the methodology and led blockchain implementation. The pilot involved more than 70 onsite training courses, reaching 1,358 teachers in 14 different states. In the second phase, 8 on-site training courses reached 160 teachers from 6 different states. blockchain was integrated throughout, with UNETE (and Tomate) registering teacher information for future use by the teachers themselves, the government, schools, future employers, and the wider public. [19]

3.4 Use of Blockchain in Africa

Blockchain has evolved from the first generation to the third generation. While the first (blockchain 1.0) and second generation (blockchain 2.0) were based on the application of blockchain in the financial sector, the third generation (blockchain 3.0) is particularly focused on other sectors besides the financial sector. Several applications of blockchain in the healthcare sector have proved that the technology can be beneficial in many ways. Perhaps, the major advantage of blockchain is that each set of transactions called a block within the blockchain is validated through encryption algorithms called hash algorithms. Thus, each previous block of transactions is validated before a subsequent transaction commences. The use of hash algorithms ensures the authenticity of each transaction within the blockchain as once the transaction is encrypted, it cannot be altered. Hence, transactions within the blockchain are immutable, that is, such transactions cannot be retrieved or altered. In case a record needs to be updated, a new record must be created and hence blockchain is labeled as an append-only ledger. In addition, all transactions within the blockchain are time-stamped, thus enhancing transparency and accountability in the transaction processes. In this case, healthcare stakeholders can monitor how healthcare data is used and when it is used. In addition, in a case whereby one node within the blockchain is compromised, it does not affect the entire ledger as the information within the ledger is replicated across many nodes within the distributed network. Blockchain is immutable and

hence, patients' health records may not be tampered with while being transmitted. Furthermore, since blockchain uses public-private keys to encrypt data, the identities of patients may not be revealed when personal health records are shared among healthcare stakeholders. Moreover, blockchain enables a patient to decide how his/her information is used or shared through smart contracts. [18]

Blockchain technology characteristics such as immutability and time-stamped ensure the authenticity and traceability of all transactions. Therefore, in the context of South Africa, blockchain can help to address issues related to accountability, unauthorized modification of medical prescriptions, and corruption in the supply chain and financing of the healthcare system. In addition, the automated processes within the blockchain can alleviate the workload associated with various validation processes within the entire healthcare continuum and clinical trial research. It is also believed that blockchain can enhance transparency in the management and administration of patient-centered services. This paper focuses on the potential use of blockchain to enhance transparency and accountability in providing of patient-centered healthcare. [18]

In South Africa, the major components of public healthcare are public hospitals and primary healthcare centers. Public hospitals include district, regional, tertiary, central, and specialized hospitals. The South African public healthcare sector provides free and affordable healthcare services to most of the South African population. The private healthcare sector is generally not affordable and hence not accessible to most of the population. [18]

A person usually reports to a public hospital for treatment. Generally, a healthcare professional will capture patients' details. Personal details are then incorporated into a blockchain. Since the patients' details are the first set of transactions with the patient's healthcare, we will call this block 1. Block 1 will contain the patient's details encapsulated within a hash function. Then, the patient is referred to a medical doctor within the same hospital. The doctor provides the diagnosis and prescribes medications. These transactions performed by the doctor are incorporated in the blockchain as block 2 with a corresponding hash algorithm and a hash of block 1 header. When the patient gets to the pharmacy, the pharmacist will perform transactions that are encapsulated in block 3 within the blockchain together with the hash algorithms of block 1 and block 2 headers. Thus, in this case, blockchain provides an audit trail whereby all the transactions are secured i.e. cannot be changed or undone and can be traced with time stamps. [18]

The use of blockchain as presented in this scenario can enhance transparency and accountability in many ways. As each transaction is recorded and time-stamped, this provides irrefutable proof of who conducted the transactions and when the transactions were conducted. This enables traceability of all transactions related to a patient's healthcare. This will assist in forensic audits to hold stakeholders in patient healthcare accountable as the integrity and validity of transactions within the patient's healthcare continuum can be verified. Should any data within the patient's records be tampered with, the hash algorithm will change and will be flagged. In addition, should a patient case need to be escalated from a district to specialized care in a specialized hospital, all is needed is to add another block to the existing blockchain network instead of creating a new blockchain network hence increasing traceability of the patient's treatment history. This will therefore ease the patient's referral process. In addition, if a blockchain with permission is used, the patient is in control, and he/she can decide who to share this information or not (through public-key cryptography). [18]

3.5 Use of Blockchain in Asia

Firms in many Asian economies have started to incorporate blockchain in smart contracts. Many high-profile projects are in China, where local and foreign companies have teamed up to develop blockchain-based smart contract solutions. This can be partly attributed to the Chinese government's initiatives to move to a higher technological gear. For instance, the Chinese President has called blockchain a “breakthrough” technology. More than 40% of Chinese startups that received government funding for breakthrough technologies in the first quarter of 2017 were blockchain-related. [17]

IBM, Walmart, and Chinese e-commerce company JD.com were reported to be working together with Tsinghua University National Engineering Laboratory for E-Commerce Technologies to improve food tracking and safety in China. They announced a Blockchain Food Safety Alliance. The goal is to create a “standards-based method” to collect data about the origin, safety, and authenticity of the food. The system will provide real-time traceability throughout the smart contracts. [17]

Provenance

The British tech start-up Provenance conducted a pilot project in Indonesia to enable traceability in the fishing industry. By using mobile phones, blockchain, and smart tagging, Provenance tracked fish caught by fishermen. The pilot successfully tracked fish in Indonesia for the first six months of 2016. Indonesian fishermen who participated in the pilot project sent simple text messages to register their catches. For each registered catch, a new blockchain-based digital asset is created. Each batch of fish that passed through a smart contract supply consisting of traders, processors, brands, and supermarkets had a blockchain-based ID. The ID also provided audit information to prove that the fish were caught legally and sustainably. [17]

Alibaba

Alibaba teamed up with New Zealand's Fonterra and New Zealand Post and Australia's Blackmores and Australia Post to develop a blockchain-based Food Trust Framework. The goal is to develop a blockchain solution model that participants across the smart contract can use. Alibaba's international marketplace Tmall Global uses blockchain and product tagging with unique QR codes to track and monitor food products and make the information available to consumers. Each step in the smart contracts is authenticated and verified. [17]

