

Università degli Studi di Padova

DIPARTIMENTO DI MATEMATICA “TULLIO LEVI-CIVITA”

CORSO DI LAUREA IN INFORMATICA



Titolo della tesi

Tesi di Laurea Triennale

Relatore

Prof.ssa Ombretta Gaggi

Laureando

Gabriel Rovesti

ANNO ACCADEMICO 2022-2023

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

— Oscar Wilde

Dedicato a ...

Sommario

Il presente documento descrive il lavoro svolto durante il periodo di stage, della durata di circa trecento ore, dal laureando Pinco Pallino presso l'azienda Azienda S.p.A. Gli obiettivi da raggiungere erano molteplici. In primo luogo era richiesto lo sviluppo di ... In secondo luogo era richiesta l'implementazione di un ... Tale framework permette di registrare gli eventi di un controllore programmabile, quali segnali applicati Terzo ed ultimo obiettivo era l'integrazione ...

“Life is really simple, but we insist on making it complicated”

— Confucius

Ringraziamenti

Innanzitutto, vorrei esprimere la mia gratitudine al Prof. Ombretta Gaggi, relatore della mia tesi, per l'aiuto e il sostegno fornitomi durante la stesura del lavoro.

Desidero ringraziare con affetto i miei genitori per il sostegno, il grande aiuto e per essermi stati vicini in ogni momento durante gli anni di studio.

Ho desiderio di ringraziare poi i miei amici per tutti i bellissimi anni passati insieme e le mille avventure vissute.

Padova, Aprile 2023

Gabriel Rovesti

Indice

1	Introduzione	1
1.1	L'azienda	1
1.2	Introduzione al progetto	1
1.3	Way of Working e strumenti	1
1.4	Organizzazione del testo	1
2	Tecnologie di interesse	3
2.1	Blockchain: concetti base	3
2.1.1	Introduzione	3
2.1.2	Blocco	3
2.1.3	Transazione	4
2.1.4	Wallet	4
2.1.5	Mining	5
2.1.6	Algoritmi di consenso	5
2.1.7	Tipi	6
2.2	Blockchain: concetti avanzati	6
2.2.1	Token	6
2.2.2	Tokenizzazione	6
2.2.3	Smart contract	6
2.2.4	Scalabilità	6
2.3	Self-Sovereign Identity	6
2.3.1	Protocolli	7
2.3.2	Casi d'uso reali	7
2.4	Zero Knowledge Proof	7
2.4.1	Tipi	7
2.4.2	Casi d'uso reali	7
3	Descrizione dello stage	9
3.1	Introduzione al progetto	9
3.2	Analisi preventiva dei rischi	9
3.3	Requisiti e obiettivi	9
3.4	Pianificazione	9
4	Analisi dei requisiti	11
4.1	Casi d'uso	11
4.2	Tracciamento dei requisiti	12
5	Progettazione e codifica	15
5.1	Tecnologie utilizzate	15
5.1.1	Codifica	15
5.1.2	Versionamento	15

5.2	Ciclo di vita del software	15
5.3	Progettazione	15
5.4	Design Pattern Utilizzati	15
5.5	Codifica	15
6	Verifica e validazione	17
7	Conclusioni	19
7.1	Consuntivo finale	19
7.2	Raggiungimento degli obiettivi	19
7.3	Conoscenze acquisite	19
7.4	Valutazione personale	19
A	Appendice A	21
	Bibliografia	25

Elenco delle figure

4.1	Use Case - UC0: Scenario principale	11
-----	---	----

Elenco delle tabelle

4.1	Tabella del tracciamento dei requisiti funzionali	13
4.2	Tabella del tracciamento dei requisiti qualitativi	13
4.3	Tabella del tracciamento dei requisiti di vincolo	13

Capitolo 1

Introduzione

Introduzione al contesto applicativo.

Esempio di utilizzo di un termine nel glossario [Application Program Interface \(API\)](#).

Esempio di citazione in linea *Manifesto Agile*. URL: <http://agilemanifesto.org/iso/it/>.

