

Università degli Studi di Padova

DIPARTIMENTO DI MATEMATICA "TULLIO
LEVI-CIVITA"

CORSO DI LAUREA IN INFORMATICA



**FabKey: sviluppo della serratura online,
tra IoT e automazione**

Tesi di laurea triennale

Relatore

Prof. Tullio Vardanega

Laureando

Federico Vegro

ANNO ACCADEMICO 2017-2018

Federico Vegro: *FabKey: sviluppo della serratura online, tra IoT e automazione*,
Tesi di laurea triennale, © Settembre 2018.

“Computers are incredibly fast, accurate, and stupid. Human beings are incredibly slow, inaccurate, and brilliant. Together they are powerful beyond imagination”

— Albert Einstein

Ringraziamenti

Indice

1	Contesto aziendale	1
1.1	Lab Network	1
1.2	Organizzazione aziendale	1
1.3	Prodotti e servizi	2
1.3.1	Prodotti	2
1.3.2	Servizi	4
1.3.3	Tecnologie	4
1.4	Lab Network e innovazione	7
1.4.1	Pensiero aziendale	7
1.4.2	Smart Specialisation Strategy	8
2	Il progetto nella strategia aziendale	10
2.1	Azienda e stage	10
2.2	Obiettivi aziendali	12
2.3	Vincoli	14
2.4	Obiettivi personali	15
3	Lo stage	16
3.1	Pianificazione	16
3.2	Studio iniziale	17
3.2.1	Documentazione individuale	17
3.3	Analisi dei requisiti	19
3.3.1	Identificazione	19
3.3.2	Strumenti a supporto	19
3.3.3	Formalizzazione dei requisiti	20
3.4	Progettazione	22
3.4.1	L'approccio iniziale	22
3.4.2	Progettazione di HWFabKey	22
3.4.3	Progettazione del gestionale accessi	26
3.5	Codifica e realizzazione	29
3.5.1	Tecnologie e linguaggi	29
3.5.2	Segnalazione bug e relativa risoluzione	30
3.5.3	Stampa 3D	30
3.5.4	Redazione dei manuali	31
3.6	Verifica e validazione	32
3.6.1	Test	32

4 Valutazione retrospettiva	36
4.1 Obiettivi raggiunti	36
4.2 Valutazione formativa	36
4.3 Considerazioni personali	36
A Appendice A	37
Bibliografia	38

Elenco delle figure

1.1	Vitruvian Game - Il simulatore di volo con tuta alare	3
1.2	Prototipo di FabKey durante la fase di programmazione	3
1.3	Smart Meter: il metro intelligente	4
1.4	Le stampanti 3D in dotazione a Lab Network Srl	5
1.5	L'ambiente virtuale realizzato con Unreal Engine durante la fase di sviluppo	6
1.6	Previsione degli infortuni sul lavoro: schema di funzionamento . . .	7
1.7	Punti chiave della Smart Specialisation Strategy. Video " <i>The Kingdom of Smart</i> "	8
1.8	Parti coinvolte e ruoli nella Smart Specialisation Strategy.	9
2.1	Dati riportanti il tempo di assunzione dopo lo stage. In percentuale sull'asse delle ordinate sono riportati gli studenti che hanno effettuato uno stage presso Lab Network Srl; sull'asse delle ascisse i mesi trascorsi tra il termine dello stage e l'impiego in un'azienda.	11
2.2	Prototipo funzionante di FabKey prima dell'inizio dello stage.	13
2.3	Il prodotto finito dopo il termine dello stage.	14
2.4	Diagramma di Gantt per la pianificazione del progetto di stage.	14
3.1	Esempio di schema creato con Fritzing. Rappresenta uno dei primi test effettuati con scheda Arduino.	18
3.2	Estratto dello schema UML rappresentante i casi d'uso dello scenario principale. Schema riferito all'uso di FabKey e non alla piattaforma online.	19
3.3	Dettaglio del caso d'uso UC_FK.4. Si può notare l'uso della notazione sopra descritta.	20
3.4	Schema di funzionamento di FabKey.	23
3.5	Una delle componenti elettroniche che compongono il lettore di codici a barre. Si tratta del vero e proprio lettore ottico.	24
3.6	Esempio di funzione hash. Risultato dei primi quattro byte della funzione hash SHA-1.	24
3.7	Schema di comunicazione Centralina/Server.	25
3.8	La prima versione dell'involucro per il lettore di codici a barre.	26
3.9	Un componente dell'involucro durante la simulazione con il software Cura.	26
3.10	Schermata principale del gestionale.	27
3.11	Schermata del pannello di controllo degli accessi.	28

3.12	I primi prototipi di involucro stampati in 3D.	31
3.13	Diagramma rappresentante l'esecuzione di un test di unità.	33

Elenco delle tabelle

3.1	Alcuni requisiti individuati per lo sviluppo della parte hardware di FabKey.	21
3.2	Alcuni requisiti individuati per lo sviluppo del gestionale online di FabKey.	22
3.3	Risultati dei test di unità.	35
3.4	Risultati dei test di sistema.	35

Capitolo 1

Contesto aziendale

1.1 Lab Network

Lab Network Srl nasce nel 2016 con lo scopo di aiutare le imprese a creare e ad innovare prodotti e processi attraverso la competenza concreta dei laboratori, sfruttando le potenzialità dei moderni strumenti digitali.

Nel dettaglio Lab Network Srl offre la possibilità di importanti avanzamenti tecnologici a PMI offrendo servizi di consulenza oltre che ricerca e sviluppo di progetti sperimentali.

L'azienda cerca di collaborare con più partner per poter ottenere una visione maggiore di prodotti e di conseguenza soddisfare i clienti.

L'ambiente lavorativo è condiviso con un'altra azienda: *Business Research Srl*, la quale si occupa di soluzioni software su misura, applicazioni per aziende, e-commerce, cloud e hosting.

Tra le due aziende è in vigore un accordo che le lega in una stretta collaborazione: infatti, nel periodo di sviluppo di un nuovo progetto, Business Research si occupa della parte software.

Durante lo stage ho di fatto interagito con il personale di quest'ultima azienda per la realizzazione del progetto.

1.2 Organizzazione aziendale

Lab Network Srl è un'azienda giovane ma intraprendente e farne parte significa entrare in un gruppo eterogeneo di aziende che collaborano per un fine comune.

L'azienda opera principalmente nel territorio veneto, cercando di coinvolgere le PMI del territorio in un processo di aggiornamento tecnologico seguendo tre fasi:

- Creare l'interesse attraverso politiche di marketing discutendo di temi specifici ad alto impatto mediatico;
- Raccogliere gruppi omogenei che condividono l'interesse ad una specifica tecnologia;
- Capire le esigenze produttive e le potenzialità che una digitalizzazione (hardware o software) può contribuire alla crescita di una PMI, attraverso attività di consulenza e formazione.

Il ruolo più importante di *business scouting* è affidato al **dirigente aziendale**, che si occupa quindi della valutazione di idee imprenditoriali, analizzandone la fattibilità.

La **segreteria** ha il compito organizzativo per quanto riguarda appuntamenti e gestione di eventi (fiere, presentazioni, convegni, ecc.) oltre che quello di relazione con i clienti.

L'**amministrazione** si occupa della gestione dell'aspetto finanziario dell'azienda: fatturazione, pagamenti e rapporti con le banche.

Il **team di sviluppo** viene creato di volta in volta in collaborazione con Business Research Srl, che fornisce le risorse umane.

Dal momento in cui una PMI decide di investire in un aggiornamento tecnologico attraverso l'introduzione di un nuovo prodotto o servizio, Lab Network Srl per prima cosa sviluppa un piano di lavoro che prevede una fase di ricerca, seguita poi dalla realizzazione vera e propria.

Nella fase di ricerca Lab Network Srl coinvolge varie aziende specializzate creando così una rete di imprese che collaborano per portare a termine il progetto innovativo, oggetto della ricerca.

Durante il periodo di stage ho potuto conoscere come lavora il team di sviluppo: viene adottata una metodologia di tipo agile, ciò consente di avere un dialogo continuo con il cliente che può decidere di modificare il progetto in corso d'opera. Ho appreso inoltre che il team fa riferimento al framework scrum per la gestione dei processi e dei ruoli.

Durante il periodo di sviluppo le giornate iniziavano spesso con una riunione alla quale presenziavano tutti i componenti del team: tale incontro è riconducibile al *Daily Scrum*. Questa operazione serve ad aggiornare il team sullo stato di avanzamento del progetto e riesaminare gli obiettivi della giornata.

Per tenere traccia dei requisiti (*Product Backlog*), gestire il controllo di versione e il ticketing abbiamo utilizzato la piattaforma di GitLab.

Sono stati spesso organizzati incontri con il cliente e con gli stakeholders, anche presso la loro sede, per discutere dell'evoluzione del progetto e, talvolta, effettuare dei test con i prototipi realizzati.

1.3 Prodotti e servizi

1.3.1 Prodotti

VITRUVIAN GAME – Wingsuit VR

Si tratta di un progetto nato dalla collaborazione con Intel® e rappresenta un simulatore di volo con tuta alare.