3.6 Use of Blockchain in Latin America

The digitization of payment processes in the context of electronic government (“e-government”) promotes financial inclusion and efficiency in private-sector business processes. Payments

related to affidavits, fees, contributions for regulations, taxes, and fines are frequently exposed to the public in both government-to-citizen and government-to-business contexts. Concerns such as non-repudiation in affidavits, data consistency, and immutability of payments thus become highly important when attempting to inspire greater societal confidence in government. [20]

A software architecture that integrates a legacy information system with a blockchain application programming interface (API) provider requires a formal method to achieve architectural drivers. The legacy solution in the context of this paper is a Java web application that registers income tax affidavits for a governmental agency in charge of collecting payments. The blockchain-as-a-service (BaaS) technology platform provides a third-party way to access an architecture whose main configuration implies a set of distributed nodes that keep identical copies of a ledger. [20]

Governmental agencies see the integration of legacy applications and blockchain platforms as deficient when considering performance as the main quality attribute. Software architecture design that joins legacy and BaaS platforms must prioritize architectural patterns and tactics that promote shorter response time. [20]

The organization responsible for the RCS determined that there was a need to ensure that affidavits were immutable and would not be rejected by taxpayers. Specifically, it was important that data like registration dates, income amounts, automatically calculated payment amounts, and arrears not be altered and remain permanently consistent. For this reason, the governmental agency decided to join the RCS as a data-as-a-platform (DaaP) solution in a BaaS context. [20]

4. Development and methodology

It is necessary to have several components for this project, we need to have the web UI in which the form with all the fields as well as the files will be filled, the S3 instance in which the files will be stored, and the nodes that are going to be mining for the blockchain.

4.1 Deploying of the web portal in an EC2 instance

First, from the AWS console, we created our Ubuntu instance, since it is one of the best Linux distributions, with support and a wide documentation in case of some inconvenience.

Once we are logged into the AWS Console, we move to the EC2 menu, which will look exactly like Figure 1.

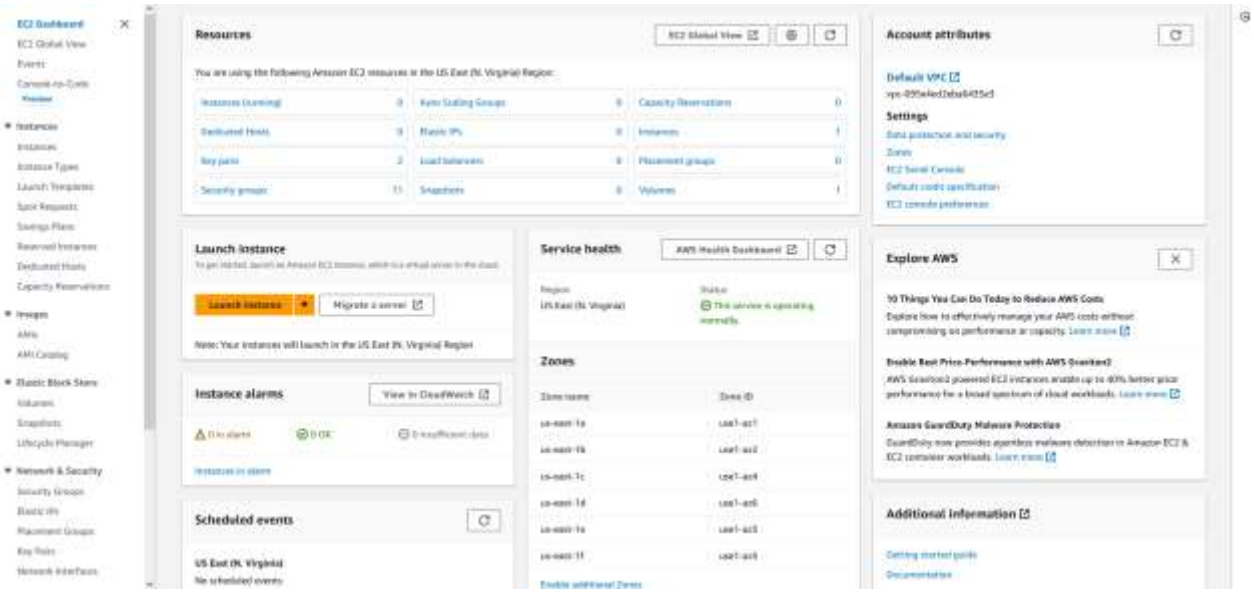


Figure 1. EC2 Menu

Then, we click on the button “Launch instance”, which will lead us to the next menu, shown in Figure 2.

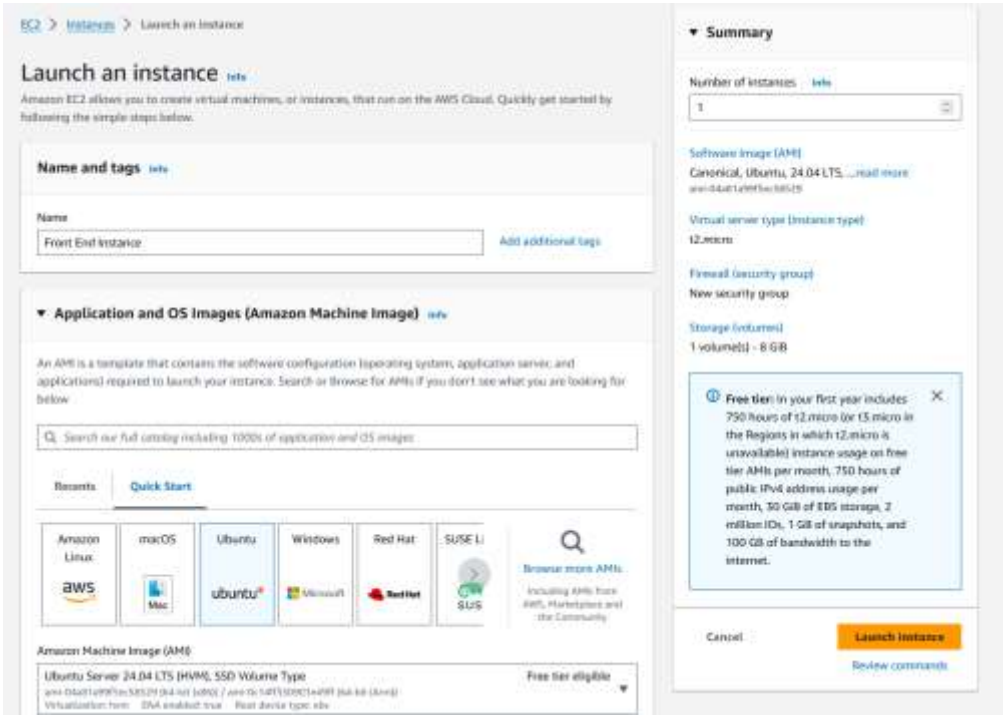


Figure 2. EC2 Instance launching menu

We need to type the name of our instance, select the Ubuntu image, as well as the parameters for our machine, we decide to use a t2.medium processor since it is the most balanced between cost and performance and it is going to be enough to host our web portal using Apache service. (As the official AWS documentation page “https://aws.amazon.com/es/blogs/aws/choosing-the-right-ec2-instance-type-for-your-application/” suggests) We also select the key pair necessary to log into this instance through SSH protocol, if we have not created one yet, we create a new one and save it in a place that we can access and remember easily.