Esempio di citazione nel pie' di pagina citazione¹

1.1 L'azienda

Sync Lab è un'azienda italiana che si occupa di sviluppo software nata nel 2002 con sede principale a Napoli, rapidamente affermata come System Integrator secondo un processo di ricerca e maturazione di competenze tecnologiche sempre diverse.

L'azienda opera in diversi settori, tra cui quello del mobile, della cybersecurity, del web e della videosorveglianza. L'azienda è in continua crescita e attualmente conta oltre 250 dipendenti, dislocati nelle altre sedi presenti: Roma, Milano, Padova e Verona.

1.2 Introduzione al progetto

Introduzione all'idea dello stage.

Riguardo la stesura del testo, relativamente al documento sono state adottate le seguenti convenzioni tipografiche:

- gli acronimi, le abbreviazioni e i termini ambigui o di uso non comune menzionati vengono definiti nel glossario, situato alla fine del presente documento;
- per la prima occorrenza dei termini riportati nel glossario viene utilizzata la seguente nomenclatura: *parola*^[g];
- i termini in lingua straniera o facenti parti del gergo tecnico sono evidenziati con il carattere *corsivo*.

1.3 Way of Working e strumenti

1.4 Organizzazione del testo

Il secondo capitolo descrive ...

[Il terzo capitolo](#) approfondisce ...

[Il quarto capitolo](#) approfondisce ...

¹Daniel T. Jones James P. Womack. *Lean Thinking, Second Edition*. Simon & Schuster, Inc., 2010.

[Il quinto capitolo](#) approfondisce ...

[Il sesto capitolo](#) approfondisce ...

[Nel settimo capitolo](#) describe ...

Capitolo 2

Tecnologie di interesse

In questa sezione viene presentata una panoramica di base delle tecnologie oggetto del mio tirocinio, al fine di descrivere in modo chiaro e conciso i concetti di base e le caratteristiche principali delle tecnologie utilizzate nel progetto di stage e oggetto di studio autonomo e autodidatta.

2.1 Blockchain: concetti base

2.1.1 Introduzione

La blockchain è una tecnologia che permette di memorizzare dati in maniera decentralizzata e distribuita. Essa è una struttura dati che si comporta come un registro distribuito, salvando le informazioni in modo sicuro ed immutabile. La struttura è stata introdotta nel 2008 da Satoshi Nakamoto, che ha pubblicato il suo white paper *Bitcoin: A Peer-to-Peer Electronic Cash System*. Nel 2009 è stato pubblicato il primo software open source per la blockchain, Bitcoin, che ha permesso di creare una moneta digitale decentralizzata.

A tal fine, non si devono confondere le cosiddette criptovalute con la blockchain. Di fatto, quest'ultima è solo la struttura che permette lo scambio di beni di qualsiasi tipo, in modo sicuro e registrato ed immutabile. Una criptovaluta è invece una moneta digitale, che può essere scambiata con altre monete digitali o con beni fisici. Essendo lo standard blockchain *open source*, è possibile crearne di nuove con molta facilità.

Normalmente, viene utilizzata per memorizzare transazioni finanziarie, ma può essere utilizzata per memorizzare qualsiasi tipo di informazione. Un altro suo nome è *distributed ledger technology* (DLT), che indica che le sue informazioni sono registrate come su un libro mastro, in cui le singole componenti della rete, definite nodi, possono accedere e modificare i dati, stabilendo se questi sono validi o meno.

2.1.2 Blocco

I dati della blockchain sono strutturati in singoli blocchi, ciascuno contenente uno specifico set di informazioni. Ogni blocco è collegato al precedente tramite un hash, che ne garantisce l'immutabilità. I blocchi sono collegati in una catena, che viene aggiornata ogni volta che viene aggiunto un nuovo blocco. Nello specifico, possiamo dettagliare una struttura formata da:

- blocchi di dati;
- nonce, un numero generato casualmente alla creazione del blocco;
- l'hash del blocco precedente;
- il numero della transazione;
- *timestamp* di generazione del blocco (data e ora).