La sua forma è ispirata all'*Uomo Vitruviano* di Leonardo da Vinci.

Il progetto prevede l'utilizzo del visore HTC Vive affiancato ad uno scenario virtuale sviluppato con motore grafico *Unreal Engine*. Per i movimenti sugli assi sono stati impiegati dei motori per automazione industriale.

L'utente può controllare i movimenti tramite dei controller posti su entrambe le mani.



Figura 1.1: Vitruvian Game - Il simulatore di volo con tuta alare

FabKey

FabKey è una serratura smart composta da un sistema di controllo connesso ad internet che permette l'apertura di una porta, andando a verificare una lista di accessi presente in cloud. Funziona tramite lettura di tag NFC_G o barcode. Il prodotto è rivolto principalmente ai FabLab, ma si adatta facilmente a qualsiasi contesto in cui sia richiesto il controllo degli accessi.
Le principali tecnologie utilizzate in questo prodotto sono Arduino e relativa programmazione in C/C++ per la parte hardware, mentre per il software crm sono stati utilizzati linguaggi come Html, Javascript, NodeJS, CSS ecc.

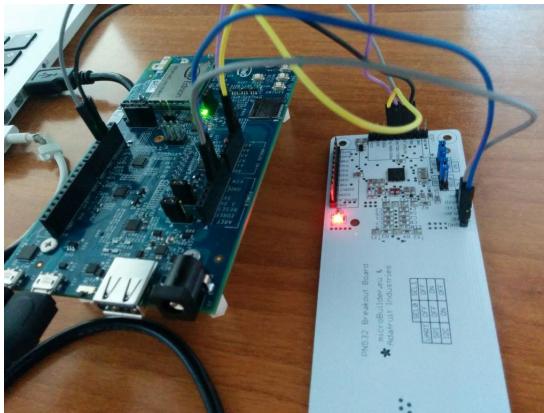


Figura 1.2: Prototipo di FabKey durante la fase di programmazione

Smart Meter

Un metro intelligente dotato di rotella metrica ed encoder ottico, capace di misurare superfici complesse e inviare direttamente i dati al software gestionale, in aggiunta anche il monitoraggio in tempo reale di quando, dove e per quanto tempo è stato utilizzato dal singolo addetto.

Il dispositivo è stato sviluppato sulla base di schede elettroniche programmabili open source.



Figura 1.3: Smart Meter: il metro intelligente

1.3.2 Servizi

Il servizio principale che Lab Network Srl offre ai propri clienti è quello di accompagnare le aziende in un percorso di aggiornamento tecnologico, fornendo supporto in termini di tecnologie e conoscenze. Durante l'esperienza di stage ho potuto verificare i passaggi che caratterizzano questo servizio:

- Analisi dettagliata delle esigenze del cliente, tramite diversi incontri con gli stakeholders;
- Studio di fattibilità grazie ad un'attenta analisi di mercato;
- Progettazione e/o implementazione dell'innovazione di processo o prodotto lavorando in sinergia con le aziende partner che più si avvicinano alle materie trattate.

Altri servizi che Lab Network Srl offre ai propri clienti sono:

- **Didattica:** corsi di formazione su misura, counseling e workshop per imparare a sfruttare in modo professionale: stampa 3D, schede elettroniche, realtà virtuale e aumentata, sviluppo applicazioni, prototipazione, Big Data, IoT ecc.
- **Noleggio:** kit e attrezzature come stampanti 3D, schede elettroniche (Arduino, Raspberry Pi, Intel) visori VR, pc e notebook, videoproiettori; affitto di aule didattiche;
- **Personale qualificato:** per i clienti sono a disposizione docenti, tecnici di laboratorio e consulenti specializzati nei principali ambiti di innovazione digitale.

1.3.3 Tecnologie

L'intero team utilizza ambienti con sistema operativo *macOS* per le attività di sviluppo.

Elettronica e Hardware

A supporto dei processi di prototipazione, Lab Network Srl utilizza diverse piattaforme hardware come ad esempio *Arduino*, *Raspberry Pi*, *Intel UP Square*, *Asus TinkerBoard*.

La programmazione delle schede Arduino avviene tramite l'omonimo ambiente di sviluppo integrato, con linguaggi C/C++.

Lab Network Srl, inoltre, sfrutta le potenzialità della stampa 3D e del disegno CAD per creare modelli prototipali in modo rapido. La disponibilità in azienda di 3 stampanti 3D permette di ridurre il tempo per la realizzazione dei prototipi da proporre al cliente.

Tali tecnologie sono state ampiamente utilizzate per lo sviluppo di progetti come FabKey e SmartMeter, soprattutto in fase di sperimentazione e prototipazione.

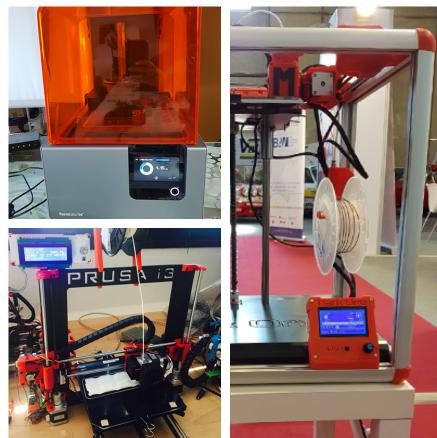


Figura 1.4: Le stampanti 3D in dotazione a Lab Network Srl

Realtà virtuale e aumentata

L'azienda utilizza software e strumenti per lo studio e lo sviluppo in ambito della realtà virtuale ed aumentata. Nel dettaglio, è stato utilizzato il motore grafico *Unreal Engine* per lo sviluppo dell'ambiente virtuale relativo al Vitruvian Game: per utilizzare e testare tale ambiente, Lab Network Srl utilizza il visore *Vive* prodotto da *HTC*.

L'azienda, in ambito di realtà aumentata, utilizza la piattaforma *Unity* per lo sviluppo di nuove applicazioni.



Figura 1.5: L’ambiente virtuale realizzato con Unreal Engine durante la fase di sviluppo

Tecnologie di supporto

A supporto dei processi di sviluppo, l’azienda utilizza la piattaforma web di *GitLab*, che consente la gestione di repository Git e di funzioni trouble ticket. L’utilizzo di questa piattaforma permette il controllo della configurazione e del versionamento di un software, oltre che la gestione di ticket. Come detto in precedenza, la piattaforma viene utilizzata anche per la gestione del Product Backlog.

Sviluppo web

Alcuni progetti richiedono lo sviluppo di applicazioni web: nel caso di FabKey è stato realizzato un software per la gestione delle autorizzazioni e il controllo degli accessi nei determinati varchi.

Tale applicazione web è stata sviluppata utilizzando PHP e MySQL per la parte backend, mentre HTML, CSS e JavaScript per la parte frontend.

Deep Learning e intelligenza artificiale

Intel Movidius è un prodotto di apprendimento profondo che permette di far girare reti neurali profonde in tempo reale direttamente dal dispositivo, consentendo di svolgere un’ampia gamma di applicazioni IA offline.

Utilizzando questa tecnologia, Lab Network Srl sta attualmente sviluppando un progetto che consiste nella previsione di futuri incidenti e infortuni sul lavoro.

Sfruttando le tecniche di *deep learning* messe a disposizione dall’hardware e analizzando grandi dati riguardanti incidenti passati, si è in grado di effettuare una stima sui possibili incidenti futuri.

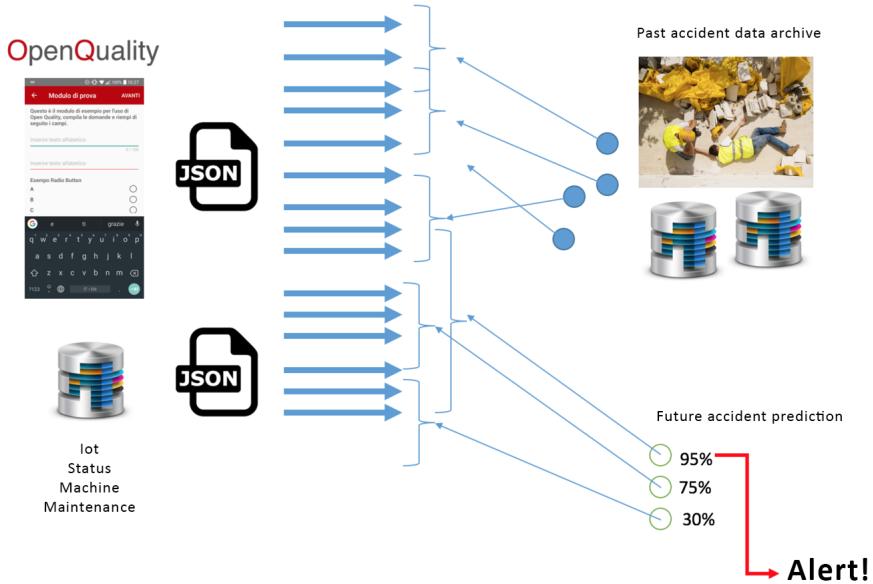


Figura 1.6: Previsione degli infortuni sul lavoro: schema di funzionamento

1.4 Lab Network e innovazione

1.4.1 Pensiero aziendale

Lab Network Srl si è sviluppata nell’ambito della *smart specialisation* attraverso logiche di coinvolgimento di una comunità di utenti misti che provengono prevalentemente dal mondo aziendale e accademico. L’azienda si propone come punto di riferimento per l’innovazione *Open Source* del Veneto, promuovendosi come centro privilegiato di interscambio di conoscenza.

