

Università degli Studi di Padova

DIPARTIMENTO DI MATEMATICA "TULLIO LEVI-CIVITA "

CORSO DI LAUREA IN INFORMATICA



**Analisi e sviluppo front-end di una web
application in ambito blockchain/NFT**

Tesi di laurea

Relatore

Prof. Gilberto Filè

Laureando

Margherita Mitillo

ANNO ACCADEMICO 2020-2021

Chi non ha visto il calar della notte non giuri d'inoltrarsi nelle tenebre.

— John Ronald Reuel Tolkien

Dedicato a Daniele, Alessandra, Gianni e Luciano

Sommario

Il presente documento descrive il lavoro svolto durante il periodo di stage, della durata di circa trecento ore, dal laureando Margherita Mitillo presso l'azienda Sync Lab S.r.l. supervisionato e coordinato dal tutor aziendale Fabio Scettro. Lo scopo principale era lo sviluppo di maschere per il front end (ovvero interfacce utente), dell'applicazione NFTLab tramite il framework Vue.js.

Oltre a questo era previsto lo studio delle tecnologie coinvolte per la progettazione e la codifica del prodotto. Infine era richiesta l'implementazione di tali interfacce e la stesura di un documento tecnico che raccolga la descrizione di ciò che è stato implementato.

“Certe strade, è meglio intraprenderle che rifiutarle, anche se il loro esito è oscuro”

— John Ronald Reuel Tolkien

Ringraziamenti

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

Ho desiderio di ringraziare il mio tutor aziendale Fabio Scettro per avermi seguito durante il periodo di stage.

Ringrazio con tanto affetto il mio ragazzo Daniele per avermi supportato e aver sempre creduto in me durante tutto periodo di studi. Senza di lui probabilmente non sarei arrivata fino a qui.

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

Ho desiderio di ringraziare poi i miei amici, in particolare Alberto, per i bellissimi anni passati assieme.

Padova, Luglio 2021

Margherita Mitillo

Indice

1	Introduzione	1
1.1	Organizzazione del testo	1
2	Descrizione dello stage	3
2.1	L'azienda	3
2.1.1	La storia	3
2.1.2	I prodotti	4
2.2	L'idea dello stage	4
2.3	Organizzazione del lavoro	5
2.3.1	Modello di sviluppo	5
2.3.2	Organizzazione del lavoro	6
3	Nozioni apprese	9
3.1	Organizzazione dello studio	9
3.2	Nozioni principali	9
3.3	Nozioni secondarie	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 e strumenti	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
	Glossario	23

Acronimi	25
Bibliografia	27

Elenco delle figure

2.1	Logo di Sync Lab	3
2.2	Logo di NFTLab	4
2.3	Metodo Scrum	6
2.4	Spezzone del report giornaliero	7
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

Lo scopo di questo progetto di stage è studiare ed implementare tramite il framework Vue.js parte delle componenti di front-end di un'applicazione web in ambito Blockchain e NFT chiamata NFTLab.

Tramite questa applicazione sarà possibile creare un proprio profilo per poter compiere diverse azioni come visualizzare o vendere le proprie opere multimediali e comprare le stesse da altri utenti.

Per realizzare questi aspetti sono state implementate diverse maschere, ovvero interfacce utente.

Un aspetto importante di questo progetto è la collaborazione con gli altri componenti del gruppo per integrare gli sviluppi con quelli dei colleghi.

1.1 Organizzazione del testo

Il secondo capitolo approfondisce la descrizione dello stage, l'organizzazione del lavoro e il progetto da sviluppare;

Il terzo capitolo approfondisce le nozioni apprese nel periodo di stage;

Il quarto capitolo approfondisce il processo di analisi dei requisiti;

Il quinto capitolo approfondisce i processi di progettazione e codifica;

Il sesto capitolo approfondisce i processi di validazione e verifica del codice prodotto;

Il settimo capitolo descrive le conclusioni tratte alla fine del periodo di 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*.

Capitolo 2

Descrizione dello stage

In questo capitolo verrà descritto in dettaglio l'idea dello stage, l'azienda dove è stato svolto e come è stato organizzato il lavoro.

2.1 L'azienda

2.1.1 La storia

Sync Lab S.r.l. è un'azienda di consulenza informatica nata nel 2002 nella sede di Napoli e trasformatasi molto velocemente nel corso degli anni in System Integrator grazie ad un processo di maturazione delle competenze tecnologiche. Supporta le esigenze di innovazione dei clienti offrendo soluzioni IT in ambito Business Consultancy, Project Financing e IT Consultancy.