Per verificare l'integrità dei dati memorizzati, vengono usate delle strutture dati chiamati *Merkle trees*, in cui ogni foglia rappresenta l'hash della transazione. Le foglie vengono poi raggruppate in coppie, e l'hash di ogni coppia viene calcolato e memorizzato in un nodo superiore. Il processo si ripete, fino a raggiungere la radice dell'albero, che rappresenta l'hash di tutte le transazioni contenute nel blocco.

2.1.3 Transazione

Una transazione all'interno di una blockchain comporta il trasferimento di beni digitali, che possono essere valute, token, o qualsiasi altro tipo di informazione. In questo senso, possiamo individuare vari componenti del processo di transazione:

- gli utenti, che avviano le transazioni firmandole digitalmente con la propria chiave privata;
- i *miners*, che attraverso un processo specifico definito come *mining*, verificano la validità delle transazioni e le includono nel blocco successivo.
- i nodi, che convalidano i blocchi di transazioni inviati dai miners prima che vengano aggiunti alle blockchain.

Nello specifico, possiamo descrivere una transazione in questo modo:

1. l'utente avvia la transazione creando una firma digitale utilizzando la propria chiave privata. La firma dimostra che l'utente ha il diritto di inviare i beni;
2. la transazione viene trasmessa alla rete di nodi o computer che eseguono il software della blockchain. Ogni nodo riceve la transazione e la aggiunge a un pool di transazioni non confermate;
3. i nodi della rete convalidano la transazione per assicurarsi che il mittente abbia fondi sufficienti per completare la transazione e che questa sia conforme alle regole del protocollo blockchain;
4. una volta che un numero sufficiente di nodi ha convalidato la transazione, questa viene aggiunta a un nuovo blocco di transazioni, insieme ad altre transazioni convalidate di recente;
5. il blocco di transazioni viene aggiunto alla blockchain in un processo chiamato mining. L'estrazione comporta la risoluzione di complesse equazioni matematiche per creare un nuovo blocco, il che richiede una grande potenza di calcolo;
6. una volta aggiunto il nuovo blocco alla blockchain, la transazione viene considerata confermata e i beni vengono trasferiti dall'indirizzo del mittente a quello del destinatario. La transazione è ora registrata in modo permanente sul libro mastro della blockchain, che può essere visualizzato e verificato da chiunque abbia accesso alla rete;

2.1.4 Wallet

Un *wallet*, detto anche *portafoglio*, è un software che permette di memorizzare e gestire le chiavi private e pubbliche, e di inviare e ricevere transazioni. In particolare, possiamo più propriamente definirli portachiavi, in quanto non contengono realmente i beni digitali, ma le chiavi utilizzate per accedervi. L'utente dispone in ogni momento di:

- una chiave pubblica, usata per inviare messaggi e ricevere pagamenti. È un codice univoco che identifica l'utente;
- una chiave privata, usata per firmare i messaggi e per accedere ai propri beni digitali. È un codice segreto che deve essere conservato in modo sicuro.

Ogni wallet dispone di una frase segreta, che contiene tutte le informazioni necessarie per recuperare ed accedere ai fondi del proprio portafoglio. Inoltre, dispone di un proprio indirizzo, matematicamente derivato dalla stessa chiave pubblica mediante l'operazione di *hashing*, con una lunghezza di 160 bit. Ciascuno è *pseudonimo*, in quanto non appartiene nello specifico ad una persona, ma non è completamente anonimo.

Distinguiamo due tipi di wallet:

- *hot wallet*, che sono i portafogli online, dunque più vulnerabili al rischio di *hacking*;
- *cold wallet*, che sono i portafogli offline, quindi considerati più sicuri, in quanto si collegano ad Internet principalmente per effettuare le transazioni.

