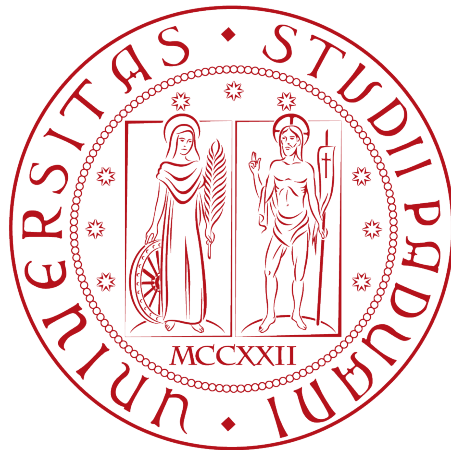


Università degli Studi di Padova

DIPARTIMENTO DI MATEMATICA “TULLIO LEVI-CIVITA”

CORSO DI LAUREA IN INFORMATICA



Titolo della tesi

Tesi di laurea

Relatore

Prof. Tullio Vardanega

Laureando

Pietro Lauriola

ANNO ACCADEMICO 2022-2023

Sommario

Il presente documento descrive il lavoro svolto durante il periodo di stage, della durata di trecentoventi (320) ore, dal laureando Pietro Lauriola presso l'azienda Sync Lab S.r.l. Gli obiettivi da raggiungere erano i seguenti:

In primo luogo era richiesto la stesura di uno Studio di fattibilità circa la possibilità di utilizzare la Zero Knowledge Proof per sviluppare una piattaforma di voto verificabile ma anonima, ovvero che raccolga le votazioni degli utenti approvati senza che sia possibile risalire a cosa abbiano votato ma rendendo facile la verifica del fatto che abbiano effettivamente votato. In secondo luogo era richiesta l'implementazione di un Proof of Concept (PoC) per dimostrare la fattibilità del progetto.

“And yet I smile”

— Ezekiel

Ringraziamenti

Padova, Settembre 2023

Pietro Lauriola

Indice

1	Contesto aziendale	1
1.1	L'azienda	1
1.2	Way of working	3
1.2.1	Tecnologie interne	3
1.3	Dallo stack tecnologico ad applicativo	5
1.4	Propensione all'innovazione	6
2	Lo stage	9
2.1	Strategia aziendale	9
2.1.1	Stage in azienda	9
2.2	Progetto proposto	9
2.3	Obiettivi	9
2.4	Vincoli	9
2.5	Motivazione della scelta	9
3	Il progetto: Svolgimento	11
3.1	Pianificazione	11
3.1.1	Interazione tutor	11
3.2	Analisi dei requisiti	11
3.2.1	Tracciamento requisiti	11
3.2.2	Verifica requisiti	11
3.3	Ricerca e studio tecnologie	11
3.3.1	Scelte tecnologiche e progettuali	12
3.4	Sviluppo in Solidity	12
3.5	Sviluppo in Angular	12
3.6	Sviluppo e pianificazione a confronto	12
3.7	Testing	12
3.7.1	Risultati	12
3.8	Conclusioni	12
4	Conclusioni	13
4.1	Copertura obiettivi	13
4.2	Importanza delle tecnologie blockchain e zero knowledge proof per la votazione elettronica	13
4.3	Conoscenze acquisite	13
4.4	Valutazione personale	13
A	Appendice A	15

Bibliografia

19

Elenco delle figure

1.1	Dati relativi all'azienda Fonte: synclab.it	1
1.2	Alcuni ambiti in cui l'azienda opera Fonte: synclab.it	2
1.3	Alcuni progetti sviluppati dall'azienda Fonte: synclab.it	3
1.4	Esempio di <i>stack</i> tecnologico Fonte: synclab.it	6

Elenco delle tabelle

Capitolo 1

Contesto aziendale

In questo capitolo verrà presentato il contesto organizzativo e produttivo dell'azienda *SyncLab S.r.L.*. Si fornirà una descrizione di ciò che ho potuto osservare riguardo le tecnologie utilizzate, i processi interni dell'azienda, il tipo di clientela e la propensione dell'azienda per l'innovazione.

1.1 L'azienda

L'azienda ospitante è stata ***SyncLab S.r.L.***, nata nel 2002 a Napoli e attiva nel settore dell' *Information and Communication Technology* (ITC). Con il passare degli anni si è espansa aprendo in tutto 6 sedi in Italia, a Napoli, Roma, Milano, Padova, Verona e Como.