Come previsto dalla Smart Specialisation Strategy, è stata condotta una prima fase di analisi dall’azienda ed è emerso che il territorio regionale è composto principalmente da piccole e medie imprese di tipo manifatturiero. Sulla base di ciò, Lab Network Srl si è posta l’obiettivo di contribuire, attraverso strumenti e modalità di funzionamento specifici, a sviluppare ed attuare la strategia regionale della “Fabbrica Intelligente Del Futuro”.

Questa mira ad indirizzare la trasformazione del settore manifatturiero verso nuovi prodotti, processi e tecnologie, attraverso lo sviluppo di attività di ricerca di alto livello.

Per questo Lab Network Srl è sempre alla ricerca di nuove tecnologie, in modo da essere sempre aggiornata

1.4.2 Smart Specialisation Strategy



Figura 1.7: Punti chiave della Smart Specialisation Strategy. Video "The Kingdom of Smart"

Fonte: <http://s3platform.jrc.ec.europa.eu/home>

“L’innovazione non può essere dettata, ma può essere coltivata attraverso scoperte imprenditoriali; ciò richiede una leadership, un impegno comune sostenuto tramite collaborazione e investimenti orientati al futuro.”

— Pubblicità progresso, Commissione Europea - 2013

La *Smart Specialisation Strategy* (S.S.S.) è una strategia concepita nell’ambito della politica di coesione riformata dalla Commissione Europea per incentivare l’innovazione regionale al fine di ottenere una crescita economica, permettendo alle regioni di focalizzare i loro punti di forza.

Delinea delle strategie di innovazione concepite a livello regionale ma messe a sistema a livello nazionale con l’obiettivo di:

- sviluppare strategie di innovazione a livello regionale, mirate a valorizzare ambiti produttivi di eccellenza, considerando il posizionamento strategico all’interno del territorio e le prospettive di sviluppo in un quadro economico globale;
- aumentare il livello di conoscenza delle Regioni in ambito tecnologico e su settori prioritari;
- migliorare il modo in cui gli interventi vengono gestiti e governati e aumentare l’efficacia delle attività di valutazione e monitoraggio dei risultati.

I passaggi da seguire per attuare questa strategia sono:

- Analizzare cos'è unico, originale, storico;
- Definire e condividere una visione per una regione;
- Definire una priorità ed effettuare una scelta;
- Trovare l'insieme di politiche migliori da implementare;
- Selezionare gli indicatori a cui fare riferimento;
- Istituire una governance;
- Valutare, rifinire e controllare regolarmente.

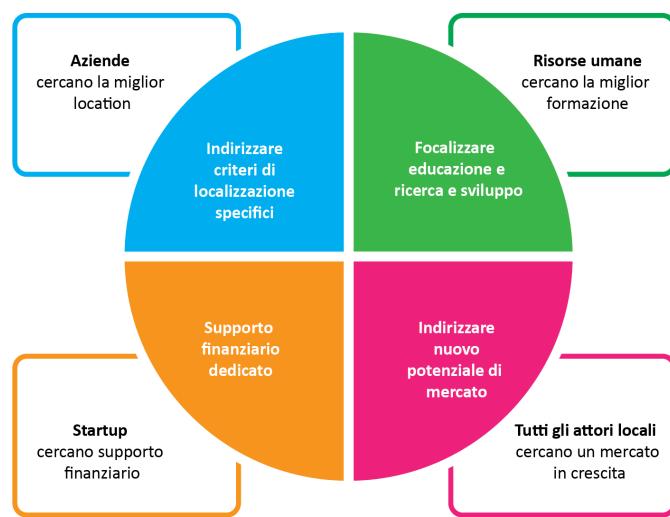


Figura 1.8: Parti coinvolte e ruoli nella Smart Specialisation Strategy.

Fonte:

<https://www.slideshare.net/TCINetwork/8-clac-18-junefrederic-miribel>

L'approccio di Lab Network alla S.S.S.

Lab Network Srl mette a disposizione un portale dedicato dove aziende e fabbricanti possono comunicare scambiando sapere tecnologico e progetti nell'ottica innovativa dell'economia condivisa.

L'analisi SWOT (Strengths, Weakness, Opportunities, Threats) condotta da Lab Network Srl sulla realtà delle PMI venete, evidenzia come ci sia uno scarso utilizzo di tecnologie dell'informazione e della comunicazione (ICT), scarsa disponibilità di laboratori di proprietà, bassi investimenti in ricerca, difficoltà a sviluppare progetti innovativi e ridotta capacità di reperire risorse e professionalità necessarie. L'obiettivo di Lab Network Srl è quello di colmare queste lacune mettendo a disposizione alle aziende un'area di lavoro a costi contenuti dove poter utilizzare i principali strumenti di innovazione digitale come: stampanti 3D, kit elettronici, macchine a controllo numerico, frese digitali e taglio laser. Tutto questo può essere definito come un laboratorio per studiare la fase prototipale.

In pratica Lab Network Srl vuole contribuire a portare l'innovazione e la tecnologia all'interno delle PMI che ancora non si sono interfacciate a questo "nuovo mondo".

Capitolo 2

Il progetto nella strategia aziendale

2.1 Azienda e stage

Lab Network Srl è solita ospitare tirocinanti e stagisti per lunghi o brevi periodi e da quest'anno ha stipulato l'accordo con l'Università di Padova che consente all'azienda di ospitare anche studenti universitari.

Le passate esperienze con altri studenti consistevano principalmente nella creazione di siti web, perché il tempo a disposizione era inferiore a quello a me concesso. Ad ogni modo, nonostante il carico di lavoro fosse minore, sono stati progetti soddisfacenti, portando un'esperienza formativa ai tirocinanti e all'azienda un prodotto finito da consegnare al cliente.

Ogni progetto di stage viene sviluppato sulla base delle aspettative e delle conoscenze attuali di ogni tirocinante, in modo da rendere l'esperienza la migliore possibile.

Ho avuto modo di parlare con alcuni studenti che in passato hanno svolto un progetto di stage presso Lab Network Srl e ho raccolto testimonianze positive: al di là del progetto stesso, i risultati finali sono un arricchimento conoscitivo e curriculare e, talvolta, queste esperienze sfociano in assunzioni.

Analizzando le precedenti esperienze di stage in azienda, è emerso che la maggior parte delle persone trovano un impiego entro un mese dallo svolgimento del tirocinio.

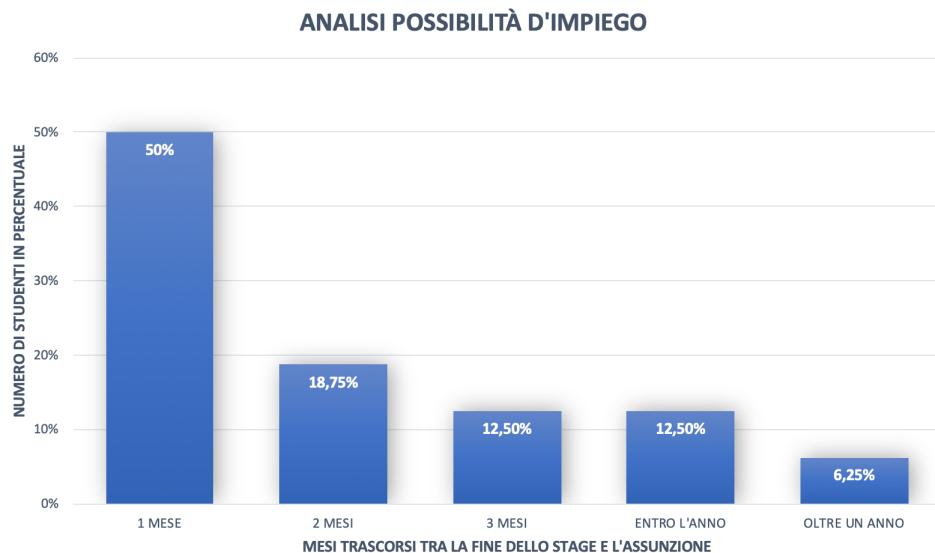


Figura 2.1: Dati riportanti il tempo di assunzione dopo lo stage. In percentuale sull'asse delle ordinate sono riportati gli studenti che hanno effettuato uno stage presso Lab Network Srl; sull'asse delle ascisse i mesi trascorsi tra il termine dello stage e l'impiego in un'azienda.

Fonte: I dati sono stati forniti da Lab Network Srl sulla base di un'analisi effettuata negli ultimi 3 anni di esercizio.