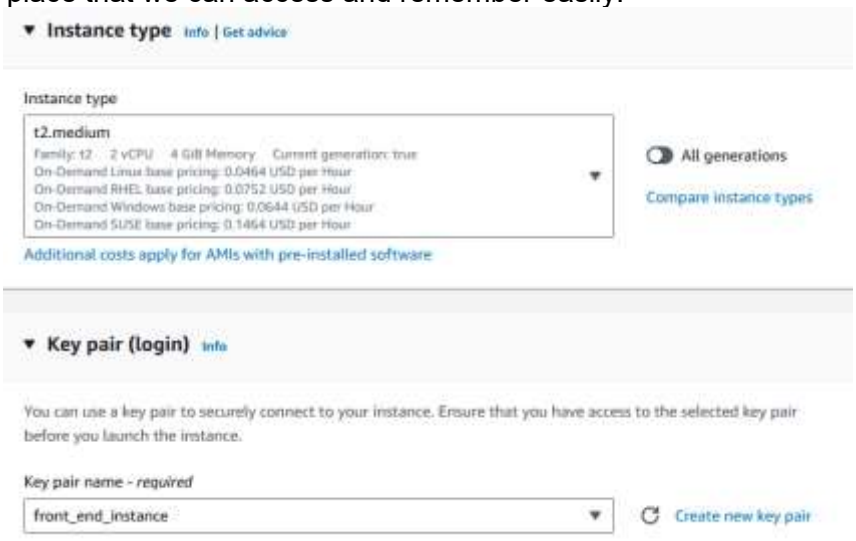


Figure 3. EC2 Instance parameters

Finally, we click on Launch Instance, wait for the environment to load and our Instance will be ready.

Figure 4. Instance created and running

Then, we connect to the instance using the ssh command if we are on MacOS or Linux distribution, or by using PuTTY if we are using Windows, once we are logged in to our instance, we update the packages to make sure everything is up to date.

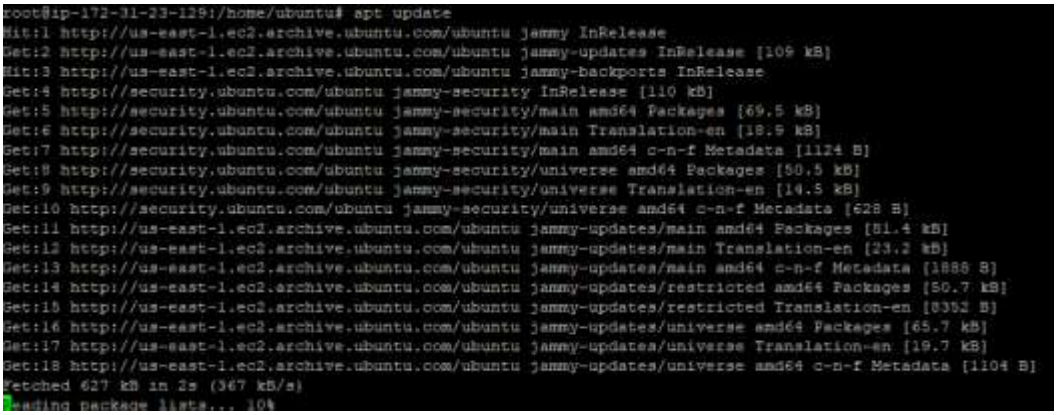


Figure 5. Packages update

Then we install all the necessary libraries like the Apache service which will be what runs our web portal, as well as git tools to clone our repository. The list of libraries is:

- apache2
- git
- python3
- python3-pip
- libapache2-mod-wsgi-py3

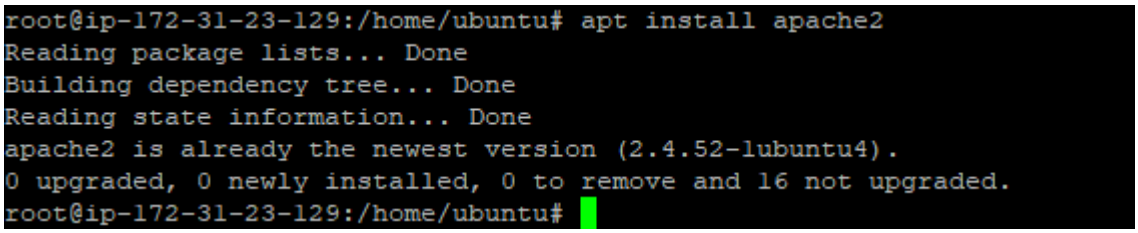


Figure 6. Installation of Apache service

The web UI is built by a Python application and a framework named Flask, this project is on the following GitHub repository: https://github.com/emanuelrobledo/doc_bc, we clone it with the following command:

```
git clone git@github.com:emanuelrobledo/doc_bc.git
```

```
root@ip-172-31-23-129:/home/ubuntu# cd /home/desarrollos/doc_bc/
root@ip-172-31-23-129:/home/desarrollos/doc_bc# ls -la
total 68
drwxrwxrwx 8 root     root     4096 May  9 23:32 .
drwxr-xr-x 3 root     root     4096 May  4 06:02 ..
drwxrwxrwx 8 root     root     4096 May  9 23:33 .git
-rwxrwxrwx 1 root     root      268 May  4 06:02 EmanuelrobledoMoran.json
drwxrwxrwx 2 root     root     4096 May  9 23:32 __pycache__
-rwxrwxrwx 1 root     root       74 May  4 06:47 ejemplo.txt
drwxrwxrwx 2 root     root     4096 May  9 23:33 files
-rwxr-xr-x 1 root     root     3379 May  9 03:50 main_flask.py
-rwxrwxrwx 1 root     root      209 May  4 06:09 main_flask.wsgi
-rwxr-xr-x 1 root     root      795 May  9 03:36 manage_json.py
-rwxrwxrwx 1 root     root      530 May  4 06:09 portal-prod.conf
drwxrwxrwx 3 root     root     4096 May  4 06:02 static
drwxrwxrwx 2 root     root     4096 May  9 23:33 templates
-rwxrwxrwx 1 root     root      577 May  4 06:02 test_qr.png
-rw-r--r-- 1 www-data www-data  398 May  9 23:33 transactions.txt
-rw-r--r-- 1 root     root     1510 May  9 23:32 upload_file_s3.py
drwxrwxrwx 4 root     root     4096 May  4 06:02 venv
root@ip-172-31-23-129:/home/desarrollos/doc_bc#
```