Figura 1.1: Dati relativi all'azienda
Fonte: synclab.it

Grazie a una struttura interna che favorisce la collaborazione, l'interazione non si limita ai colleghi della stessa sede, ma si estende globalmente in tutta l'azienda.

Lo scopo è quello di promuovere lo scambio di conoscenze all'interno dell'organizzazione e creare un ambiente in cui il progresso personale non sia il risultato esclusivo degli sforzi individuali, ma anche della collaborazione attiva tra i membri del *team*.

È rilevante sottolineare che ho notato che i dipendenti presenti, almeno nella sede di Padova, sono principalmente giovani.

SyncLab S.r.L. è identificabile come *System Integrator*, sebbene sia nata come una *Software House*. La differenza tra i due ambiti è rilevante per comprendere il modus operandi dell'azienda.

Software House: un'azienda che sviluppa internamente delle soluzioni software che soddisfino una certa opportunità di mercato, e offre i propri prodotti ai clienti interessati.

System Integrator: un'azienda che, contattata da aziende esterne, effettua manutenzione e evoluzione delle funzionalità di prodotti software già sviluppati e in uso.

Si tratta quindi di due approcci allo sviluppo ben diversi: mentre una *Software House* si concentra sulla creazione e sviluppo di soluzioni software innovative da zero, basandosi sulle esigenze e opportunità del mercato, un *System Integrator* si focalizza sull'ottimizzazione, l'integrazione e la manutenzione di software esistenti in base alle esigenze dei clienti che già utilizzano tali prodotti.

SyncLab S.r.L. collabora con numerosi clienti, che operano in diversi ambiti, tra cui: *EHealth*, *Telco*, *Web and Mobile*, *Data Management*, *Blockchain*, *Maritime*.



Figura 1.2: Alcuni ambiti in cui l'azienda opera

Fonte: synclab.it

Dopo aver discusso degli attuali ambiti di operatività di *SyncLab*, è interessante fare un salto indietro e analizzare le radici dell'azienda.

Quando *SyncLab* operava come *Software House*, ha creato una serie di prodotti e soluzioni innovative che hanno contribuito a definire il suo prestigio nel settore IT. Prodotti e soluzioni che non solo hanno consolidato la sua posizione nel mercato, ma hanno anche gettato le basi per la sua evoluzione futura.

Vediamo alcuni dei progetti più rappresentativi e influenti realizzati da *SyncLab* in quella fase cruciale della sua storia.

- **SynClinic:** Una soluzione nel settore dell'EHealth progettata per centralizzare e facilitare la gestione sia clinica che amministrativa di ospedali, strutture sanitarie e residenze mediche.

Il suo obiettivo principale è assistere il personale sanitario nell'identificazione e nella gestione del rischio clinico, garantendo al contempo un'ottima tracciabilità e organizzazione delle varie tappe del trattamento del paziente.

- **SeaStream:** Questa piattaforma è stata progettata per potenziare l'efficienza, la sicurezza e incoraggiare l'innovazione nel dominio marittimo. SeaStream presenta un Fleet Operation Center (FOC) che offre una sorveglianza

avanzata delle flotte navali in operazione globalmente.

Inoltre, mette a disposizione una Harbor Operation Platform (HOC), che fornisce una gamma completa di servizi destinati ai professionisti del settore portuale.

- **Fast Reservation:** Una piattaforma digitale ideata per la gestione delle prenotazioni, adattabile a diverse realtà come stabilimenti balneari, parchi e attività nel settore della ristorazione. L'obiettivo è rendere il processo di prenotazione fluido e intuitivo per gli utenti.
- **Sobereye:** Una soluzione basata sul web progettata per analizzare e monitorare lo stato psicofisico di una persona mediante l'osservazione della pupilla. Questa applicazione è particolarmente utile per identificare alterazioni potenzialmente causate da stanchezza eccessiva o assunzione di sostanze come alcool e droghe, contribuendo a ridurre potenziali rischi sul luogo di lavoro.



Figura 1.3: Alcuni progetti sviluppati dall'azienda

Fonte: synclab.it

1.2 Way of working

Durante il percorso di tirocinio, sebbene la sede dell'azienda fosse a Padova, la maggior parte del lavoro è stato svolto da remoto. L'azienda adotta un metodo di lavoro che consiste nel trovarsi in sede circa una volta a settimana per confrontarsi sul lavoro svolto, sui progressi fatti e sulle difficoltà riscontrate.