I dati riportati nel grafico riguardano gli stage di ambito informatico effettuati presso Lab Network Srl negli ultimi 3 anni e il tempo di impiego indica l'assunzione presso un'azienda che lavora nel settore ICT.

Il tirocinio è vantaggioso sia per l'azienda che per lo stagista:

- **Dal punto di vista aziendale:** ospitare uno studente, o in generale una persona, per un periodo all'interno di un progetto di stage, consente all'azienda di valutare le capacità lavorative dello stagista e, se quest'ultimo risulta valido, considerare il tirocinio come un periodo di formazione finalizzato all'assunzione.

Lo stage, inoltre, consente di portare a compimento alcuni progetti non troppo complessi, catalogati come secondari, che fino a quel momento non avevano trovato spazio all'interno dell'azienda.

- **Dal punto di vista dello studente:** uno stage è un'ottima opportunità per mettere in pratica quanto appreso durante gli studi; inoltre, spesso consente di accrescere la propria formazione, perché si presenta la necessità dell'utilizzo di nuove tecnologie e nuovi strumenti di lavoro.

Sono venuto a conoscenza di Lab Network grazie alla diffusione mediatica di alcuni loro progetti che stavano avendo successo, come ad esempio Vitruvian Game, e così ho deciso di proporli all'azienda per lo svolgimento del tirocinio: quest'ultima ha accettato proponendomi alcuni progetti disponibili tra cui FabKey.

FabKey è stato proposto come progetto di stage, in quanto era stato fino ad allora catalogato come secondario e non aveva ancora trovato il suo spazio all'interno dell'azienda. Per di più, coinvolgere uno studente universitario nello sviluppo del progetto, sarebbe stato vantaggioso sfruttando le capacità che lo stagista ha sviluppato durante il corso di studi.

2.2 Obiettivi aziendali

L'obiettivo principale del progetto di stage è l'ampliamento del già collaudato sistema "FabKey", il quale permette l'apertura di una porta attraverso un tag NFC controllando una lista di accessi presente in un database online. Nello specifico, tale sistema doveva essere revisionato e ampliato, utilizzando un modulo che andrà ad autenticare l'utente tramite codice a barre anziché tag NFC.

Gli obiettivi concordati con il tutor aziendale sono stati classificati secondo tre gradi di priorità:

1. **Obbligatori:** obiettivi il cui sviluppo è necessario per la riuscita del progetto;
 - **Integrazione di un sistema completo per l'apertura di serrature con lettura di codice a barre e NFC.** L'obiettivo principale richiesto dall'azienda è quello di ampliare il sistema esistente, che consiste in una centralina di controllo e di un modulo NFC, con un modulo per la lettura di codici a barre. Si rende necessaria la riprogrammazione della centralina, aggiungendo la nuova parte di codice per la gestione dei codici a barre;
 - **Realizzazione della piattaforma web per la gestione degli accessi.** La piattaforma web per il controllo degli accessi è uno strumento indispensabile per l'utilizzo del sistema FabKey, pertanto la sua realizzazione è un requisito obbligatorio. Questa applicazione web sarà composta da un pannello di controllo che permette l'utente di modificare le autorizzazioni d'accesso a uno o più varchi e di consultare un report contenente tutti gli accessi effettuati;
 - **Creazione del modello 3D dell'involucro e sua realizzazione con stampa 3D.** Durante la definizione degli obiettivi è stato accordato che tutta la parte elettronica sarà inserita in un involucro disegnato su misura e stampato in 3D;
 - **Redazione della manualistica completa.** A corredo del kit, all'utente finale sarà consegnata tutta la manualistica necessaria all'installazione e all'uso. Sono necessari anche dei manuali interni per la guida all'assemblaggio, alla codifica e alla corretta configurazione per il cliente finale;
2. **Desiderabili:** il loro sviluppo non è necessario ai fini del progetto, ma forniscono un valore aggiunto considerevole;

- **Cura e definizione dell’interfaccia grafica della piattaforma web.** Seppur trattandosi di un aspetto non vincolante ai fini del progetto, una grafica curata accresce in modo significativo il valore del prodotto finale;
 - **Ottimizzazione del sistema esistente in termini di efficienza e prestazioni.** Con questo obiettivo si vogliono ridurre i tempi di risposta dell’intero sistema, ottimizzando l’esecuzione del codice in generale e in particolare le chiamate HTTP al server;
3. **Facoltativi:** il loro sviluppo diventa apprezzabile, ma dal valore aggiunto trascurabile.
- **Creazione di un modello 3D modulare espandibile per future versioni.** In previsione di future espansioni con l’utilizzo di elettronica differente, è apprezzabile la creazione del modello 3D in modo modulare e universale, compatibile quindi con eventuali modifiche future.

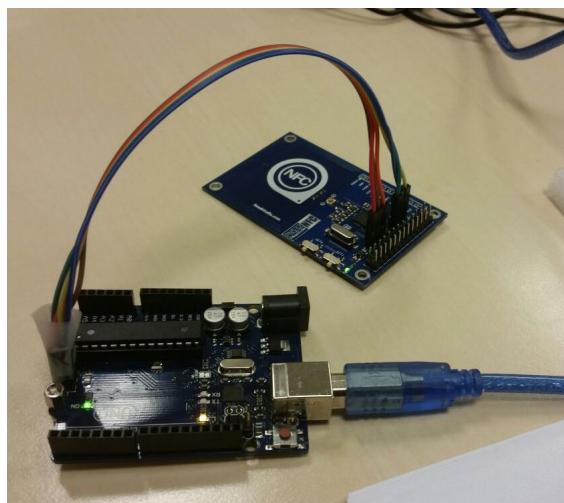


Figura 2.2: Prototipo funzionante di FabKey prima dell’inizio dello stage.

La realizzazione di questo progetto può portare diversi vantaggi a Lab Network Srl tra cui:

- Grazie all’immissione sul mercato di FabKey, l’azienda ha l’opportunità di entrare nel mercato consumer, ampliando così il tipo di clientela. Data la semplicità di installazione e configurazione, il prodotto finito potrà essere venduto anche a clienti privati;
- La fornitura prevista a tutti i FabLab del territorio fa sì che il prodotto si diffonda sin dalla prima immissione sul mercato.



Figura 2.3: Il prodotto finito dopo il termine dello stage.

2.3 Vincoli

L’azienda mi ha imposto dei vincoli in termini di tempo e tecnologie, al fine di un corretto svolgimento del progetto oltre che una buona integrazione con il resto del team di sviluppo.

Ho provveduto successivamente alla redazione di un piano di lavoro, in collaborazione con il tutor aziendale, per riportare quanto stabilito a monte dello stage.

I vincoli tecnologici sono:

- **GitLab:** si tratta di una piattaforma web per la gestione di repository Git. L’azienda mi ha richiesto di utilizzare questo strumento anche per la gestione di ticket e del Product Backlog;
- **Arduino IDE:** applicazione che consente la programmazione di schede Arduino e compatibili. Viene fornito con un editor di testo, un compilatore e una serie di esempi per ogni libreria disponibile. Utilizzando schede Arduino per la gestione dei moduli NFC e barcode, risulta necessario l’utilizzo di questo ambiente;
- **Rhinoceros:** ho ricevuto disposizioni da parte dell’azienda, dopo un’adeguata documentazione, di utilizzare questo software per la modellazione dell’involturo contenente l’elettronica di FabKey.

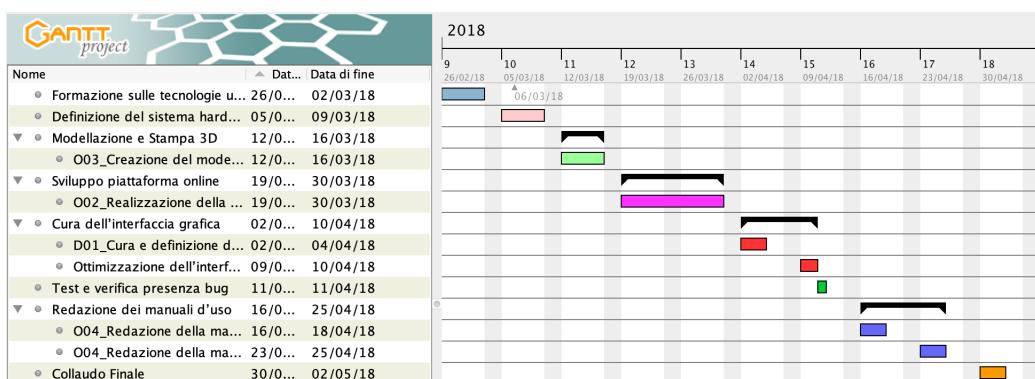


Figura 2.4: Diagramma di Gantt per la pianificazione del progetto di stage.

Il tempo a disposizione per lo svolgimento del tirocinio è di almeno 300 ore, le quali sono state suddivise in 40 ore settimanali per le prime 5 settimane e le rimanenti ore suddivise su una base di 24 ore settimanali.