Figure 7. Frontend App directory

Then we install Python 3 (which is the main language used for the web server), mod_wsgi which is a connection between the python flask app and Apache.

```
apt -y install python3 python3-pip libapache2-mod-wsgi-py3
```

The repository has already a ".conf" file that has the configuration for the apache2 service to run, we just need to copy it to the apache2 directory, indicating the location of the WSGI file, the port used for the applications, among other general things. The content of such a file is the following:

```
<VirtualHost *:80>
     # Add machine's IP address (use ifconfig command)
     # Give an alias to to start your website url with
     WSGIScriptAlias / /home/desarrollos/doc_bc/main_flask.wsgi
     <Directory /home/desarrollos/doc_bc/>
            # set permissions as per apache2.conf file
           Options FollowSymLinks
           AllowOverride None
           Require all granted
     </Directory>
     ErrorLog ${APACHE_LOG_DIR}/error.log
     LogLevel warn
     CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

After copying the configuration, we restart the apache2 service via the command:

```
service apache2 restart
```

Since the EC2 instance is already running, there is assigned a public IPv4 address used to access the web portal, it is in the column "Public IPv4 address" in the instances list (the last one from left to right in the example), as shown in figure 8

Figure 8. UI Login Page

Once the above steps are completed, we access to the UI by placing the IP of the instance followed by the port specified in the configuration file in any browser of our preference, we should be able to log in.

Figure 9. UI Login Page

Besides the web server, there is needed a place where all our files will be located, for that, we need an S3 bucket, so we create one through the AWS console.

4.2 Creation of the S3 Bucket