Il restante del tempo lavorativo viene svolto in autonomia e, qualora ci si trovi in difficoltà, si possono utilizzare gli strumenti di comunicazione da remoto per risolvere dubbi o problemi.

1.2.1 Tecnologie interne

In questa sezione tratteremo alcune tecnologie di cui ho avuto esperienza diretta. Tuttavia, non si procederà a fornire specifiche dettagliate riguardo al loro utilizzo interno per la gestione del lavoro.

SyncLab utilizza un'ampia gamma di tecnologie, che includono linguaggi di programmazione e *framework* all'avanguardia nel settore dell'*Information and Communication Technology*.

L'azienda fa ampio uso di linguaggi di programmazione come **Java**, **Python**, **Solidity**, **Javascript**, **Typescript**.

Solitamente questi linguaggi vengono affiancati da *framework*, che offrono un'infrastruttura predefinita per lo sviluppo di applicazioni. Questi *framework* facilitano la creazione di *software* efficiente e scalabile, fornendo librerie predefinite, strumenti di sviluppo e modelli architetturali. Grazie alla loro natura modulare e flessibile, i *framework* consentono ai programmatori di concentrarsi sull'implementazione delle funzionalità specifiche, migliorando la qualità del software sviluppato. Di seguito i principali *framework* utilizzati:

- **Angular:** *Framework* di sviluppo *front-end* basato su JavaScript. Offre una potente piattaforma per la creazione di applicazione web scalabili e reattive. Attualmente è una delle soluzioni più utilizzate nel settore.
- **Java Spring:** *Framework* di sviluppo *back-end* basato su Java. Fornisce una vasta gamma di moduli e funzionalità per la creazione di applicazioni Java robuste. Attualmente si è imposto come *standard de facto* per lo sviluppo di servizi *web* in Java.
- **Odoo:** *Framework open source* per lo sviluppo di applicazioni di gestione aziendale, basato in Python. Odoo è altamente personalizzabile e modulare grazie al vasto insieme di moduli offerti per la gestione delle vendite, degli acquisti, delle risorse umane, della contabilità e molti altri.

Questa vasta gamma di tecnologie consentono di adattarsi alle esigenze specifiche dei clienti e offrire soluzioni *software* all'avanguardia che combinano efficienza, funzionalità e usabilità.

All'interno di un **gruppo di lavoro**, la normazione, regolamentazione e sincronizzazione delle attività sono fondamentali per garantire un flusso di lavoro efficiente ed efficace. A tal fine, l'utilizzo di *software* appositamente progettati svolge un ruolo cruciale.

Di seguito alcuni *software* che svolgono questo ruolo:

- **Git:** Un **sistema di controllo di versione distribuito**, ampiamente adottato. Consente di tenere traccia delle modifiche apportate ai file e coordinare il lavoro di più persone. Attraverso *Git*, i membri del gruppo possono collaborare in modo sincronizzato, gestire i conflitti, apportare modifiche senza sovrascrivere il lavoro degli altri e recuperare versioni precedenti dei *file*.
- **VS Code:** Un **ambiente di sviluppo integrato** (IDE) che offre un'interfaccia unificata per la scrittura del codice, la gestione dei *file* e la condivisione dei progetti. *VS Code* facilita la codifica collaborativa, fornendo strumenti per *debugging*, completamento del codice e integrazione con *Git*, agevolando la gestione e la tracciabilità delle modifiche al codice.
- **IntelliJ IDEA:** Un **ambiente di sviluppo integrato** (IDE) concepito principalmente per facilitare la programmazione in *Java*. Offre una vasta gamma di funzionalità destinate a assistere lo sviluppatore in ogni fase della codifica. Tra le caratteristiche più rilevanti, *IntelliJ IDEA* incorpora l'analisi statica del codice, permettendo di rilevare e segnalare errori sia logici che sintattici ancor prima dell'esecuzione del programma. Inoltre, si integra perfettamente con strumenti di versionamento esterni, come *Git*, agevolando la gestione e la tracciabilità delle modifiche al codice.