Al termine dello stage, ho tralasciato lo sviluppo di alcuni degli obiettivi tra i desiderabili e i facoltativi principalmente per mancanza di tempo. Per migliorare il prodotto, ho ricevuto la proposta da parte del tutor di continuare a lavorare nella sua azienda.

2.4 Obiettivi personali

Le motivazioni che mi hanno spinto a scegliere questo progetto presso Lab Network Srl sono:

- **Codifica e correlazione tra hardware e software:** un aspetto dell'informatica che mi ha da sempre affascinato è l'interazione tra hardware e software; non avendo potuto approfondire l'argomento durante gli studi, ho trovato questa proposta di stage un'ottima opportunità per studiare sul campo la programmazione hardware. Il progetto, inoltre, si affaccia al mondo dell'*Internet Of Things* che è in continua crescita;
- **Modellazione e stampa 3D:** è un campo molto diffuso e in continua crescita. Con questo progetto di stage mi è stata offerta la possibilità di usare in esclusiva una stampante 3D per la realizzazione dei primi prototipi dopo averli correttamente modellati;
- **Lavoro in team:** fare parte di un team di sviluppo cogliendo gli aspetti in comune e le differenze tra un progetto accademico come quello svolto durante il corso di Ingegneria del Software e un progetto in ambito lavorativo.

Le mie aspettative nello svolgimento di questo progetto di stage erano principalmente:

- **Formazione lavorativa in ambito software:** il mio interesse era di poter vivere un'esperienza lavorativa presso un'azienda del settore ICT, mettendola a confronto con i miei precedenti impieghi presso aziende di altri settori;
- **Accrescere le mie conoscenze:** apprendere velocemente sul campo sia mettendo in pratica quanto appreso durante di studi che utilizzando nuove tecnologie e nuovi strumenti.

Capitolo 3

Lo stage

3.1 Pianificazione

Prima dell'inizio del progetto, ho redatto assieme al tutor aziendale un piano di lavoro. All'interno di questo ho riportato gli obiettivi individuati, la pianificazione del lavoro sotto forma di processi, suddivisi a loro volta in attività. Ad ognuna di queste sono state assegnate delle ore di lavoro, la cui somma sarà il totale delle ore a disposizione per lo stage.

Le attività individuate sono:

- **Formazione sulle tecnologie utilizzate:** la fase iniziale dello stage consiste nella formazione sulle tecnologie usate, oggetto dei vincoli posti dall'azienda. Questa è stata una fase molto importante per poter lavorare correttamente ed in sincronia con il resto del team;
- **Definizione del sistema hardware/software e relativa documentazione**
 - **Analisi del problema e del dominio applicativo:** la prima attività da svolgere è stata quella di analizzare il problema e il contesto in cui il prodotto dovrà operare. Questa attività include anche lo studio del sistema esistente in modo da comprenderne in modo esaustivo il funzionamento. Come prodotto in uscita ho ottenuto tutti i requisiti relativi alla parte hardware del progetto;
 - **Adattamento e revisione della piattaforma esistente:** una volta compreso il funzionamento del sistema esistente e formalizzato i requisiti ottenuti dall'attività precedente, ho progettato la modifica da apportare per poter lavorare con il nuovo modulo ed in seguito ho riportato il tutto su codice;
 - **Test piattaforma hardware:** ho pianificato ed eseguito alcuni test d'unità che andassero a verificare la correttezza sulle nuove parti di codice;
 - **Stesura documentazione:** per poter garantire un'adeguata manutenzione al prodotto, ho redatto in questa attività tutta la documentazione relativa alle nuove modifiche;

- **Modellazione e Stampa 3D**

- **Design e modellazione involucro:** ho dedicato del tempo per la creazione digitale del modello tridimensionale per l'involucro che conterrà l'elettronica del nuovo modulo aggiunto;
- **Slicing del modello e stampa 3D:** ho preparato il modello creato nell'attività precedente per la stampa andando ad impostare vari parametri al computer ed infine ho provveduto alla stampa vera e propria;

- **Sviluppo piattaforma online**

- **Pianificazione e analisi dei requisiti:** ho pianificato e analizzato i requisiti per lo sviluppo della piattaforma online. Questo strumento serve al cliente finale per monitorare i vari accessi al varco controllato da FabKey e per gestire gli accessi autorizzati;
- **Sviluppo della piattaforma con iniziale attenzione al lato server:** a seguito della pianificazione e raccolta dei requisiti, ho proseguito con lo sviluppo della piattaforma, con priorità al lato server;

- **Cura dell'interfaccia grafica:** ho prestato attenzione alla cura dell'interfaccia grafica relativa alla piattaforma creata nelle attività precedenti. Durante questa attività ho curato anche la parte front-end dell'applicazione online;
- **Test e verifica presenza bug:** al termine della fase di sviluppo ho eseguito dei test sul sistema completo;
- **Redazione dei manuali d'uso:** in questa attività ho redatto tutti i manuali d'uso sia per l'utente (uso e installazione del sistema), sia per uso interno (assemblaggio e configurazione del sistema);

- **Collaudo Finale**

- **Collaudo:** collaudo finale prima della presentazione ufficiale;

Ho riportato gli obiettivi individuati sul piano di lavoro, categorizzandoli secondo tre tipologie come spiegato nella sezione 2.2

3.2 Studio iniziale

3.2.1 Documentazione individuale

Gli obiettivi e i vincoli fissati comprendevano alcune tecnologie e strumenti che ancora non facevano parte del mio bagaglio formativo, pertanto lo stage ha avuto inizio con una fase di studio individuale. Supportato dal tutor aziendale e dai colleghi, ho iniziato lo studio degli strumenti che l'azienda usa regolarmente durante lo sviluppo di un progetto, come GitLab, la piattaforma online per repository Git. In precedenza, durante alcuni progetti universitari, tra i quali quello di Ingegneria del Software, ho utilizzato il software di versionamento Git tramite la piattaforma

di GitHub, simile per molti aspetti a GitLab. Per questo l'apprendimento non è stato difficile.

Ha richiesto più tempo lo studio del software di modellazione 3D Rhinoceros. Non essendomi mai approcciato al disegno e modellazione tridimensionale, apprendere le basi è stata un'operazione più lunga del previsto, ma, una volta compreso il principio alla base, la creazione di modelli semplici è un processo abbastanza intuitivo.

Ho avuto la possibilità di studiare il funzionamento della stampa 3D grazie alla disponibilità di una stampante a mio uso esclusivo. Ciò mi ha permesso di effettuare diversi test modificando i parametri di stampa e sperimentando diversi materiali, il tutto prima dello sviluppo del progetto.

L'ultimo strumento oggetto di studio è stato Arduino. Ho iniziato studiando l'editor e le modalità di collegamento alla scheda. Prima di iniziare con dei test di funzionamento, ho preferito utilizzare il software Fritzing¹ per disegnare gli schemi di collegamento dei componenti elettronici alla scheda; questo strumento si è reso utile anche durante la redazione dei manuali ad uso interno per la guida all'assemblaggio.

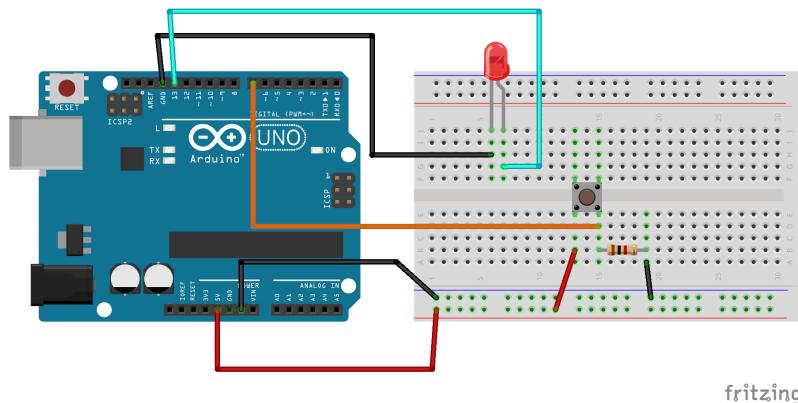


Figura 3.1: Esempio di schema creato con Fritzing. Rappresenta uno dei primi test effettuati con scheda Arduino.

Lo sviluppo del progetto è partito da uno studio preliminare sul sistema già esistente composto da una centralina di controllo ed un lettore di schede NFC. Scendendo nei dettagli, la fase di studio è iniziata dalla centralina, composta da una scheda Arduino. Per poterne apprendere il funzionamento, il tutor aziendale mi ha fornito una scheda di test sulla quale eseguire del codice di esempio con dei semplici circuiti elettronici. L'apprendimento si è rivelato piuttosto rapido anche grazie alla vasta disponibilità di materiale online e ad una consolidata community.

Una volta appreso il funzionamento generale di Arduino, ho iniziato lo studio sul codice che governa il comportamento della centralina di controllo, caratterizzato principalmente da 3 fasi:

- **Configurazione iniziale:** il sistema esegue una configurazione iniziale in modo da integrarsi perfettamente nella rete alla quale è collegato;

¹<http://fritzing.org/>.