2.1.5 Mining

Il processo di mining consente di creare nuovi blocchi sulla catena, al fine di convalidare le transazioni e ottenere nuove criptovalute come ricompensa per il proprio "sforzo". Questo obiettivo viene raggiunto attraverso un processo chiamato consenso, che prevede la risoluzione di complessi puzzle matematici utilizzando la potenza di calcolo. Il miner che riesce a risolvere il puzzle prima degli altri, vince il diritto di aggiungere il blocco alla blockchain.

Si considera composto da due fasi:

- *hashing*, che consiste nella risoluzione di un puzzle matematico, che consiste nel trovare un numero che, una volta applicata una funzione di hash, abbia un valore inferiore ad un valore prefissato. Il valore di questo numero viene chiamato *nonce*;
- *ricerca del consenso*, che consiste nella verifica della validità del blocco, che viene effettuata da tutti i nodi della rete. Se il blocco è valido, viene aggiunto alla blockchain.

I minatori (miners) utilizzano un software speciale per risolvere il problema matematico incredibilmente complesso di trovare un nonce che generi un hash accettato. Poiché il nonce è di soli 32 bit e l'hash di 256, ci sono circa quattro miliardi di possibili combinazioni nonce-hash che devono essere estratte prima di trovare quella giusta.

Quando un blocco viene estratto con successo, la modifica viene accettata da tutti i nodi della rete e il miner viene ricompensato finanziariamente. Il primo minatore che risolve il puzzle e aggiunge un nuovo blocco alla blockchain viene ricompensato con un blocco di criptovaluta di nuovo conio. Il processo richiede un software specializzato e una grande potenza di calcolo, che può essere ottenuta utilizzando un computer o un gruppo di computer con grosso dispendio di energia e risorse.

2.1.6 Algoritmi di consenso

Il meccanismo di ricerca di consenso prevede numerose varianti, che differiscono per il modo in cui i nodi della rete si accordano per aggiungere un nuovo blocco alla blockchain.

Possiamo principalmente distinguere:

1. Proof of Work (PoW)
2. Proof of Stake (PoS)
3. Byzantine Fault Tolerance (BFT)

2.1.7 Tipi

2.2 Blockchain: concetti avanzati

2.2.1 Token

2.2.2 Tokenizzazione

2.2.3 Smart contract

2.2.4 Scalabilità

2.3 Self-Sovereign Identity

La self-sovereign identity (SSI) è un approccio all'identità digitale che dà agli individui il controllo sulle informazioni che usano per dimostrare chi sono a siti web, servizi e applicazioni in tutto il web. Senza l'SSI, gli individui con account (identità) persistenti su Internet devono affidarsi a una serie di fornitori terzi, come Facebook, Google e altri, che hanno il controllo delle informazioni associate alla loro identità.

Esistono molti modi per implementare l'SSI utilizzando le chiavi crittografiche e ne analizzeremo due.

- l'utilizzo di firme digitali, attraverso un processo di *firma digitale*, che permette di firmare un documento con una chiave privata, e di verificare la firma con la chiave pubblica;
- *Decentralized Identifier (DID)*, che è un identificatore univoco alfanumerico per un soggetto, che può essere utilizzato per identificare una persona, un'organizzazione, un dispositivo, un servizio, un documento, ecc.

Le parti coinvolte in questo processo sono principalmente tre:

1. emittente, detto anche *holder*, ossia l'entità che emette una credenziale, ad esempio un documento d'identità governativo;
2. titolare, detto anche *issuer*, ossia il proprietario della credenziale, cioè l'entità su cui l'emittente genera la credenziale;
3. verificatore, detto anche *verifier*, cioè l'entità che controlla la validità e l'autenticità della credenziale presentata dal titolare.

Si consideri che lo scopo principale di questa tecnologia è consentire agli utenti un'esistenza indipendente da provider terzi, permettendo loro di controllare la propria identità in modo sicuro e accedendovi senza dover affidarsi a terzi.

Inoltre, i sistemi e gli algoritmi che la supportano devono essere trasparenti, mantenendo le informazioni in modo trasparente e permanente, rendendo però semplice la portabilità e l'interoperabilità di queste all'interno delle varie piattaforme.