We need to type the name of our S3 Bucket, as well as select the region, we should select the same as our EC2 instance, we decided to use the region “us-east-1” since it is the cheapest and closest, thus with the lowest latency, to our country, we select the recommended option about object ownership since this will provide an extra layer of security in the objects uploaded because only the account that created the bucket will have access to what it contains.

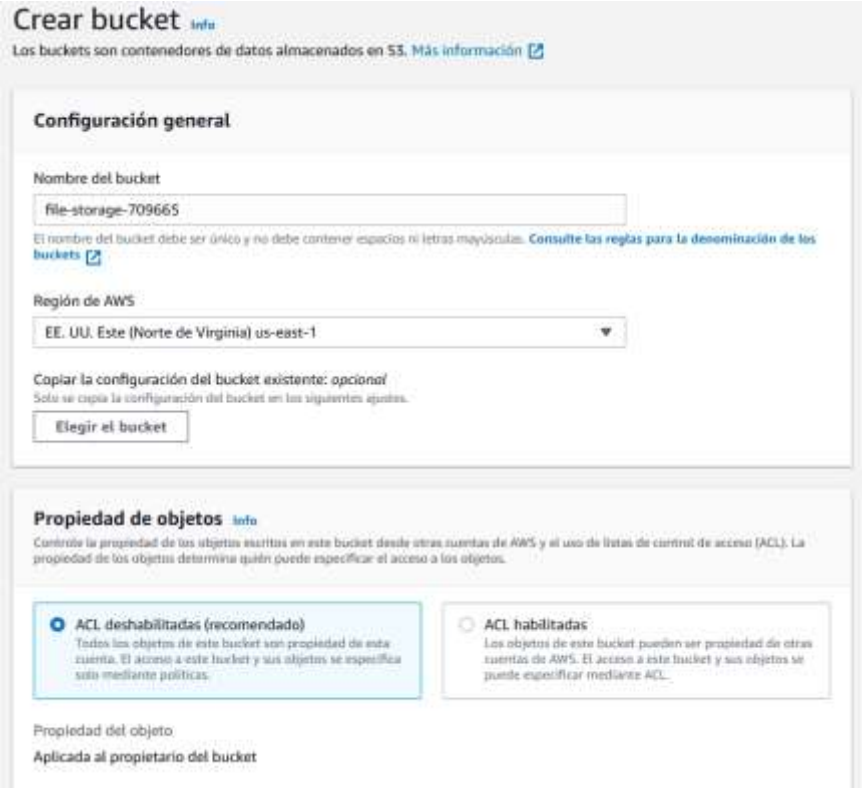


Figure 10. S3 Bucket creation

We do not need any of the special configurations, so we keep everything as it is and click Create Bucket

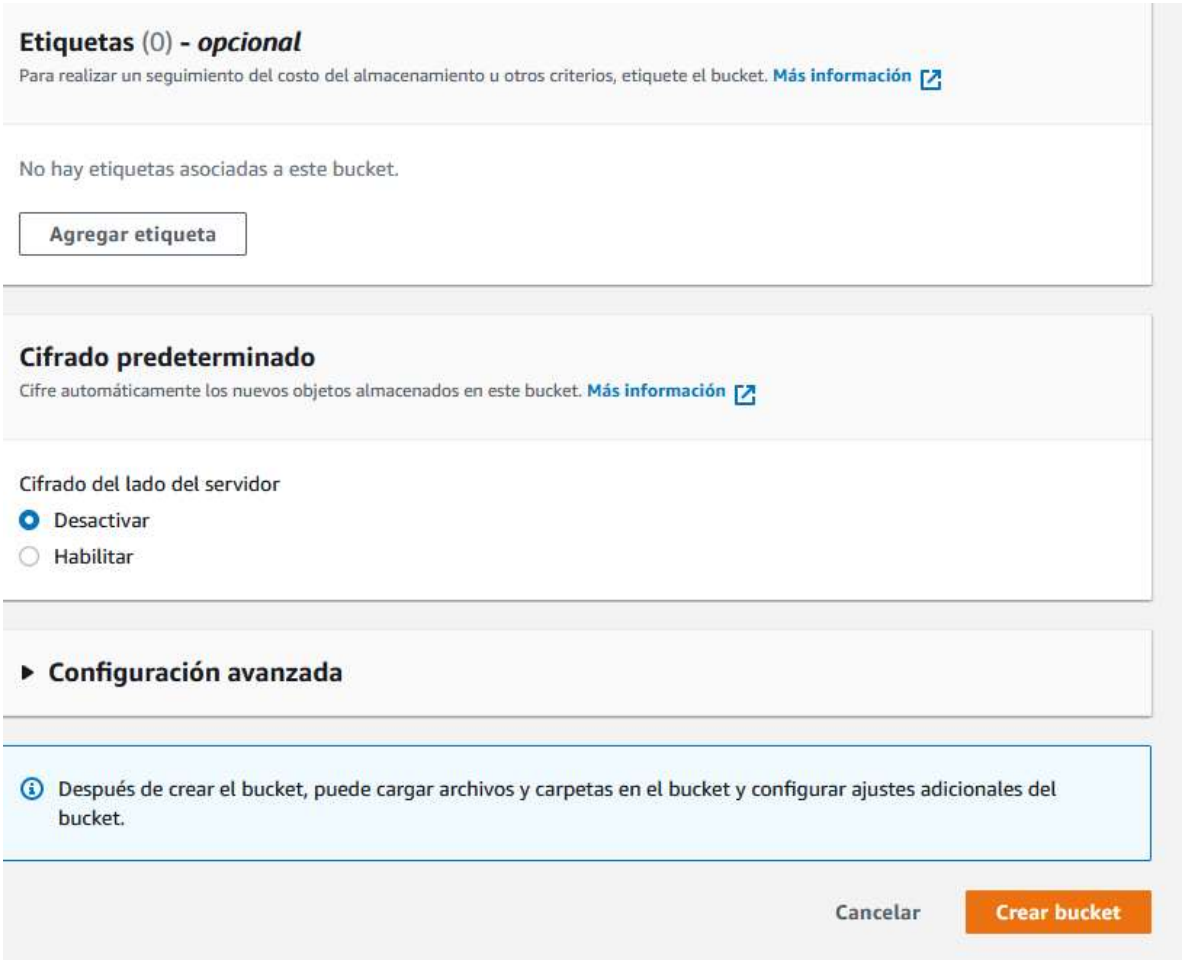


Figure 11. Bucket Parameters

After this, our bucket is created as can be seen in Figure 12, at the beginning we will not have any files, the ones shown are associated with the project and are only an example.

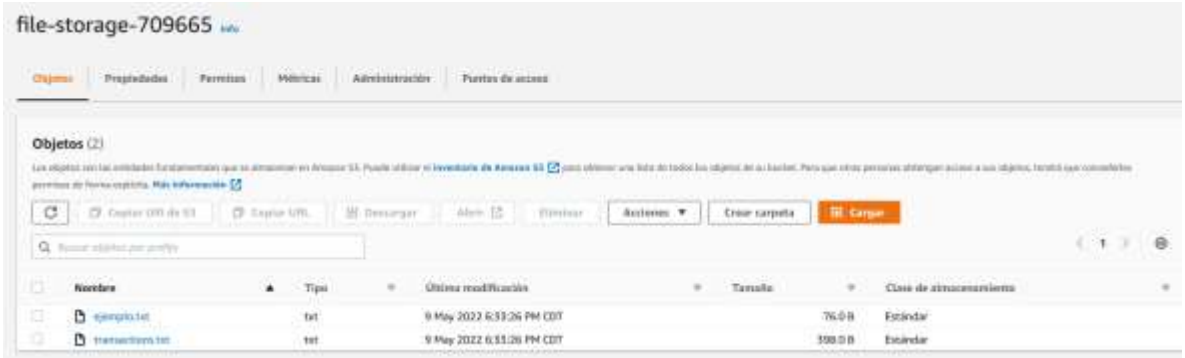


Figure 12. Bucket created

4.3 Example of usage of web portal

Now, we return to the web portal again and see that there is a form with a field for all the information in the actual file, as shown in Figure 13, that is because it is going to be used to create a hash based on this information so that this cannot be modified, or in other words, falsified. Since a simple change in any of the information, would result in a massive change in the resulting hash.

Figure 13. Web UI file form

We also make sure to upload the actual file, and then we click the save button, to upload the file to the S3 bucket, as well as process all the information.

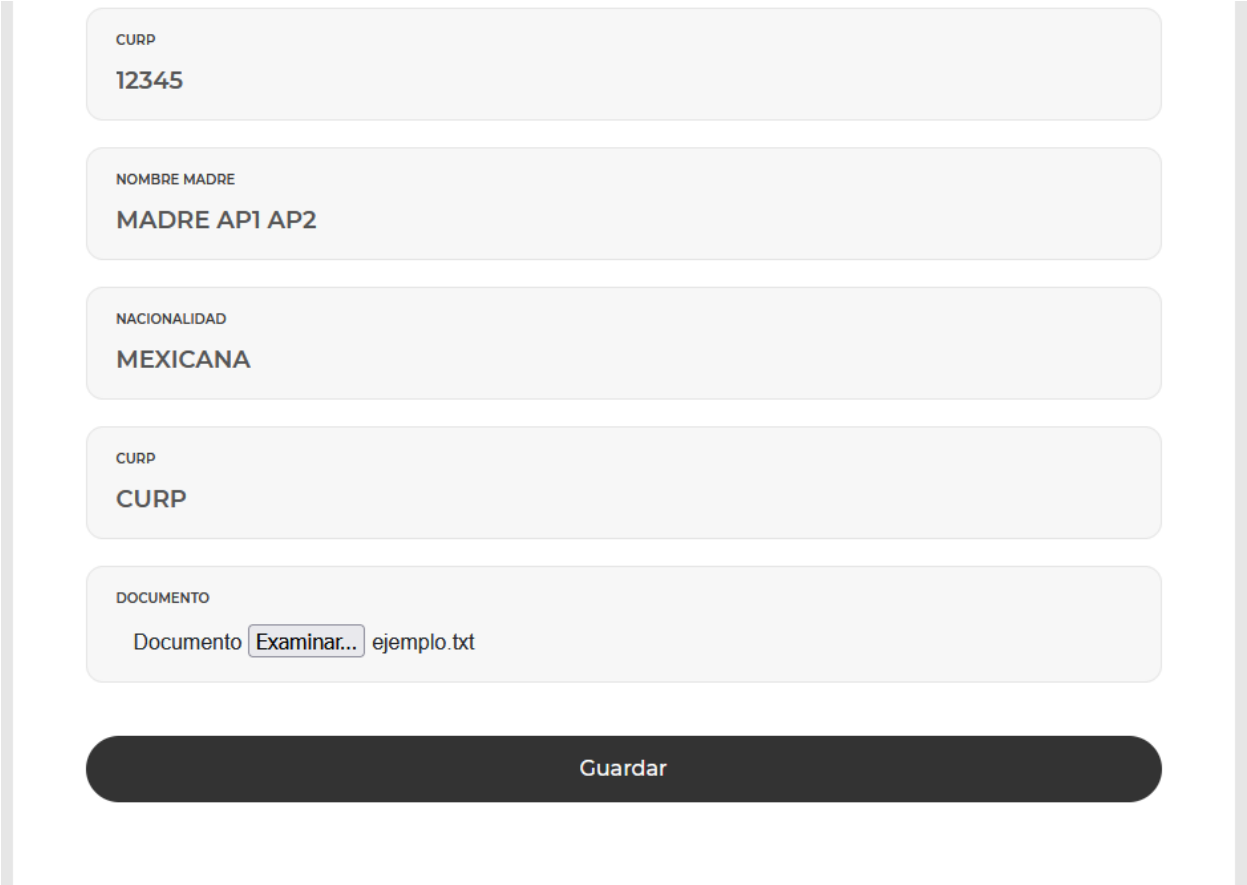


Figure 14. File field and Save button.

4.4 Addition of transactions in the blockchain

In Figure 15, it is shown how there is added a transaction for every file uploaded once we click save in the Web UI, so that after mining for some time, and the right nonce is found, there is created a block in which every transaction made in that time span will be saved.