- **Listener:** una porzione di codice che rimane in continuo ascolto per catturare gli eventi derivanti dai moduli collegati;
- **Crittografia e HTTP request:** una funzione che si occupa della cifratura dei codici letti e del loro invio tramite richieste HTTP ad un server.

Terminata questa prima fase di studio, è susseguita quella di analisi.

3.3 Analisi dei requisiti

3.3.1 Identificazione

Per poter identificare correttamente tutti i requisiti, a seguito di vari colloqui con il tutor aziendale e il cliente, ho definito i vari casi d'uso, disegnandone in seguito i rispettivi diagrammi UML. Ogni riunione è stata riportata su di un verbale e da quest'ultimo ho in seguito ricavato i vari requisiti. Ho anche analizzato approfonditamente i casi d'uso in modo da poter identificare ulteriori requisiti. Una volta individuati tutti i requisiti sono stati riportati in un documento soggetto a versionamento.

3.3.2 Strumenti a supporto

Casi d'uso

I casi d'uso e i relativi diagrammi UML sono stati un ottimo strumento di supporto durante la fase di analisi: mi hanno permesso di individuare alcuni requisiti non emersi durante le riunioni con tutor e stakeholders.

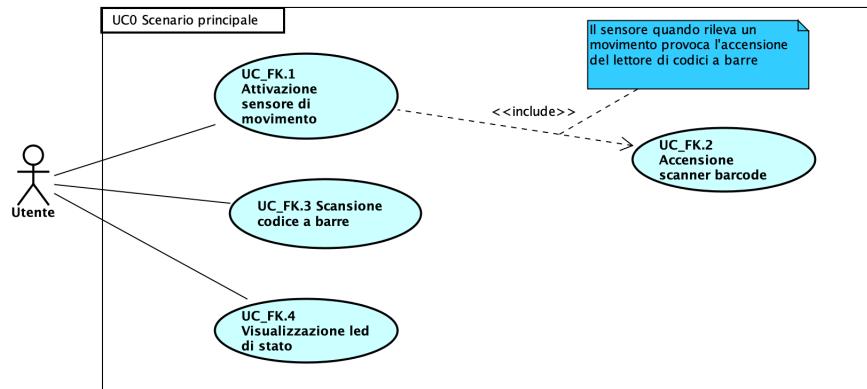


Figura 3.2: Estratto dello schema UML rappresentante i casi d'uso dello scenario principale. Schema riferito all'uso di FabKey e non alla piattaforma online.

Essendo il progetto composto da due sistemi, FabKey e piattaforma online, i casi d'uso sono stati categorizzati da una notazione univoca, che facilita l'identificazione del caso d'uso nel contesto:

UC_[Contesto].[IDUnivoco]

- **Contesto:**
 - **FK:** FabKey;
 - **PO:** Piattaforma online di gestione.
- **IDUnivoco:** Un codice numerico progressivo che identifica in maniera univoca il caso d'uso a cui si riferisce. Questo codice può essere gerarchico in modo da identificare casi d'uso innestati.

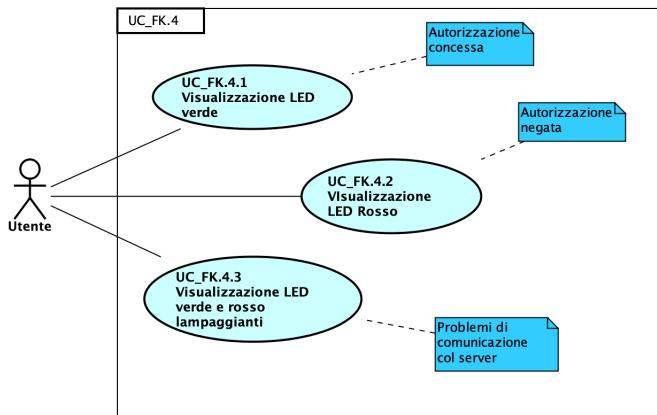


Figura 3.3: Dettaglio del caso d'uso UC_FK.4. Si può notare l'uso della notazione sopra descritta.

Ogni caso d'uso ha una specifica provenienza, ovvero alcune determinate fonti che lo hanno generato. Queste sono:

- **Verbali:** ottenuti trascrivendo quanto emerso durante i colloqui con i portatori di interesse;
- **Le specifiche del sistema:** descrivono quali sono le caratteristiche che il sistema deve avere;
- **Conoscenza del dominio:** la conoscenza del contesto d'utilizzo e la consultazione di materiale ad esso inerente.

Ho utilizzato queste fonti anche per l'identificazione dei requisiti.

3.3.3 Formalizzazione dei requisiti

Dopo aver individuato tutti i requisiti necessari allo sviluppo del progetto, ho creato una codifica che li potesse identificare in maniera univoca e allo stesso tempo catalogare per tipo e priorità. La codifica utilizzata è la seguente:

R[Componente]_[Tipologia]_[Priorità][Numero Identificativo]

Componente

- **FK:** requisito inerente allo sviluppo della parte elettronica di FabKey;
- **PO:** requisito relativo allo sviluppo del portale online per la gestione degli accessi.

Tipologia

- **F:** un requisito di tipo funzionale, cioè che aggiunge funzionalità al sistema;
- **Q:** requisito di qualità, ovvero che, se soddisfatto, aumenta la qualità generale del prodotto;
- **P:** requisito prestazionale, mirato a migliorare il prodotto in termini di prestazioni.

Priorità

- **Ob:** requisiti obbligatori, il cui sviluppo è condizione necessaria per ottenere il risultato finale;
- **De:** requisiti desiderabili, ovvero soddisfarli significa apportare importanti miglioramenti al prodotto anche se non strettamente vincolanti al suo funzionamento;
- **Op:** requisiti opzionali, che apportano valore aggiunto al sistema, ma il cui sviluppo è trascurabile.

Numero Identificativo

Un codice numerico progressivo che identifica in maniera univoca il requisito a cui si riferisce. Questo codice può essere gerarchico.

Riporto di seguito alcuni dei requisiti individuati:

Hardware FabKey

Codice	Tipo	Descrizione
RFK_F_Ob1	Funzionale Obbligatorio	Il sistema deve permettere la lettura di un codice a barre
RFK_F_Ob2	Funzionale Obbligatorio	Il sistema deve attivare il lettore quando rileva un utente nelle vicinanze
RFK_Q_Ob1	Qualità Obbligatorio	Il modulo di lettura deve essere contenuto in un involucro stampato in 3D
RFK_Q_Op1	Qualità Opzionale	L'involucro esterno deve essere modulare ed espandibile per versioni successive

Tabella 3.1: Alcuni requisiti individuati per lo sviluppo della parte hardware di FabKey.

Portale online

Codice	Tipo	Descrizione
RPO_F_Ob1	Funzionale Obbligatorio	L'applicazione deve permettere di visualizzare l'elenco degli utenti autorizzati
RPO_F_Ob2	Funzionale Obbligatorio	L'applicazione deve permettere di gestire le autorizzazioni
RPO_F_Ob2.1	Funzionale Obbligatorio	L'applicazione deve permettere di eliminare un'autorizzazione
RPO_F_De1	Funzionale Desiderabile	L'applicazione deve presentare un'interfaccia grafica completamente scalabile

Tabella 3.2: Alcuni requisiti individuati per lo sviluppo del gestionale online di FabKey.

Dopo aver individuato e codificato tutti i requisiti, ho composto una tabella in cui ho riportato il tracciamento tra i requisiti e i casi d'uso, in modo da identificare in quale caso d'uso viene soddisfatto un determinato requisito.

3.4 Progettazione

3.4.1 L'approccio iniziale

Ho gestito la progettazione del prodotto in modo separato rispetto alle due componenti principali che ne fanno parte: l'hardware di FabKey (composto da centralina e lettore) e l'applicazione online per la gestione delle autorizzazioni. Per comodità chiamerò queste due componenti **HWFabKey** e **Gestionale accessi** rispettivamente.

La fase di progettazione è stata molto importante, in quanto la sua corretta realizzazione mi avrebbe permesso una fase di codifica rapida minimizzando i rischi di errore.