La tecnologia blockchain, per mezzo della sua natura decentralizzata, permette di creare un sistema di identità digitale che soddisfi questi requisiti. In particolare:

- il titolare della credenziale possiede il suo DID firmato dalla propria coppia di chiavi che certifica la sua identità;
- l'emittente fornisce delle *Verifiable Credentials (VC)*, che certificano in modo digitale e crittograficamente protetto la validità del proprio ruolo;
- il verificatore controlla che, tramite ciascun blocco, sia stata rilasciata una VC valida e che il titolare sia il legittimo possessore di quella VC.

L'obiettivo principale è quello di fornire un utilizzo delle tecnologie blockchain tali da permettere agli utenti di selezionare quali credenziali mostrare (*selective disclosure o divulgazione selettiva*), secondo opportuni standard definiti gradualmente dall'organizzazione internazionale *W3C*, tra cui il citato VC.

2.3.1 Protocolli

2.3.2 Casi d'uso reali

2.4 Zero Knowledge Proof

2.4.1 Tipi

2.4.2 Casi d'uso reali

Capitolo 3

Descrizione dello stage

Breve introduzione al capitolo

3.1 Introduzione al progetto

3.2 Analisi preventiva dei rischi

Durante la fase di analisi iniziale sono stati individuati alcuni possibili rischi a cui si potrà andare incontro. Si è quindi proceduto a elaborare delle possibili soluzioni per far fronte a tali rischi.

1. Performance del simulatore hardware

Descrizione: le performance del simulatore hardware e la comunicazione con questo potrebbero risultare lenti o non abbastanza buoni da causare il fallimento dei test.

Soluzione: coinvolgimento del responsabile a capo del progetto relativo il simulatore hardware.

3.3 Requisiti e obiettivi

3.4 Pianificazione

Capitolo 4

Analisi dei requisiti

Breve introduzione al capitolo

4.1 Casi d'uso

Per lo studio dei casi di utilizzo del prodotto sono stati creati dei diagrammi. I diagrammi dei casi d'uso (in inglese *Use Case Diagram*) sono diagrammi di tipo **Unified Modeling Language (UML)** dedicati alla descrizione delle funzioni o servizi offerti da un sistema, così come sono percepiti e utilizzati dagli attori che interagiscono col sistema stesso. Essendo il progetto finalizzato alla creazione di un tool per l'automazione di un processo, le interazioni da parte dell'utilizzatore devono essere ovviamente ridotte allo stretto necessario. Per questo motivo i diagrammi d'uso risultano semplici e in numero ridotto.

