



PineSU -
Workflow

Paolo Speziali

Operazioni

Creazione

Registrazione

Progettazione

Apprendimento

Implementa-
zione

Resoconto

Fonti

PineSU - Workflow

Sviluppo di un'applicazione distribuita su Blockchain
Ethereum

Paolo Speziali

Università degli Studi di Perugia - Dipartimento di Ingegneria



A.A. 2020/2021



Operazioni eseguibili

PineSU -
Workflow

Paolo Speziali

Operazioni

Creazione

Registrazione

Progettazione

Apprendimento

Implementa-
zione

Resoconto

Fonti

Nell'applicativo è possibile eseguire le seguenti operazioni:

- 1 Creazione di una Storage Unit (SU)
- 2 Registrazione di una o più SU nella blockchain Ethereum
- 3 Esportazione di sottoinsiemi di file da una SU
- 4 Controllo d'integrità su una SU
- 5 Controllo d'integrità di singoli file esportati da altre SU

Oltre a poter eseguire tutti i comandi di Git. Vedremo ora questi comandi uno per uno utilizzando una directory campione da trasformare in Storage Unit.



Directory campione

PineSU -
Workflow

Paolo Speziati

Operazioni

Creazione

Registrazione

Progettazione

Apprendimento

Implementa-
zione

Resoconto

Fonti

Utilizzeremo come directory campione per mostrare l'evoluzione della Storage Unit una cartella chiamata *sample* con la seguente struttura (dove *first* e *second* sono sottodirectory di *sample* e *third* è sottodirectory di *second*):

```
sample
sample/first
sample/first/astar.js
sample/first/graph.js
sample/graphCreator.js
sample/second
sample/second/priorityQueue.js
sample/second/third
sample/second/third/main.js
sample/second/third/vertex.js
```



Creazione di una SU

PineSU -
Workflow

Paolo Speziali

Operazioni

Creazione

Registrazione

Progettazione

Apprendimento

Implementa-
zione

Resoconto

Fonti

La creazione di una Storage Unit è in realtà un'operazione che comprende sia la trasformazione in una Git Repository della directory, sia il calcolo degli hash che serviranno poi per registrare la nostra SU nella blockchain.

Le informazioni della nostra SU sono conservate nel file **.pinesu.json** nella root della directory, la presenza di questo file indica al programma che la directory è già una SU.

La creazione potrebbe eventualmente venire suddivisa in due azioni diverse, ma a quel punto la prima fase si limiterebbe ad eseguire un comando "git init" sulla directory, comando che potrebbe non essere necessario.



Creazione di una SU

PineSU -
Workflow

Paolo Speciali

Operazioni

Creazione

Registrazione

Progettazione

Apprendimento

Implementa-
zione

Resoconto

Fonti

Tuttavia, per come il programma è ora strutturato, la creazione della Storage Unit la rende pronta per essere registrata (eventualmente insieme ad altre) nella blockchain.

Andiamo quindi a vedere quali sono le operazioni che avvengono in questa fase e come la nostra directory *sample* cambia.



Creazione di una SU

PineSU -
Workflow

Paolo Spezioli

Operazioni

Creazione

Registrazione

Progettazione

Apprendimento

Implementa-
zione

Resoconto

Fonti

- Controlla se nella directory è già presente **.pinesu.json**, in caso negativo continua;
- Permette all'utente di selezionare dei file da ignorare da tutto il processo, ciò produce un file **.gitignore**;
- Aggiunge tutti i file della directory nella Git Staging Area e crea un commit fantoccio;
- Recupera la lista dei file dal commit;
- Calcola gli hash di tutti i file (non directory);
- Calcola gli hash delle directory creando dei Merkle Tree con gli hash del loro contenuto e usa l'hash della radice come hash della directory, ciò avviene tramite un approccio bottom-up: calcola prima gli hash delle directory senza file al loro interno;