Ad oggi l'azienda conta un organico di circa 200 dipendenti ed è riuscita a coprire



Figura 2.1: Logo di Sync Lab

tutto il territorio nazionale fino ad ottenere un totale di cinque sedi a Roma, Napoli, Verona, Padova e Milano. L'azienda, propone sul mercato prodotti software, attraverso essi Sync Lab ha gradualmente conquistato significativamente fette di mercato nei seguenti settori: mobile, videosorveglianza e sicurezza delle infrastrutture informatiche aziendali. Il loro obiettivo è quello di supportare il cliente nella Realizzazione, Messa in Opera e Governance di soluzioni IT, sia dal punto di vista Tecnologico, sia nel Governo del Cambiamento Organizzativo.

2.1.2 I prodotti

I prodotti più noti dell'azienda sono i seguenti:

- * SynClinic: sistema informativo sanitario per la gestione dei processi clinici ed amministrativi di ospedali, cliniche e case di cura. Lo scopo è diventare uno

strumento indispensabile di supporto per il personale nella gestione del rischio clinico di ogni paziente;

- * DPS4.: web application per gestire gli adempimenti della privacy GDPR, acronimo per General Data Protection Regulation, con lo scopo di rafforzare la privacy dei dati dei cittadini;
- * StreamLog, sistema in grado di effettuare in modo semplice ed efficace il controllo degli accessi degli utenti ai sistemi;
- * StreamCrusher, software in grado di analizzare i dati che aiuta ad essere informati adeguatamente su quando bisogna prendere decisioni di business, ad identificare criticità e riorganizzare il sistema;
- * Wave: acronimo per Wide Area Videosurveillance Environment, è un prodotto software nato per ottenere un'integrazione sinergica tra i mondi della sorveglianza e dei Sistemi Informativi Territoriali (GIS) in modo da ottenere una copertura totale del territorio;
- * Seastream: piattaforma in grado di migliorare la sicurezza, l'efficienza e il processo di innovazione in campo marittimo.

2.2 L'idea dello stage

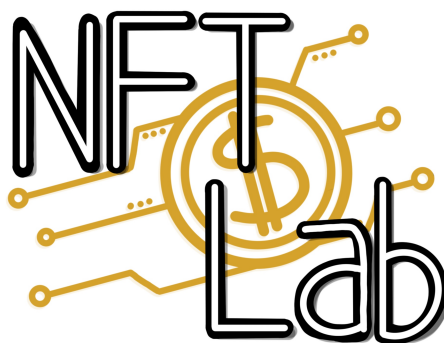


Figura 2.2: Logo di NFTLab

Il progetto da svolgere proposto dall'azienda per lo stage è una web application chiamata NFTLab legata all'ambito blockchain e NFT, due argomenti che stanno spopolando al momento e rendono contemporanea ed interessante il progetto di stage. In dettaglio quando l'utente accede al sito potrà visualizzare le opere multimediali in vendita, fare l'accesso o la registrazione al sito. Dopo essere acceduto, l'utente potrà caricare le proprie opere e decidere se metterle in vendita o meno. Inoltre potrà sfogliare il catalogo delle opere presenti e decidere di comprarne a sua volta.

Essendo legato al concetto di blockchain e NFT le azioni di compravendita verranno effettuate tramite lo scambio di una moneta virtuale detta Ethereum. Un utente nel momento in cui carica un'opera diventa l'autore ed il proprietario di quest'ultima, ma se la venderà l'acquirente diventerà il nuovo proprietario ma solo l'utente che ha caricato l'opera è e sarà sempre l'autore.

Per realizzare questi aspetti sono state implementate diverse maschere, ovvero interfacce utente, tramite il framework Vue.js, uno strumento utile allo sviluppo di web application.

Parte del percorso di stage è stata riservata ad un periodo di studio (senza implementazioni se non a scopo didattico) delle tecnologie legate al back end , in particolare il framework Spring.

2.3 Organizzazione del lavoro

2.3.1 Modello di sviluppo

L'azienda adotta il modello di sviluppo Agile che si basa sull'interazione continua con gli stakeholder e predilige:

- * gli individui e le loro interazioni piuttosto che i processi e gli strumenti;
- * il software funzionante piuttosto che una documentazione esaustiva;
- * la collaborazione con il cliente piuttosto che la negoziazione dei contratti;
- * essere propositivi verso il cambiamento piuttosto che rifiutarlo.