Nell'ambito della **comunicazione interna**, l'utilizzo di strumenti dedicati è essenziale per garantire una comunicazione efficace, per questo sono state individuate soluzioni come:

- **Discord:** Piattaforma di **comunicazione vocale e testuale**, ampiamente utilizzata e consente ai membri del gruppo di scambiare messaggi istantanei ed effettuare chiamate. *Discord* offre anche funzionalità aggiuntive, come la creazione di canali tematici per avere una comunicazione *topic based*.
- **Google Meet:** Piattaforma per **videoconferenze**, che permette di tenere riunioni online, condividendo schermi, documenti e presentazioni in tempo reale. Questo strumento è stato maggiormente usato durante e in seguito alla pandemia di SARS-CoV-2.
- **Google Calendar:** **Software gestionale** per la creazione di calendari privati e condivisi tra più utenti. Attraverso Google Calendar è possibile creare eventi, impostare promemoria e condividere le proprie disponibilità con gli altri membri. Viene anche utilizzato per organizzare l'alternanza tra *smart working* e lavoro in presenza.
- **Trello:** **Software gestionale** che adotta la filosofia Kanban, essendo ispirato alla metodologia della Scrum board tipica del modello agile. Permette di organizzare e tracciare il progresso dei lavori attraverso l'uso di schede specifiche, ciascuna associata a un determinato task. Questa strutturazione consente di avere una visione chiara e aggiornata dello stato dei progetti, e di facilitare la comunicazione e sincronizzazione tra i membri del team di sviluppo.

1.3 Dallo stack tecnologico ad applicativo

Nel vasto dominio dell'ingegneria del software, non si possono considerare le tecnologie in isolamento. Esse sono piuttosto dei pezzi che, messi insieme, costituiscono uno "stack tecnologico".

Questa integrazione ha particolare rilevanza quando ci si concentra sullo sviluppo di applicazioni web. Queste applicazioni, infatti, si articolano in due settori chiave: il *front-end*, focalizzato sull'esperienza e l'interfaccia utente, e il *back-end*, che gestisce la logica, la manipolazione dei dati e la loro persistenza.

Se dovessimo esemplificare utilizzando gli strumenti menzionati precedentemente:

- **Front-end:** La scelta potrebbe cadere su Angular come framework, concepito per creare interfacce web dinamiche. Questa scelta trascina con sé l'uso del linguaggio TypeScript e dell'HTML5. Sono tecnologie intrinsecamente collegate, funzionando in tandem per fornire l'esperienza desiderata.
- **Back-end:** In questo contesto, si potrebbe optare per Java Spring, ideale per elaborare logiche dati complesse, mentre la scelta del DBMS per la gestione dei dati potrebbe essere più flessibile e non direttamente vincolata dalla scelta del framework.

A queste tecnologie principali, si aggiungono strumenti che supportano il processo di sviluppo: IDE come IntelliJ IDEA per Java e VSCode per TypeScript; strumenti di

versionamento come Git e soluzioni per la comunicazione e gestione di progetto, come Discord e Trello.

Questi stack, o insiemi di tecnologie, interagiscono tra loro, formando un mosaico tecnologico che costituisce lo scheletro di qualsiasi progetto.

L'immagine seguente offre una sintesi visuale delle tecnologie discusse, categorizzate per ambito di utilizzo.

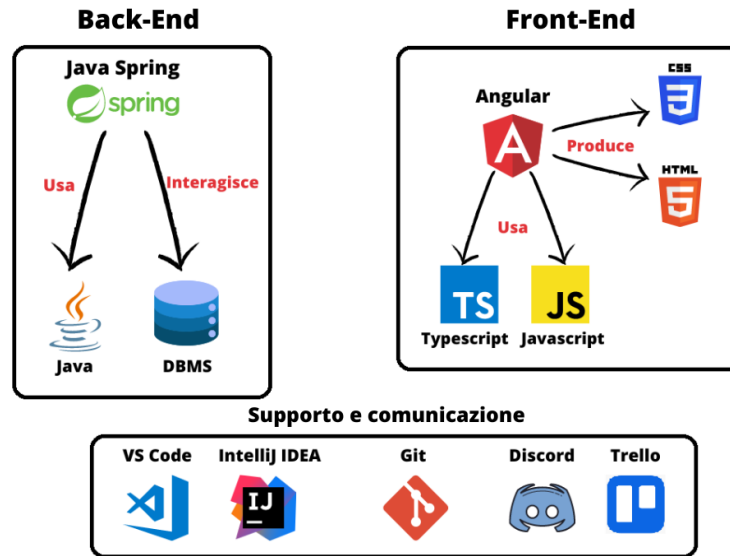


Figura 1.4: Esempio di *stack* tecnologico

Fonte: synclab.it

Il punto cardine è che le tecnologie adottate da SyncLab (o qualsiasi team di sviluppo) devono essere viste come elementi di un ecosistema più ampio, con una visione orientata verso l'obiettivo finale piuttosto che sui dettagli specifici.