Creazione di una SU

PineSU -
Workflow

Paolo Spezioli

Operazioni

Creazione

Registrazione

Progettazione

Apprendimento

Implementa-
zione

Resoconto

Fonti

- Calcola l'hash della SU creando un Merkle Tree con tutti gli hash dei file e delle directory in essa contenute e ne prende la radice;
- Viene generato il file **.pinesu.json** con al suo interno nome, descrizione, visibilità, data, hash dell'utente, hash della SU, lista dei file e delle sottodirectory con hash associati;
- Aggiunge tutti i file della directory nella Git Staging Area e crea un commit fantoccio;
- Aggiungo il percorso di questa SU con il suo hash alla lista della SU generate dall'utente (memorizzata nella cartella d'installazione del programma);
- Viene annullato il commit fantoccio e creato uno effettivo con informazioni consistenti e aggiunto **.pinesu.json**;



Creazione di una SU - Risultato

PineSU -
Workflow

Paolo Speciali

Operazioni

Creazione

Registrazione

Progettazione

Apprendimento

Implementa-
zione

Risultato

Fonti

L'operazione di creazione ha prodotto nella cartella *sample*, ovviamente, una sottocartella *.git* e il file **.pinesu.json**.

L'ordine di calcolo degli hash è stato questo:

- 1 sample/first/astar.js
- 2 sample/first/graph.js
- 3 sample/graphCreator.js
- 4 sample/second/priorityQueue.js
- 5 sample/second/third/main.js
- 6 sample/second/third/vertex.js



Creazione di una SU - Risultato

PineSU -
Workflow

Paolo Speziati

Operazioni

Creazione

Registrazione

Progettazione

Apprendimento

Implementa-
zione

Resoconto

Fonti

- 7 sample/first (tramite MT dei due file da lui contenuti)
- 8 sample/second/third (tramite MT dei due file da lui contenuti)
- 9 sample/second (tramite MT del file e della sottodirectory *third* da lui contenuti)
- 10 sample (tramite MT degli hash dei tutti i file e le directory appena visti)

Si può quindi facilmente osservare come questa tecnica di calcolo di hash bottom-up viene messa in atto calcolando prima i file effettivi, poi le directory senza sottodirectory e infine tutto il resto delle directory.



Registrazione di una o più SU nella blockchain

PineSU -
Workflow

Paolo Spezioli

Operazioni

Creazione

Registrazione

Progettazione

Apprendimento

Implementa-
zione

Resoconto

Fonti

In questa fase all'utente viene data la libertà di scegliere se registrare con un unico hash una o più SU insieme, poiché il workflow è praticamente identico andrò a separare i punti esclusivi ad una registrazione “multipla” scrivendoli in color oliva:

- Mostra la selezione multipla della SU registrate dall'utente per permettere di registrarne una sola o più in una volta;
- Per ogni **Storage Unit selezionata** creo nella root della SU il file **.registration.json** che contiene l'hash della directory, l'hash che verrà salvato nella blockchain (stesso valore se registriamo una sola SU, **valore della root del MT di tutti gli hash delle SU selezionate se la registrazione è multipla**)



Excursus sulle Applicazioni Distribuite

PineSU -
Workflow

Paolo Speziali

Operazioni

Creazione

Registrazione

Progettazione

Apprendimento

Implementa-
zione

Resoconto

Fonti



Su tale rete è possibile mettere a disposizione delle applicazioni che chiunque può utilizzare dietro pagamento di una piccola commissione. Queste applicazioni sono realizzabili dagli sviluppatori interessati tramite vari framework e suite di applicativi, il più celebre è **Truffle**.



Excursus su Git

PineSU -
Workflow

Paolo Speziati

Operazioni

Creazione

Registrazione

Progettazione

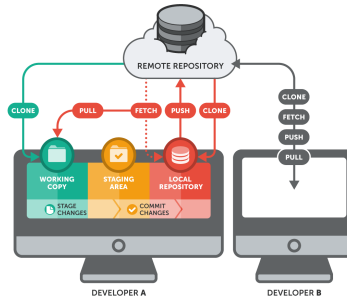
Apprendimento

Implementa-
zione

Resoconto

Fonti

Git è un software che permette in maniera semplice di gestire insieme di file tramite un sistema di controllo di versione, una grande risorsa per gli sviluppatori che devono contribuire in maniera condivisa ad uno stesso progetto o che devono tenere sotto controllo i vari cambiamenti che sono stati apportati ai vari file e documenti.