L'idea di questo modello non si basa quindi su uno sviluppo sequenziale ma sul concetto di rivedere di continuo le specifiche adeguandole durante l'avanzamento dello sviluppo software. In questo modo si possono apportare agilmente modifiche mediante metodologie iterative ed incrementali ed un continuo scambio di pareri ed informazioni tra gli sviluppatori ed il committente. Il progetto di stage è stato portato avanti utilizzando il metodo Scrum.

Scrum è il metodo più diffuso e prevede di dividere il progetto in blocchi di lavoro detti sprint, alla fine dei quali si crea un incremento del prodotto software. Esso prevede vari meeting per controllare il lavoro svolto ed organizzare il lavoro da svolgere nel prossimo periodo. Inoltre nel corso dello sviluppo l'azienda organizza diversi meeting con gli stakeholder e i membri del progetto per valutare l'andamento dello sprint.

Gli incontri legati a questo modello di sviluppo che vengono svolti di solito sono:

- * sprint planning meeting: tenuto ad inizio sprint per poter organizzare il lavoro;
- * daily scrum: riunione giornaliera del team di sviluppo per monitorare l'andamento dello sprint;
- * sprint review: tenuto a fine sprint per valutare i risultati ottenuti e quali cambiamenti apportare per lo sprint successivo.

La situazione di emergenza sanitaria ancora non conclusa mi ha portato a vivere un'esperienza lavorativa diversa dai colleghi degli anni precedenti, infatti ha fatto sì che lo stage fosse organizzato in modalità mista: in parte da remoto ed in parte in azienda (di solito uno o due giorni alla settimana). A causa di ciò non ho potuto vivere appieno questo modello di sviluppo vista l'impossibilità di essere sempre presente in azienda.

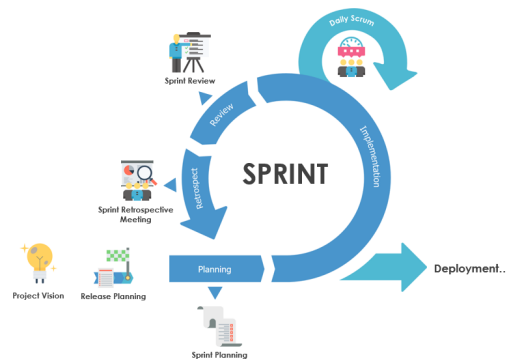


Figura 2.3: Metodo Scrum

Per questo motivo il giorno concordato con i colleghi stagisti ed il tutor aziendale per andare in azienda è stato sfruttato per fare un daily scrum alternativo data la sua cadenza settimanale.

Nonostante l'emergenza sanitaria ancora in corso ho potuto comunque constatare l'efficacia e la funzionalità di questo modello di sviluppo, soprattutto per un progetto software in evoluzione quale è stato quello a cui ho lavorato.

2.3.2 Organizzazione del lavoro

Trello

Per tenere traccia delle attività da svolgere è stato deciso di utilizzare la piattaforma online Trello. Tramite essa l'azienda ha creato due bacheche:

- * personale: disponibile solo allo stagista ed al tutor aziendale, dove sono stati inseriti i compiti decisi nel piano di lavoro suddivisi per settimana;
- * collettiva: disponibile a tutti gli stagisti che lavorano allo stesso progetto e ai loro rispettivi tutor aziendali, dove sono state inserite le prime idee e funzionalità legate al progetto.

Report giornaliero

Oltre a tenere traccia delle attività tramite Trello, l'azienda ha deciso di far scrivere allo stagista un file Excel su Drive come report giornaliero delle attività svolte. Il report era suddiviso in quattro colonne:

- * data: indica il giorno in è stata fatta un'attività in formato AAAA-MM-GG;
- * descrizione: breve descrizione dell'attività svolta dal tirocinante compilata alla fine di ogni giornata lavorativa;
- * nota: campo dove il tutor aziendale può aggiungere note di consiglio riguardo all'attività svolta, inoltre in questo campo vengono specificati i giorni festivi;
- * spunta: casella di spunta dove il tutor indica di aver visionato l'attività svolta.

Giorno	Attività	Nota	Conferma del tutor
2021-05-03	Organizzazione lavoro, ripasso Java e Servlet		ok
2021-05-04	Ripasso del concetto di JSON, inizio dello studio del pattern REST	per approfondire Rest vs Restful: https://martinfowler.com/articles/richardsonMaturityModel.html	ok
2021-05-05	Continuazione studio del pattern REST		ok
2021-05-06	Studiato Rest vs Restful, principi del REST e best practice delle REST API	Consiglio: se non l'hai mai visto dai un'occhiata a postman	ok
2021-05-07	Concluso lo studio di REST con esercizi pratici e ripasso generale delle conoscenze apprese nel corso della settimana		ok

Figura 2.4: Spezzone del report giornaliero

Discord

Per comunicare con l'intero gruppo di stagisti l'azienda ha proposto di utilizzare Discord, una piattaforma messaggistica dove noi stagisti potevamo comunicare sia con il personale aziendale che singolarmente con il proprio tutor per consigli, delucidazioni o motivi organizzativi.