```
root@ip-172-31-23-129:/home/desarrollos/doc_bc# cat transactions.txt
59ec54a634fdbe739884ff2078fc68898a9aeef48984f9728d3c9f83
8158b78bfb7a0f7d822b4047fdee71055ce417e0ddae653c88a77892
523502603349df12d640a64095c65e002c31602be7cf41072ae50ed0f
93a04076145af989beb3558dd9fdcf9856bc5f237cadf981a37921a4
bcaaela82022136f2caa2c82e6f6ebb0236575c177c0647434ale73a
ffd22f83b5a5bd9b6e2cb82474c8a5a687fca3742ab42f753a58ae5f
c5bcbb4c464977e30708099a8ee717a7217b9d3d76ee299dc290dfd9root@ip-172-31-23-129:/home/desarrollos/doc_bc#
```

Figure 15. Transactions file

The following image shows how the file is uploaded to the AWS S3 bucket, and how the transactions file is updated.

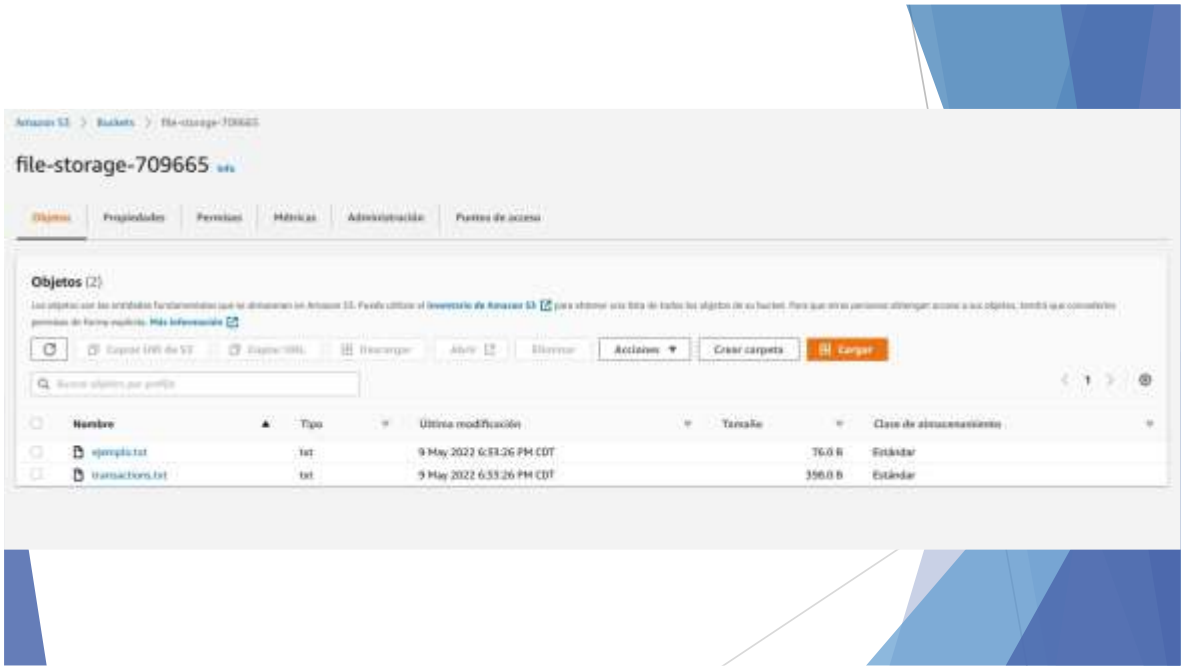


Figure 16. Transactions and example files

4.5 Deployment of Ethereum Nodes

On the other side, there must be local Ethereum nodes with users with a certain balance, which will be the ones that are going to be mining.

To achieve this, we need to install the “geth” library, and start a new Ethereum blockchain with a genesis file, which looks as follows:

```
{
  "config": {
  "chainId": 33,
  "homesteadBlock": 0,
  "eip155Block": 0,
  "eip158Block": 0
},
  "nonce": "0x0000000000000033",
  "timestamp": "0x0",
  "parentHash": "0x0000000000000000000000000000000000000000000000000000000000000000",
  "gasLimit": "0x8000000",
  "difficulty": "0x100",
  "mixhash": "0x0000000000000000000000000000000000000000000000000000000000000000",
  "coinbase": "0x3333333333333333333333333333333333333333",
  "alloc": {
    "9811ebc35d7b06b3fa8dc5809a1f9c52751e1deb": {"balance": "10000000000000000"},
    "001762430ea9c3a26e5749afdb70da5f78ddbb8c": {"balance": "20000000000000000"}
          }
}
```

The main path for the project is:

```
MAIN_DIRECTORY = "/Users/emanuel/Documents/ProE/IDI/blockchain-network"
```

The directory for the node is:

```
NODE_DIRECTORY = $MAIN_DIRECTORY + "/node01"
```

We need to add them to the environment variables for the next commands to work.

To initialize the node blockchain, the next command is executed:

```
geth --datadir NODE_DIRECTORY init MAIN_DIRECTORY/genesis.json
```

The command “geth” is the main instruction for every command related to the blockchain, after we installed the geth library. The flag “--datadir” is used to indicate where the files created after the initialization of the node will be located, and the “init” instruction is used to indicate where the genesis file is located.

Once the node or nodes are initialized, they need to be started with the next command:

```
geth --identity "node01" --authrpc.port 8000 --http.corsdomain "" --datadir
NODE_DIRECTORY --port 30303 --nodiscover --http.api
"db,eth,net,web3,personal,miner,admin" --networkid 1900 --nat "any" --
authrpc.vhosts "" --authrpc.jwtsecret " NODE_DIRECTORY geth/jwtsecret"
```

Here is an explanation of the flags used in the above command:

- identity: The name of the node
- authrpc.port: Listening port for authenticated APIs
- http.corsdomain: Comma separated list of domains from which to accept cross origin requests
- datadir: Data directory for the databases and keystore
- port: Network listening port
- nodiscover: Disables the peer discovery mechanism (manual peer addition)
- http.api: API's offered over the HTTP-RPC interface
- networkid: Explicitly set network id (integer)
- nat: NAT port mapping mechanism (any|none|upnp|pmp|pmp:<IP>|extip:<IP>)
- authrpc.vhosts: Comma separated list of virtual hostnames from which to accept requests (server enforced). Accepts '*' wildcard.
- authrpc.jwtsecret: Path to a JWT secret to use for authenticated RPC endpoints

Once the node is running, the console will show something like this:

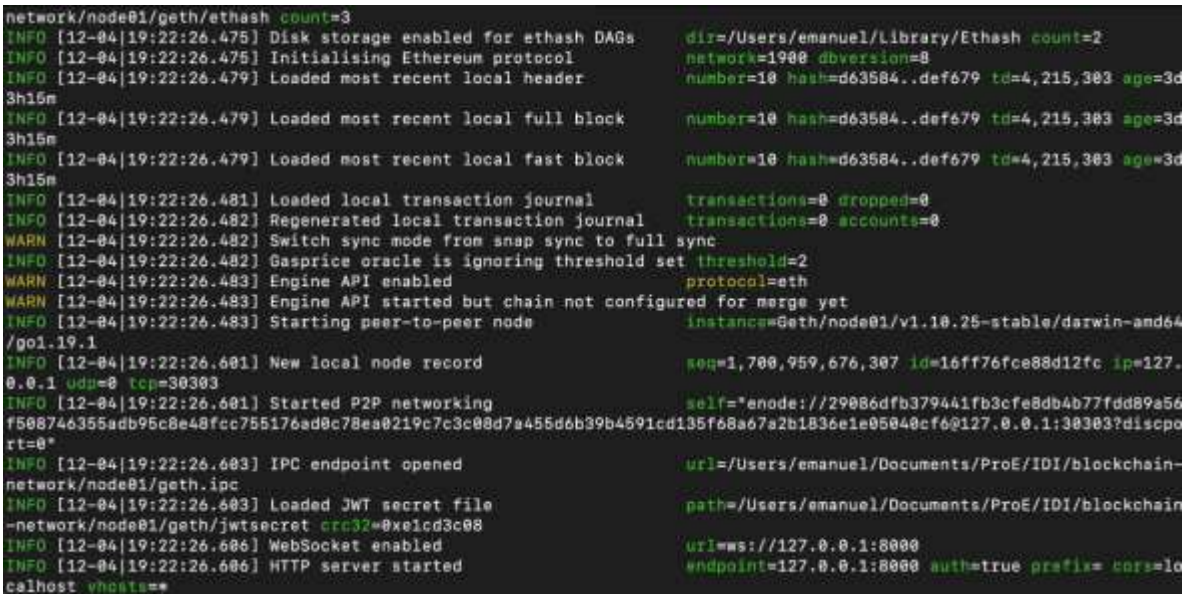


Figure 17. Node Initialized

When our node is up, it is possible to connect via API to it, it can be achieved through the command line, or Python, as shown below, for this, a URL provided every time the node is activated is needed, or it can be using the file with extension ".ipc", which is also generated every time our node is up and running, this last one is the method used for this project.

So, in the directory created for our node01, we execute the following command:

```
geth attach geth.ipc
```

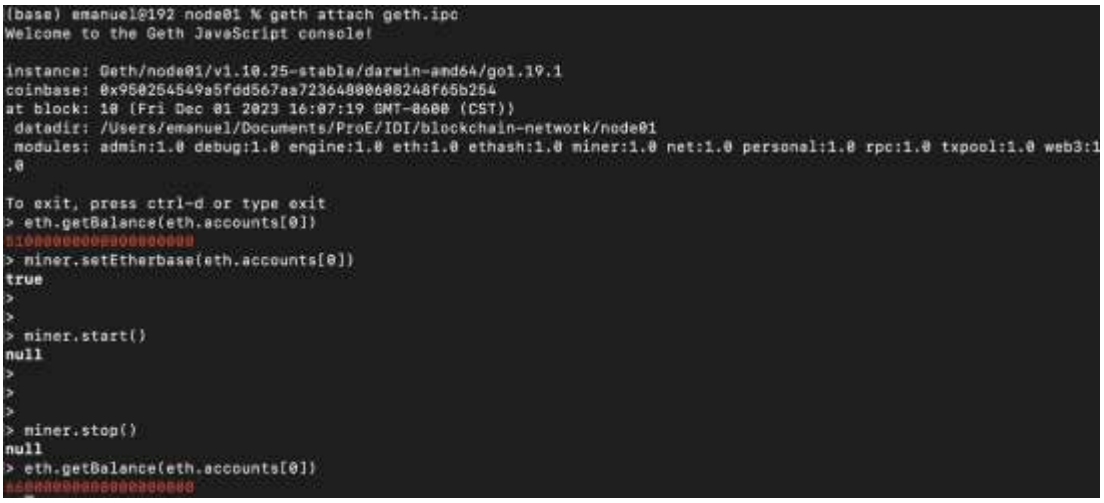


Figure 18. Node command line

The information of the main node, needed for every other node to connect to it, is shown using the command “admin.nodeInfo”, as shown in Figure 19:

Figure 19. Node Information

5. Conclusions

When there is a resource as powerful as blockchain, there can be endless ways to take advantage of it, even if it is different than what it was thought at the beginning, but very useful in other applications, nevertheless.

That is the case for this project, because making use of the advantages of a blockchain, such as its veracity, falsification difficulty, and the inherent relation between the last blocks with the new ones, makes a solid robust system capable of managing important documents with a very low risk of being falsified, and a very easy and efficient way to access those files.

With the necessary adjustments, and the implementation of the techniques acquired in the master's program, such as the AWS cloud, and other ones explored and learned more specifically for the project, like the creation and management of a blockchain, this work done brings an improvement to the traditional saving and storing of official files.

Since the same nature of a blockchain does not leave room for the addition or removal of transactions, in this case, files, not even falsified ones, because if there were to be a failure or inconsistency in a node, there would be multiple nodes to contradict that failure, and at the end of the day, the majority will be what it is considered true.

This project, if were to be implemented on a big scale, and made official, I think that could be something that the government of multiple countries could start to consider for the management of their documents, due to its veracity, efficiency, and practicality.

5.1 Future work

What I consider to be an improvement for this project is, first, I think that there could be a wider range of files that the system can manage, and even think of not making it exclusive to manage government files.

Besides that, I think that a powerful improvement could be the use of character recognition to fill every field of information in the documents, since at the moment, it has to be filled in the form through the web UI, so, it would be a considerable upgrade to make it possible to upload the file and making it part of the blockchain without having to fill every field of the form.

Also, and may be a more complex topic, but if it is something that gets massive enough, there could be the incorporation of multiple kinds of blockchains in the project. Since every blockchain uses a different protocol, it could get difficult to mix them, but I see it as a possibility, to let the user decide which kind of blockchain to use when the file is uploaded to the system.

References

[1] M. Cabalero, (2019),” Bitcoin, blockchain y tokenización para inquietos, 1st ed.”, Madrid: Bubok Publishing.

[2] A. Lewis, (2018), ”The basics of bitcoins and blockchains, 1st ed.”, Coral Gables: Mango Publishing.

[3] Damilola Lawrence, (2023), “Redes blockchain descentralizadas vs distribuidas, ¿cuál es mejor?”, Online: https://www.cryptopolitan.com/es/blockchain-descentralizado-vs-distribuido/

[4] Jeffrey Hoffstein, Jill Pipher, Joseph H. Silverman, (2014), ”An Introduction to Mathematical Cryptography, 2nd ed.”, Springer.

[5] Jean-Philippe Aumasson, (2018),” Serious Cryptography: A Practical Introduction, 1st ed.”, San Francisco: No starch press.

[6] S.N., (S.F), “Tolerancia a las fallas bizantinas, una guía rápida”, Online: https://ciberseguridad.com/guias/nuevas-tecnologias/criptomoneda/tolerancia-fallas-bizantinas/

[7] Nigel P. Smart, (2016), ”Cryptography made simple, 1st ed”, Switzerland: Springer International Publishing AG Switzerland.

[8] “What is blockchain and how does it work? | Synopsys.” https://www.synopsys.com/glossary/what-is-blockchain.html

[9] M. A. Z. Bin Idrus, F. D. A. Rahman, O. O. Khalifa, and N. M. Yusoff, "Blockchain-based Security for Cloud Data Storage," 2023 IEEE 9th International Conference on Smart Instrumentation, Measurement, and Applications (ICSIMA), Kuala Lumpur, Malaysia, 2023, pp. 73-77.

[10] Z. Fu, X. Cao, J. Wang, and X. Sun, "Secure Storage of Data in Cloud Computing," *2014 Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, Kitakyushu, Japan, 2014, pp. 783-786, doi: 10.1109/IIH-MSP.2014.199.

[11] J. J. N. Hermosillo, O. P. Alvarez, E. F. Ramirez, and P. Salazar-Linares, "Evolution of autograph signature to advanced electronic signature in smart cities environment," *2018 IEEE International Smart Cities Conference (ISC2)*, Kansas City, MO, USA, 2018, pp. 1-5, doi: 10.1109/ISC2.2018.8656666.

[12] SN, (SF), “Redes blockchain descentralizadas vs distribuidas, ¿cuál es mejor?”, Online: https://www.cryptopolitan.com/es/blockchain-descentralizado-vs-distribuido/

[13] SN, (SF), “What is Ethereum?”, Online: https://ethereum.org/en/what-is-ethereum/

[14] Abhishek Bhati, (June 15, 2022), “Attacks on blockchain”, Online: https://wesecureapp.com/blog/attacks-on-blockchain/

[14] SN, (February 29, 2024), “Attacks on blockchain”, Online: https://webisoft.com/articles/how-to-use-blockchain-api/

[15] C. E. Shannon, (September 1, 1945), “Mathematical Theory of Cryptography”, USA.

[16] Kerman Kohli, (June 18, 2019), “Learning Cryptography, Part 1: Finite Fields”, Online: https://medium.com/loopring-protocol/learning-cryptography-finite-fields-ced3574a53fe

[17] N. Kshetri and E. Loukoianova, (Jan.-Feb. 2019) "Blockchain Adoption in Supply Chain Networks in Asia," in *IT Professional*, vol. 21, no. 1, pp. 11-15.

[18] P. Ndayizigamiye and S. Dube, (2019) "Potential Adoption of Blockchain Technology to Enhance Transparency and Accountability in the Public Healthcare System in South Africa," *2019 International Multidisciplinary Information Technology and Engineering Conference (IMITEC)*, Vanderbijlpark, South Africa.

[19] A. L. Franzoni, C. Cárdenas and A. Almazan, (2019), "Using Blockchain to Store Teachers' Certification in Basic Education in Mexico," *2019 IEEE 19th International Conference on Advanced Learning Technologies (ICALT)*, Maceio, Brazil, pp. 217-218

[20] F. M. Santos López, J. M. Portella Delgado, E. G. Santos De la Cruz and E. L. Cáceres, (2021), "Performance-based Software Architecture Design and Blockchain as a Service for Peruvian E-government," *2021 IEEE 12th International Conference on Software Engineering and Service Science (ICSESS)*, Beijing, China.

[21] S.N., (Feb 2022), “What Are Token Standards? An Overview”, Online: https://crypto.com/university/what-are-token-standards

[22] Somer Anderson, (May 2024), “What is Ethereum and how does it work?”,

Online: https://www.investopedia.com/terms/e/ethereum.asp

[23] John Williams, (August 2022), “The 12 best blockchain Node Providers in Web3”,

Online: https://www.alchemy.com/overviews/blockchain-node-providers