Excursus sulle funzioni di hashing

PineSU -
Workflow

Paolo Speziati

Operazioni

Creazione

Registrazione

Progettazione

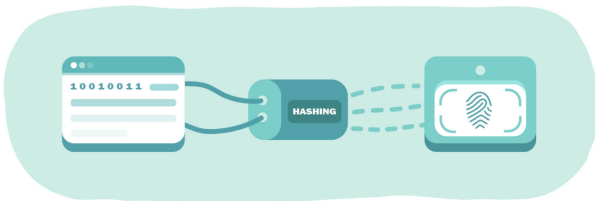
Apprendimento

Implementa-
zione

Resoconto

Fonti

Le funzioni di hashing sono funzioni non invertibili che permettono di associare in maniera univoca (o quasi) stringhe di caratteri (e quindi anche documenti di varia natura tradotti in stringhe) a delle stringhe alfanumeriche di lunghezza fissa.





Fase 3: Progettazione - Prima stesura

PineSU -
Workflow

Paolo Speziali

Operazioni

Creazione

Registrazione

Progettazione

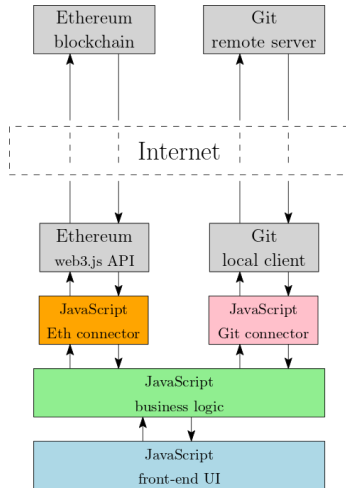
Apprendimento

Implementa-
zione

Resoconto

Fonti

La stesura iniziale è stata svolta dal professore che mi ha quindi fornito una linea guida da cui poter prendere spunto per poter realizzare il progetto dopo aver concordato sulle sue funzionalità, tale architettura è stata tuttavia modificata abbastanza in quanto le tecnologie utilizzate sono ancora troppo sperimentali e non offrono gli strumenti per potere essere adattati ad una struttura del genere.

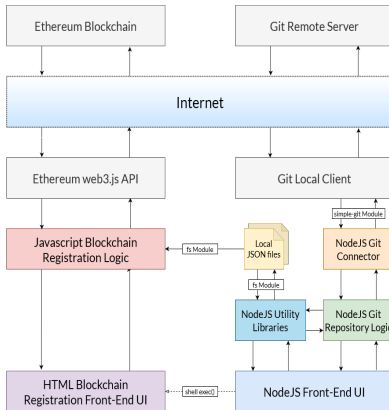




Fase 3: Progettazione - Seconda stesura

PineSU - Workflow

Progettazione



Il Front-End in NodeJS che comunica con una parte logica è stato mantenuto, sono stati però introdotti alcuni moduli Utility che vengono utilizzati da entrambe le componenti.

La connessione alla Blockchain avviene attraverso l'avvio di un Web Server apposito e l'apertura di una scheda del browser, l'interazione da parte dell'utente deve avvenire tramite l'add-on **Metamask**.



Fase 4: Apprendimento

PineSU -
Workflow

Paolo Spezioli

Operazioni

Creazione

Registrazione

Progettazione

Apprendimento

Implementa-
zione

Resoconto

Fonti

In concomitanza con la fase di progettazione è stato necessario l'apprendimento di alcune tecnologie in modo da poter imparare le loro modalità di utilizzo e poter effettuare la seconda stesura dello schema progettuale.

Oltre ai già citati Truffle Suite e Metamask, abbastanza intuitivi nell'utilizzo, le tecnologie che seguono sono quelle che hanno occupato la maggior parte di questa fase.



METAMASK



Fase 4: Apprendimento - Solidity

PineSU -
Workflow

Paolo Speziali

Operazioni

Creazione

Registrazione

Progettazione

Apprendimento

Implementa-
zione

Resoconto

Fonti

Per l'apprendimento della sintassi e delle peculiarità del linguaggio da utilizzare per scrivere DAPP per la Blockchain Ethereum mi sono avvalso della documentazione ufficiale e del tutorial interattivo **CryptoZombies**.