3.4.2 Progettazione di HWFabKey

Le componenti alle quali dovevo prestare particolare attenzione durante la fase di progettazione erano quattro:

- **Gestione delle componenti elettroniche che compongono il lettore di codice a barre:** durante la progettazione ho ideato lo schema per i collegamenti del lettore di codici a barre (composto da diverse componenti

elettroniche), prestando attenzione a non creare interferenze con altre componenti già collegate nella versione esistente, come ad esempio il circuito apri porta;

- **Sicurezza nella trasmissione dei dati:** i dati trasmessi verso il server andavano crittografati per evitare di incorrere in alcuni problemi di sicurezza;
- **Protocollo di comunicazione con il server:** dovevo scegliere in questa fase il modo in cui la centralina avrebbe comunicato con il server, analizzando pro e contro delle varie alternative.
- **Creazione e stampa 3D dell'involucro:** dopo aver abbozzato su carta alcune ipotesi e averne discusso le caratteristiche con il tutor, ho disegnato e stampato in 3D l'involucro;

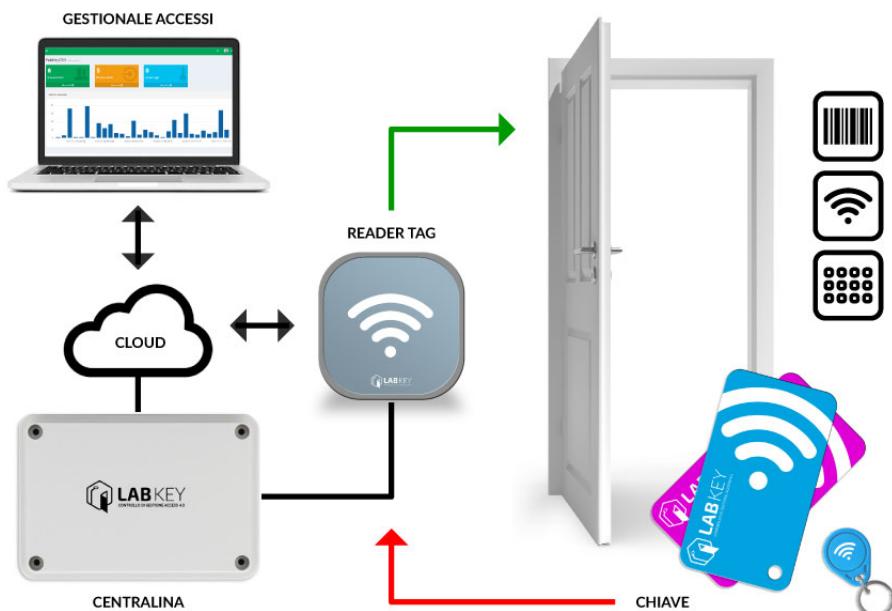


Figura 3.4: Schema di funzionamento di FabKey.

Fonte: <https://www.labkey.it>

Gestione dell'elettronica

L'utilizzo del nuovo modulo per la lettura di codici a barre ha comportato la gestione di 4 nuove componenti elettroniche, le quali andavano collegate ad Arduino senza interferire con le componenti base del sistema. Ho risolto tutto questo disegnando lo schema completo ed aiutandomi con un simulatore.

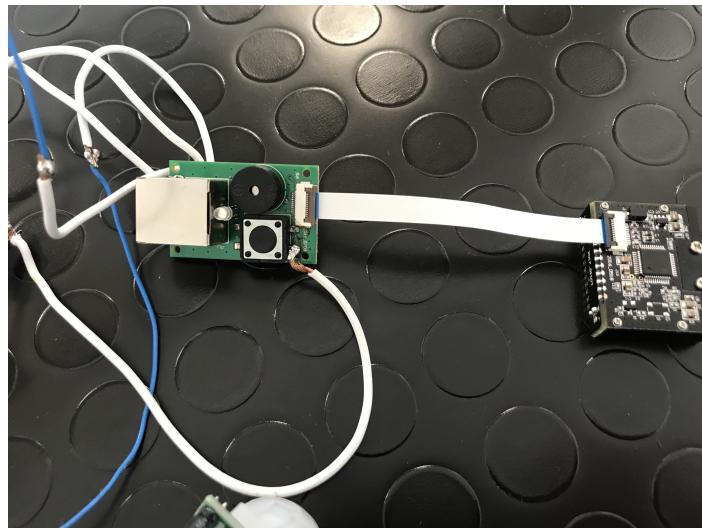


Figura 3.5: Una delle componenti elettroniche che compongono il lettore di codici a barre. Si tratta del vero e proprio lettore ottico.

Sicurezza nella trasmissione dati

Il maggior problema che ho riscontrato nella gestione della trasmissione dei dati è stato quello riguardante la sicurezza. A primo impatto ho pensato all'uso di una codifica con funzioni crittografiche hash, come ad esempio MD5 o SHA-1, ma non avevo preso in considerazione un aspetto molto importante: la memoria disponibile.

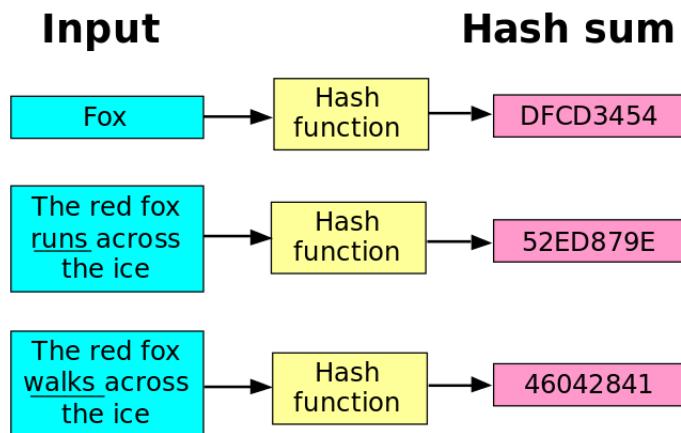


Figura 3.6: Esempio di funzione hash. Risultato dei primi quattro byte della funzione hash SHA-1.

Fonte: <https://it.wikipedia.org/wiki/Hash>

Nell'hardware a disposizione le risorse erano estremamente limitate: avevo tralasciato questo aspetto dal momento che ho sempre lavorato con dispositivi la cui potenza di calcolo era trascurabile per questo tipo di operazioni. Nel caso specifico, 2 KB di memoria SRAM non erano sufficienti per la gestione dell'intero programma

di controllo e contemporaneamente memorizzare una variabile di 40 byte o più, contenente la stringa con codifica hash. Per cui, collaborando con un collega, abbiamo ideato un algoritmo di crittografia semplificato, basato sul principio della funzione hash, ma in grado di produrre un output nettamente ridotto.

Protocollo di comunicazione

Uno dei requisiti prevedeva lo scambio di informazioni tra la centralina di controllo, connessa in rete tramite cavo LAN, e il server centrale, contenente il database con gli accessi autorizzati. Il compito del team era di soddisfare questo requisito in modo che la comunicazione fosse il più rapida possibile. La soluzione adottata consiste in una semplice chiamata HTTP verso il server: viene creato un *URL* composto dall'indirizzo del server e il codice di accesso richiesto, codificato tramite l'algoritmo sopra descritto; il server, ricevuta la richiesta, ricava dall'*URL* il codice crittografato e lo confronta con i dati presenti nel database, rispondendo con un segnale positivo o negativo.

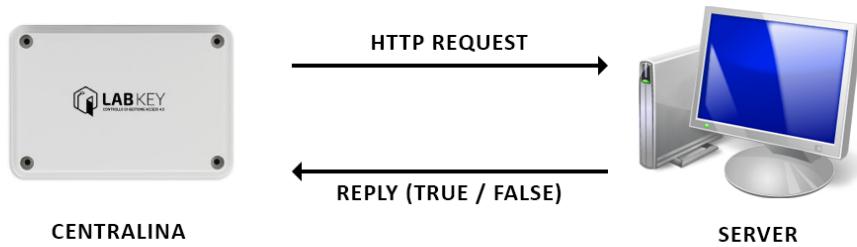


Figura 3.7: Schema di comunicazione Centralina/Server.

Progettazione dell'involucro e stampa 3D

L'idea era di realizzare un involucro in grado di contenere tutte le componenti elettroniche del lettore e che fosse allo stesso tempo il più compatto possibile. Per prima cosa ho abbozzato dei disegni con alcune ipotesi sulle forme generali e, una volta approvato dal tutor, ho iniziato la creazione del modello tridimensionale.

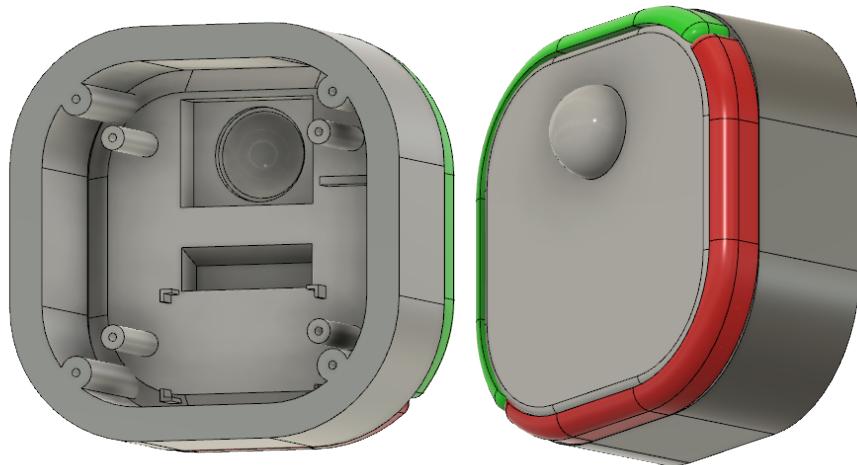


Figura 3.8: La prima versione dell’involturo per il lettore di codici a barre.

Una volta ottenuto il modello finito, ho configurato il processo di stampa grazie al software open source Cura di Ultimaker. Grazie a questa applicazione è possibile, dopo aver impostato i diversi parametri, simulare la stampa, in modo da verificare eventuali errori di configurazione o sul modello.