Vale la pena sottolineare che gli stack tecnologici proposti qui sono illustrativi. Sono modellati intorno alle tecnologie prevalenti in SyncLab e servono come esempi di come si potrebbero combinare diverse tecnologie. La scelta di uno stack non è prescrittiva; dipende da numerosi fattori e non è intenzione di questo lavoro prescrivere un approccio unico o limitante.

1.4 Propensione all'innovazione

L'azienda si distingue per la sua costante propensione all'innovazione, che costituisce uno dei pilastri fondamentali della sua filosofia aziendale.

L'arduo e persistente impegno di *SyncLab* nel perseguire l'innovazione è palpabile in ogni aspetto della sua attività. Questa dedizione si traduce in una incessante e meticolosa ricerca di metodologie all'avanguardia, tecnologie di ultima generazione e approcci rivoluzionari.

Vi è un dualismo, che emerge dalla filosofia di *SyncLab*:

Da un lato, vi è l'incessante impegno per affinare le dinamiche interne, dall'ottimizzazione della gestione dei progetti, al potenziamento delle relazioni con i clienti, dalla ricerca di una maggiore efficienza operativa alla dedizione nella formazione continua del suo team. Queste iniziative riflettono la passione dell'azienda per l'eccellenza in ogni aspetto della sua operatività.

Dall'altro lato, *SyncLab* non si limita a perfezionare l'infrastruttura interna. La sua visione trascende queste operazioni interne, rivolgendo lo sguardo verso il panorama esterno. L'obiettivo predominante è la creazione e proposizione di soluzioni di avanguardia, formulate per rispondere alle sfide crescenti e intricate del mercato attuale. Questi prodotti e servizi non sono semplicemente reattivi, ma anticipatori, mirando a generare un impatto significativo e a lungo termine nell'ecosistema tecnologico.

La visione pionieristica

Oltre a rispondere alle immediate necessità dei clienti, l'intento è di anticipare le tendenze future e posizionarsi come pionieri nel settore. La visione dell'azienda è chiara: migliorare costantemente sia l'infrastruttura interna sia le soluzioni offerte, assicurando che entrambe siano sempre all'avanguardia.

La mentalità aperta all'innovazione e la capacità di adattamento rappresentano quindi i driver fondamentali che guidano l'azienda nel suo percorso di crescita e successo nel panorama competitivo odierno.

Un esempio tangibile della visione avanguardista di *SyncLab* è la sua dedizione verso la tecnologia *blockchain*.

Questo impegno non si riflette solo in una mera scelta tecnologica, ma evidenzia un profondo desiderio dell'azienda di essere al passo con le innovazioni più pregnanti nel dominio IT.

La mia esperienza all'interno dell'azienda ha rivelato come *SyncLab* non solo riconosca l'importanza cardine della *blockchain*, allocando risorse significative per comprenderla in profondità, ma stia anche realizzando lavori concreti basati su questa tecnologia per conto di clienti che ne hanno riconosciuto il potenziale e ne hanno richiesto l'implementazione.

Tali lavori, ancorati nella pratica e richiesti da clienti reali, mirano a sfruttare tutto il potenziale della *blockchain*.

L'obiettivo è doppio: da un lato, innovare e ottimizzare i processi esistenti, dall'altro, fornire nuove soluzioni che possano determinare una svolta nel panorama tecnologico attuale.

Questo approccio, che fonde analisi approfondita con applicazione diretta, dimostra la determinazione di *SyncLab* nel portare avanti l'innovazione.

L'azienda non si accontenta di stare al passo con le ultime tendenze, ma si posiziona attivamente come un agente di cambiamento, erogando servizi di alto valore ai suoi clienti.

Capitolo 2

Lo stage

2.1 Strategia aziendale

Quanto l'azienda si impegni in maniera proattiva all'avviamento di stage. I suoi rapporti con scuole ed università. Tipologie e ambiti delle attività proposte e come si riflettono sull'interesse nell'innovazione.