Fase 4: Apprendimento - Javascript Asynchronous Programming

PineSU -
Workflow

Paolo Speciali

Operazioni

Creazione

Registrazione

Progettazione

Apprendimento

Implementa-
zione

Resoconto

Fonti

Async () => { Await }

L'utilizzo di alcuni moduli all'interno dell'applicazione ha richiesto che io spendessi diverso tempo ad imparare le tecniche di programmazione asincrona di Javascript in quanto lo scorretto utilizzo delle keyword `async` e `await` sono state fonte di svariati problemi nella prima fase della stesura del codice.



Fase 4: Apprendimento - Modulo Merkle Tree

PineSU -
Workflow

Paolo Spezioli

Operazioni

Creazione

Registrazione

Progettazione

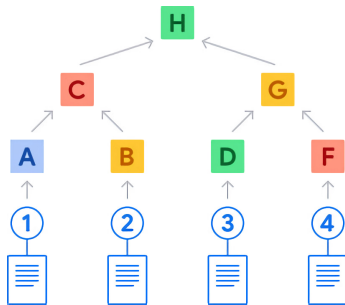
Apprendimento

Implementa-
zione

Resoconto

Fonti

La necessità di dover calcolare una singola stringa Hash per una moltitudine di file ha portato il professore a proporre l'utilizzo di questa struttura, è stato quindi necessario da parte mia non solo comprenderne bene il funzionamento ma anche essere in grado di poter utilizzare al meglio i moduli che fornivano metodi per lavorare con questi particolari alberi.





Fase 4: Apprendimento - Modulo InquirerJS

PineSU -
Workflow

Paolo Speciali

Operazioni

Creazione

Registrazione

Progettazione

Apprendimento

Implementa-
zione

Resoconto

Fonti



Anziché realizzare un Front-End dotato di GUI, ho preferito optare per una interfaccia testuale. Tuttavia, non volendo rinunciare all'immediatezza e la semplicità che un approccio grafico e user-friendly poteva portare al progetto ho imparato ad utilizzare il modulo di InquirerJS, il quale consente di realizzare menù a scelta singola, multipla e libera in tutta comodità.



Fase 5: Implementazione

PineSU -
Workflow

Paolo Speciali

Operazioni

Creazione

Registrazione

Progettazione

Apprendimento

Implementa-
zione

Resoconto

Fonti

La fase di implementazione è divisibile in tre macro-sezioni:

- 1** Creazione delle librerie di utility e dei moduli di interfacciamento con Git
- 2** Creazione della CLI e del flow di interazione con le librerie e la logica di Git
- 3** Creazione del modulo di interrogazione e registrazione per la Blockchain



Creazione delle librerie di utility e dei moduli di interfacciamento con Git

PineSU -
Workflow

Paolo Speciali

Operazioni

Creazione

Registrazione

Progettazione

Apprendimento

Implementa-
zione

Riscontro

Fonti

La prima stesura di codice è avvenuta nella creazione della classe “connettore” per Git con il modulo “simple-git” e del relativo modulo Logic il quale richiama le sue funzioni a seconda della necessità della CLI.



Creazione delle librerie di utility e dei moduli di interfacciamento con Git

PineSU -
Workflow

Paolo Speziali

Operazioni

Creazione

Registrazione

Progettazione

Apprendimento

Implementa-
zione

Riscontro

Fonti

In concomitanza ho scritto i moduli del package “lib”:

- **files:** Lettura e scrittura di file JSON in cui conservare le informazioni riguardanti la Storage Unit o l'utente che sta utilizzando l'applicativo;
- **inquirer:** Contiene tutte le scelte che vengono poi presentate all'utente nella CLI;
- **treelist:** Si occupa di effettuare tutte le operazioni riguardanti l'hashing di file, l'assegnazione di hash alle subdirectories e la creazione e gestione di Merkle Tree.



Creazione della CLI e del flow di interazione con le librerie e la logica di Git

PineSU -
Workflow

Paolo Speciali

Operazioni

Creazione

Registrazione

Progettazione

Apprendimento

Implementa-
zione

Resoconto

Fonti

Ho proseguito andando a creare l'effettivo workflow del programma richiamando le scelte dal modulo inquirer e funzioni differenti in base alle selezioni dell'utente.