Capitolo 3

Nozioni apprese

In questo capitolo verranno elencate e spiegate brevemente le nozioni apprese durante questo percorso di stage.

3.1 Organizzazione dello studio

Le ore complessive dello stage possono essere considerate suddivise in 3 periodi:

- * periodo di studio della durata di circa 160 ore;
- * periodo di sviluppo della durata di circa 120 ore;
- * periodo di convalidazione finale della durata di circa 40 ore.

Nelle sezioni qui di seguito verranno elencate e spiegate le nozioni apprese nel periodo di studio.

Per chiarezza sono state suddivise in principali e secondarie, non per dare meno importanza ad alcuni argomenti studiati ma per suddividerli in funzione dello sviluppo del progetto. Infatti le prime mi sono maggiormente servite per poter sviluppare le maschere, mentre le seconde mi sono servite per conoscenza generale e migliore collaborazione con i colleghi addetti al reparto back end.

3.2 Nozioni principali

Le nozioni principali da apprendere legate a questo progetto furono:

- * Javascript;
- * Vue.js;

3.3 Nozioni secondarie

Le nozioni secondarie da apprendere legate a questo progetto furono:

- * Java;
- * Concetti web;

- * Spring;
- * Blockchain;
- * Ethereum;
- * NFT.

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.

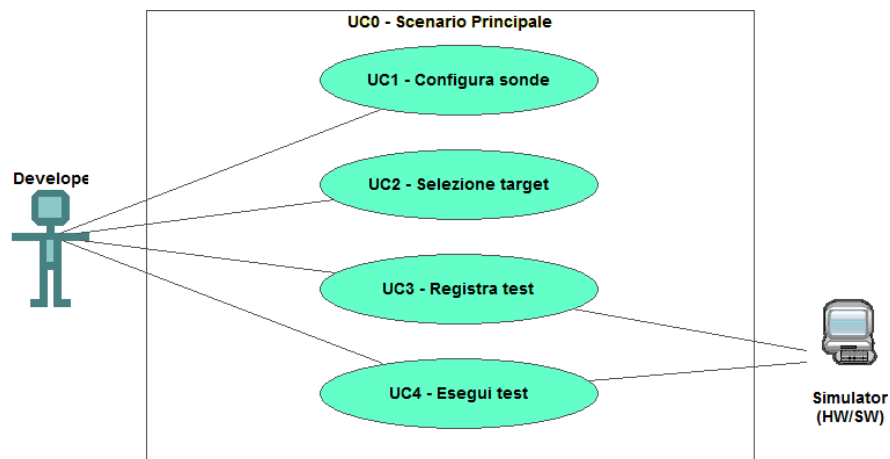


Figura 4.1: Use Case - UC0: Scenario principale

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 e strumenti

Di seguito viene data una panoramica delle tecnologie e strumenti utilizzati.

Tecnologia 1

Descrizione Tecnologia 1.

Tecnologia 2

Descrizione Tecnologia 2

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

Glossario

API in informatica con il termine *Application Programming Interface API* (ing. interfaccia di programmazione di un'applicazione) si indica ogni insieme di procedure disponibili al programmatore, di solito raggruppate a formare un set di strumenti specifici per l'espletamento di un determinato compito all'interno di un certo programma. La finalità è ottenere un'astrazione, di solito tra l'hardware e il programmatore o tra software a basso e quello ad alto livello semplificando così il lavoro di programmazione. [21](#)

UML in ingegneria del software *UML, Unified Modeling Language* (ing. linguaggio di modellazione unificato) è un linguaggio di modellazione e specifica basato sul paradigma object-oriented. L'*UML* svolge un'importantissima funzione di “lingua franca” nella comunità della progettazione e programmazione a oggetti. Gran parte della letteratura di settore usa tale linguaggio per descrivere soluzioni analitiche e progettuali in modo sintetico e comprensibile a un vasto pubblico. [21](#)

Acronimi

API [Application Program Interface](#). 1, 19

UML [Unified Modeling Language](#). 7, 19

Bibliografia