# **Documentazione Progetto Block Chain**

Una blockchain è un tipo speciale di database. Ci si riferisce alle blockchain con il termine “tecnologia a registro distribuito” (o DLT) ma in molti casi, si riferiscono alla stessa cosa.

Una blockchain ha delle proprietà uniche, ci sono regole che governano il modo in cui i dati possono essere

aggiunti, e una volta archiviati, è virtualmente impossibile modificarli o cancellarli.

I dati vengono aggiunti nel corso del tempo in strutture chiamate “*blocchi”*.

Ciascun blocco è costruito sopra il precedente e include un pezzo di informazione che lo collega ad esso. Esaminando il blocco più recente, possiamo verificare che sia stato creato dopo il suo precedente. Quindi, se continuiamo fino in fondo alla “catena di blocchi” (da qui il nome *blockchain*) raggiungeremo il primo blocco, denominato *“blocco genesi”.*

La blockchain utilizza le funzioni di hash per collegare i blocchi tra di loro : genera un hash utilizzando i dati inseriti in ogni riga per compilare la riga successiva. Questo comporta che se cambiassimo i dati del primo input otterremmo una combinazione diversa in tutte le altre caselle.

In modo schematico questo mostra come l'hashing sia la colla che tiene insieme i blocchi. Consiste nel prendere dati di qualsiasi dimensione e passarli attraverso una funzione matematica per produrre un output (hash) che ha sempre la stessa lunghezza.

Le hash usate nelle blockchain sono interessanti, in quanto le probabilità di trovare dati differenti che producono lo stesso output sono quasi nulle, e qualsiasi modifica, anche minima, dei dati di input porterà a un output completamente diverso. Un algoritmo hash funziona in una sola direzione: ciò significa che da qualsiasi contenuto possiamo generare il suo hash (la sua "impronta digitale") ma da un hash non c'è modo di generare il contenuto ad esso associato**.**

**L'acronimo SHA-256 si riferisce alla funzione hash che è ampiamente utilizzata per il funzionamento di molte criptovalute (anche di Bitcoin) in quanto offre un elevato livello di sicurezza, che la rende perfetta**

**per proteggere e codificare in modo sicuro le proprie informazioni ed in oltre è resistente alle collisioni.**

Questa funzione hash crittografica genera un'impronta digitale di un file nota come *“checksum”* , cioè un dato di piccole dimensioni derivato da un blocco di dati più grande allo scopo di rilevare gli errori che possono essere stati introdotti durante la trasmissione o l'archiviazione.

All'interno della [rete blockchain](https://academy.bit2me.com/it/cos%27%C3%A8-la-blockchain-blockchain/) tutti [i nodi](https://academy.bit2me.com/it/assemblare-il-nodo-bitcoin-in-meno-di-10-minuti/) avrebbero una copia dell'hash di 64 caratteri che rappresenta l'informazione che, ad esempio, costituisce un intero blocco. Una volta che l'informazione è stata convalidata dalla rete (risulta già registrata nella catena), qualsiasi manipolazione di quell'informazione, che cerchi di modificare qualche carattere dell'hash convalidato, verrebbe rilevata immediatamente e scartata.

Immagine che contiene testo

Descrizione generata automaticamente

Segue una possibile implementazione in linguaggio C++ di una blockchain ;

Le prime righe del programma prevedono l’inclusione di alcune librerie indispensabili per il funzionamento del codice . Dato che verranno utilizzati molti oggetti *string* e diversi *cout*, *endl* ecc. si include anche il *namespace standard* per evitare innumerevoli utilizzi dell’operatore di risoluzione di scope *“ std:: ”* .

#include <iostream>

#include <vector>

#include <sstream>

#include <ctime>

using namespace std;

Una blockchain nasce con l’attivazione del block genesis, al quale si concatenano una serie di blocchi che contengono dati e ogni blocco contiene una rappresentazione crittografica del blocco precedente, il che significa che diventa molto difficile modificare il contenuto di qualsiasi blocco senza poi dover cambiare ogni blocco successivo;

Immagine che contiene testo, screenshot, monitor, schermo

Descrizione generata automaticamente