Creazione della CLI e del flow di interazione con le librerie e la logica di Git - Workflow

PineSU -
Workflow

Paolo Speziari

Operazioni

Creazione

Registrazione

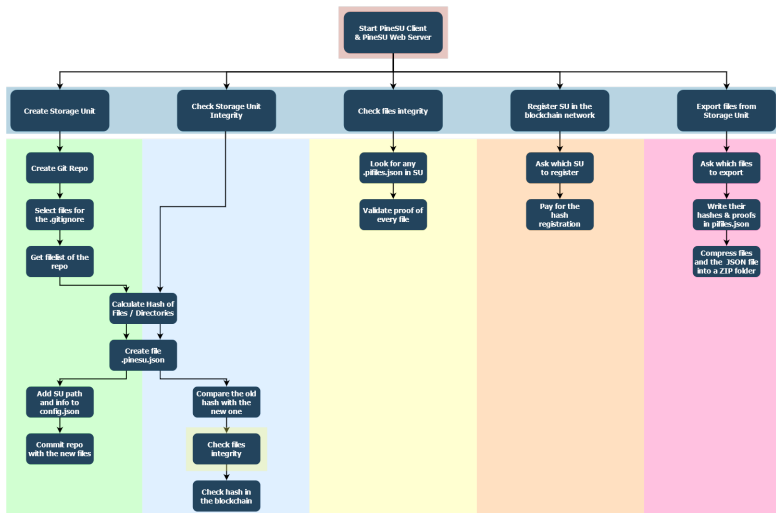
Progettazione

Apprendimento

Implementa-
zione

Resoconto

Fonti





Creazione del modulo di interrogazione e registrazione per la Blockchain

PineSU -
Workflow

Paolo Spezioli

Operazioni

Creazione

Registrazione

Progettazione

Apprendimento

Implementa-
zione

Resoconto

Fonti

The screenshot shows the MetaMask extension window titled 'Estensione: (MetaMask) - MetaMask Notificat...'. It displays a transaction confirmation for 'Account 1' (0x7216...4d99) to 'Localhost 7545'. The transaction is for 'INTERAZIONE CONTRATTO' at 'http://localhost:3000'. The gas fee is 0.00272 ETH, and the total amount is 0.00272 ETH. The interface includes tabs for 'DETAILS' and 'DATA', and buttons for 'Annulla' and 'Conferma'.

| DETAILS | |
|--------------------------------------|------------|
| GAS FEE | 0.00272 |
| Tasso di conversione non disponibile | |
| Prezzo del Gas (GWEI) | Gas Limite |
| 20 | 136020 |
| AMOUNT + GAS FEE | |
| TOTAL | 0.00272 |
| Tasso di conversione non disponibile | |

Figura: Transazione per registrare un hash nella blockchain

L'ultima parte della fase di implementazione è stata la realizzazione del Web Server locale che permette all'utente di registrare gli hash delle proprie Storage Unit nella blockchain, il risultato è stato ottenuto con una versione pesantemente modificata dell'applicazione *sample* fornita dal sito della suite Truffle.



Resoconto

PineSU -
Workflow

Paolo Speziali

Operazioni

Creazione

Registrazione

Progettazione

Apprendimento

Implementa-
zione

Resoconto

Fonti

- Data di inizio tirocinio: 15/2/2021
- Data di fine tirocinio: 28/5/2021
- Totale Ore: 150 (25 ore · 6 CFU)
- Professore Tutor: Luca Grilli
- Studente Tirocinante: Paolo Speziali
- Anno Accademico: 2020/2021



Fonti

PineSU -
Workflow

Paolo Speciali

Operazioni

Creazione

Registrazione

Progettazione

Apprendimento

Implementa-
zione

Resoconto

Fonti

- Strumenti Ethereum per sviluppatori
- Tutorial Truffle DAPPs - Pet Shop
- Build a JavaScript CLI with Node.js
- Tutorial di Mozilla su async / await
- Immagini reperite dai siti ufficiali degli strumenti eccetto per alcune scaricate da queste pagine web:
 - Funzioni di Hashing
 - Concorso pubblico
 - Ethereum Blockchain
 - Git repository
 - async / await
 - Merkle Tree
- Strumento di upscaling delle immagini