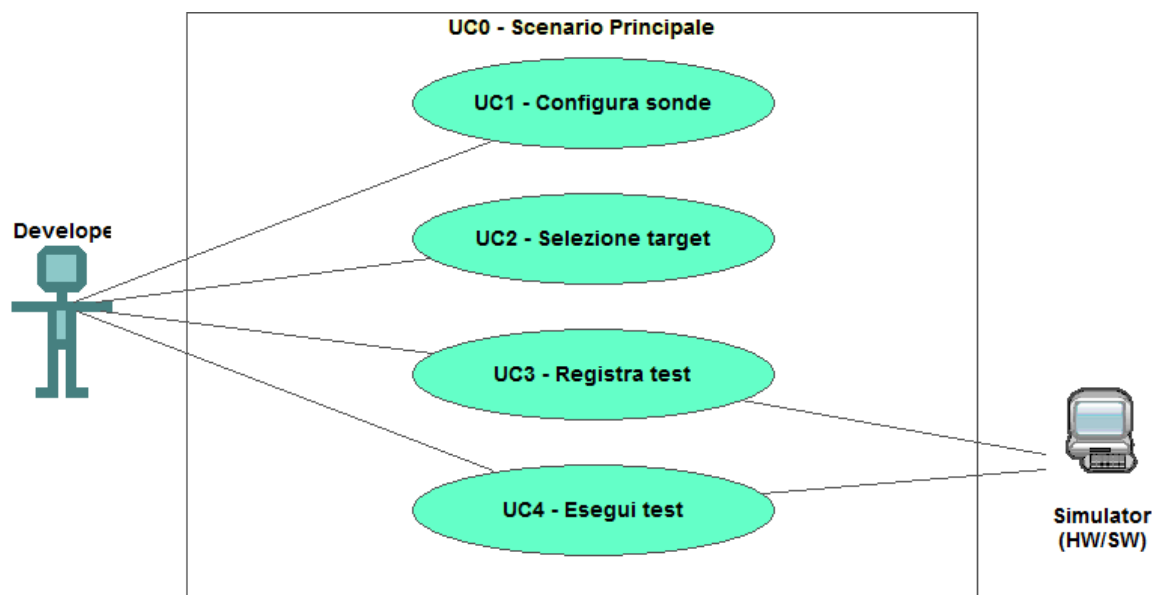


Figura 4.1: Use Case - UC0: Scenario principale

UC0: Scenario principale

Attori Principali: Sviluppatore applicativi.

Precondizioni: Lo sviluppatore è entrato nel plug-in di simulazione all'interno dell'IDE.

Descrizione: La finestra di simulazione mette a disposizione i comandi per configurare, registrare o eseguire un test.

Postcondizioni: Il sistema è pronto per permettere una nuova interazione.

4.2 Tracciamento dei requisiti

Da un'attenta analisi dei requisiti e degli use case effettuata sul progetto è stata stilata la tabella che traccia i requisiti in rapporto agli use case.

Sono stati individuati diversi tipi di requisiti e si è quindi fatto utilizzo di un codice identificativo per distinguerli. Il codice dei requisiti è così strutturato $R(F/Q/V)(N/D/O)$ dove:

R = requisito

F = funzionale

Q = qualitativo

V = di vincolo

N = obbligatorio (necessario)

D = desiderabile

Z = opzionale

Nelle tabelle [4.1](#), [4.2](#) e [4.3](#) sono riassunti i requisiti e il loro tracciamento con gli use case delineati in fase di analisi.

Tabella 4.1: Tabella del tracciamento dei requisiti funzionali

Requisito	Descrizione	Use Case
RFN-1	L'interfaccia permette di configurare il tipo di sonde del test	UC1

Tabella 4.2: Tabella del tracciamento dei requisiti qualitativi

Requisito	Descrizione	Use Case
RQD-1	Le prestazioni del simulatore hardware deve garantire la giusta esecuzione dei test e non la generazione di falsi negativi	-

Tabella 4.3: Tabella del tracciamento dei requisiti di vincolo

Requisito	Descrizione	Use Case
RVO-1	La libreria per l'esecuzione dei test automatici deve essere riutilizzabile	-

Capitolo 5

Progettazione e codifica

Breve introduzione al capitolo

5.1 Tecnologie utilizzate

In questa sezione, saranno elencate le tecnologie principali utilizzate durante lo sviluppo del sistema oggetto del tirocinio.

5.1.1 Codifica

Solidity

ethers.js

web3.js

5.1.2 Versionamento

GitHub

5.2 Ciclo di vita del software

5.3 Progettazione

Namespace 1

Descrizione namespace 1.

Classe 1: Descrizione classe 1

Classe 2: Descrizione classe 2

5.4 Design Pattern Utilizzati

5.5 Codifica

Capitolo 6

Verifica e validazione

Capitolo 7

Conclusioni

7.1 Consuntivo finale

7.2 Raggiungimento degli obiettivi

7.3 Conoscenze acquisite

7.4 Valutazione personale

Appendice A

Appendice A

Citazione

Autore della citazione

Bibliografia

Riferimenti bibliografici

James P. Womack, Daniel T. Jones. *Lean Thinking, Second Editon*. Simon & Schuster, Inc., 2010 (cit. a p. [1](#)).

Siti web consultati

Manifesto Agile. URL: <http://agilemanifesto.org/iso/it/> (cit. a p. [1](#)).