2.1.1 Stage in azienda

Come l'azienda interpreta gli stage in generale, in funzione dei propri obiettivi

2.2 Progetto proposto

Descrizione più approfondita del progetto che ho svolto.

2.3 Obiettivi

Introduzione: Parla dell'importanza di definire obiettivi chiari. Tabella degli obiettivi: Elenca gli obiettivi specifici che avevi per il progetto, con una breve descrizione di ciascuno.

2.4 Vincoli

Introduzione: Spiega perché è importante considerare i vincoli in qualsiasi progetto. Tabella dei vincoli: Elenca i vincoli specifici che hai affrontato durante il progetto, con una breve descrizione di ciascuno.

2.5 Motivazione della scelta

Ho scelto questo stage perché la tecnologia Blockchain mi affascina e mi sembra estremamente promettente. La possibilità di creare un sistema di votazione sicuro utilizzando zero-knowledge proof su blockchain Ethereum mi ha stimolato fin da subito. Inoltre, il lavoro con un'organizzazione all'avanguardia mi ha permesso di acquisire

competenze tecniche avanzate e di contribuire alla creazione di soluzioni innovative per risolvere i problemi reali delle persone.

Capitolo 3

Il progetto: Svolgimento

Breve introduzione al capitolo, se sarà ritenuta necessaria.

3.1 Pianificazione

Pianificazione del lavoro, gestione delle ore, con rappresentazione grafica.

3.1.1 Interazione tutor

Come abbiamo iniziato; Frequenza degli incontri, come venivano svolti; Revisioni e feedback intermedie, finali.

3.2 Analisi dei requisiti

Breve paragrafo che tratta il come è stata effettuata l'AdR con il Tutor. Alcuni requisiti sono stati definiti a priori dal tutor, altri sono stati ottenuti a seguito di discussioni tra me e il tutor.

3.2.1 Tracciamento requisiti

Quali sono stati i requisiti, obbligatori e non, funzionali e non e come sono stati tracciati.

3.2.2 Verifica requisiti

In che modo si è pianificato di verificare i requisiti sopra elencati.

3.3 Ricerca e studio tecnologie

Ho dedicato un tot di tempo alla ricerca delle tecnologie da utilizzare e il loro studio. Descrizione di alcune importanti tecnologie scelte.

3.3.1 Scelte tecnologiche e progettuali

Spiegazione di alcune scelte effettuate, ad esempio perché ho utilizzato alcune librerie piuttosto che altre, per la ZKP. Descrizione di alcune idee progettuali avute, e perché sono state scartate a favore di altre, ad esempio riguardanti il cercare di evitare, o meno, che qualche utente possa utilizzare più account per effettuare votazioni.

Potrebbero essere aperte altre sottosezioni, se il contenuto di questa sarà particolarmente ampio e/o complesso.

3.4 Sviluppo in Solidity

Come è stato effettuato lo sviluppo in Solidity, definizione del contratto e sue caratteristiche, come è stato caricato su blockchain. Vi saranno eventuali sottosezioni per separare gli argomenti, se il contenuto di essi sarà particolarmente ampio.

Un esempio, con titoli simbolici : "Creazione del contratto", "Caratteristiche del contratto", "Caricamento del contratto"

3.5 Sviluppo in Angular

Come è stato effettuato lo sviluppo in Angular.

3.6 Sviluppo e pianificazione a confronto

Diagramma di Gantt e confronto tra la pianificazione iniziale e l'andamento reale del progetto.

3.7 Testing

Descrizione di quali test sono stati realizzati e in che modo.

3.7.1 Risultati

Risultati ottenuti, con screenshot e/o grafici

3.8 Conclusioni

Capitolo 4

Conclusioni

4.1 Copertura obiettivi

Quali obiettivi sono stati raggiunti e quali non.
Causa del perché alcuni non sono stati raggiunti.

4.2 Importanza delle tecnologie blockchain e zero knowledge proof per la votazione elettronica

Riflessioni riguardanti l'uso della Blockchain e ZKP per lo svolgimento di una votazione elettronica. Pro e contro.

Considerazioni sia personali che oggettive, riguardo le tecnologie Blockchain e ZKP.
Sia utilizzate insieme, che prese singolarmente.

4.3 Conoscenze acquisite

Quali conoscenze considero acquisite.

4.4 Valutazione personale

Appendice A

Appendice A

Citazione

Autore della citazione

Bibliografia