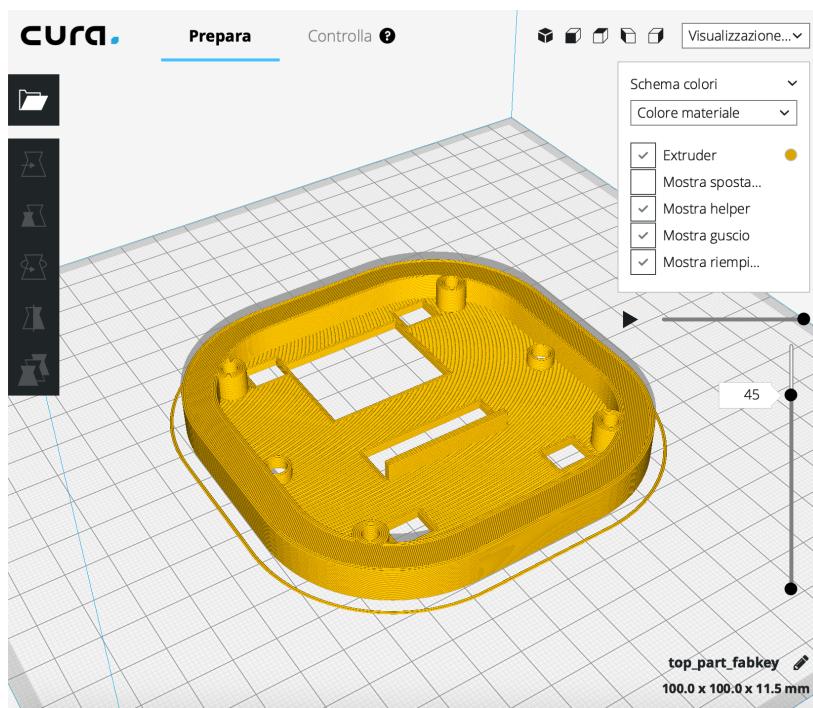


Figura 3.9: Un componente dell’involturo durante la simulazione con il software Cura.

3.4.3 Progettazione del gestionale accessi

L’applicazione web per la gestione delle autorizzazioni è composta da un lato front-end e da un lato back-end. La prima parte è formata dall’interfaccia grafica e dalle

componenti lato client, mentre la seconda contiene il database e le funzioni per l'elaborazione della risposta alle chiamate http da parte della centralina.

Front-end - Interfaccia grafica

Facendo riferimento ai casi d'uso, ho iniziato a progettare l'interfaccia grafica della WebApp, definendo le varie sezioni di cui sarà composta. Ho previsto un menù di navigazione laterale contenente i collegamenti a tutte le pagine presenti.

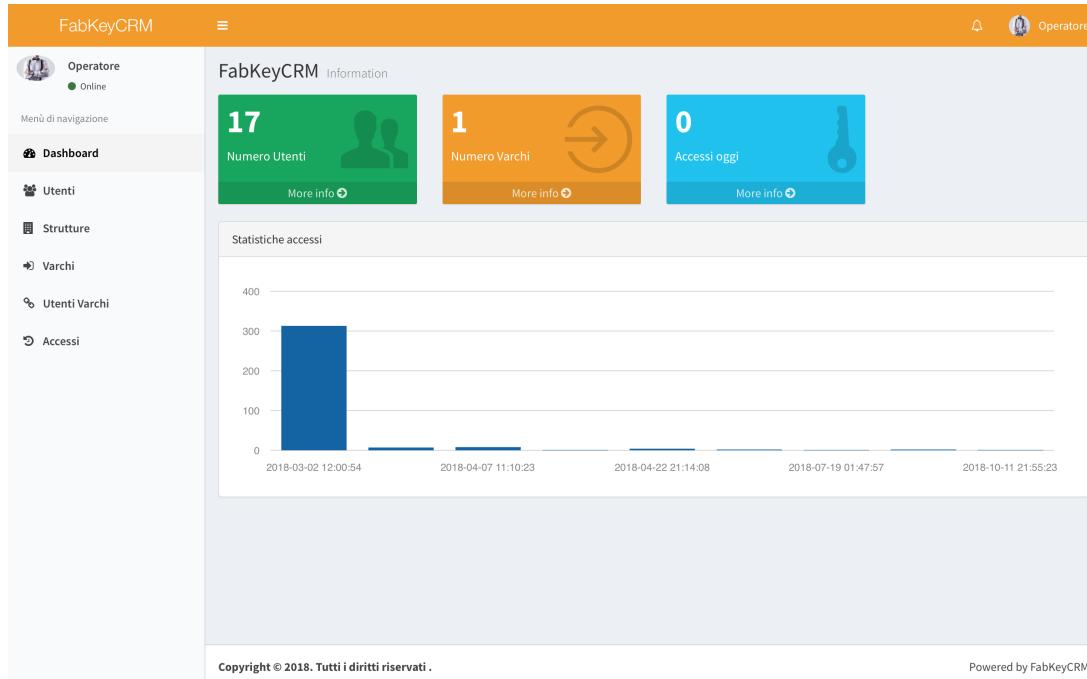


Figura 3.10: Schermata principale del gestionale.

Le sezioni presenti sono:

- **Schermata di login:** per poter accedere all'applicazione è necessario collegarsi al sistema con le proprie credenziali ed è possibile effettuarlo in questa schermata;
- **Dashboard:** rappresenta la pagina iniziale dopo il login; fornisce all'utente alcune informazioni generali sul sistema, ad esempio il numero di utenti attivi, i varchi, gli accessi effettuati in giornata e una statistica sugli accessi;
- **Utenti:** è la pagina che permette la gestione degli utenti con operazioni di inserimento, modifica, eliminazione e assegnazione di una chiave;
- **Strutture:** l'applicazione permette tramite questa pagina la gestione di strutture geograficamente dislocate permettendo di aggiungerne di nuove, eliminare le presenti o modificarle;
- **Varchi:** i varchi rappresentano le porte di una struttura, pertanto ogni varco è associato ad una struttura;

- **Utenti Varchi:** è la sezione che permette di assegnare un utente ad uno o più varchi; in parole povere l'utente può definire quale utente può attraversare determinate porte;
- **Accessi:** viene proposto in questa pagina un log degli accessi effettuati con dettagli su utente, varco e orario dell'accesso.

Utente	Nome	Cognome	Varco	Nome Varco	Data accesso	Azione
8	[REDACTED]	[REDACTED]	1	Entrata Principale	2018-10-11 21:55:23	
18	[REDACTED]	[REDACTED]	1	Entrata Principale	2018-07-26 18:28:53	
18	[REDACTED]	[REDACTED]	1	Entrata Principale	2018-07-26 18:02:53	
8	[REDACTED]	[REDACTED]	1	Entrata Principale	2018-07-19 01:47:57	
8	[REDACTED]	[REDACTED]	1	Entrata Principale	2018-07-18 20:03:05	
8	[REDACTED]	[REDACTED]	1	Entrata Principale	2018-07-18 18:46:22	
12	[REDACTED]	[REDACTED]	1	Entrata Principale	2018-04-22 22:12:58	
12	[REDACTED]	[REDACTED]	1	Entrata Principale	2018-04-22 21:24:48	
12	[REDACTED]	[REDACTED]	1	Entrata Principale	2018-04-22 21:16:39	
12	[REDACTED]	[REDACTED]	1	Entrata Principale	2018-04-22 21:14:08	
8	[REDACTED]	[REDACTED]	1	Entrata Principale	2018-04-19 22:07:48	
7	[REDACTED]	[REDACTED]	1	Entrata Principale	2018-04-07 14:51:24	
7	[REDACTED]	[REDACTED]	1	Entrata Principale	2018-04-07 14:50:37	
7	[REDACTED]	[REDACTED]	1	Entrata Principale	2018-04-07 14:49:47	
7	[REDACTED]	[REDACTED]	1	Entrata Principale	2018-04-07 14:49:32	

Figura 3.11: Schermata del pannello di controllo degli accessi.

Back-end - Database

Per la raccolta dei dati abbiamo scelto di creare un database di tipo relazionale SQL. Ho previsto assieme al team la creazione di tabelle che andassero a rappresentare i vari elementi caratterizzanti del sistema: gli utenti, i varchi, le strutture e alcune tabelle di collegamento per le relazioni di tipo N-N. Essendo una tecnologia a me già nota, è stata relativamente semplice la progettazione logica del database, portata a termine con il supporto di appositi schemi E-R.

Back-end - Funzione di risposta

Parte dell'applicazione è la funzione che gestisce le richieste da parte della centralina. Il principio di funzionamento di questa procedura è relativamente semplice: alla richiesta HTTP della centralina viene estratta la chiave crittografata e il varco dal quale è arrivata la richiesta; successivamente si confrontano queste informazioni con i record presenti nel database tramite una query SQL e viene fornita una risposta positiva o negativa.

3.5 Codifica e realizzazione

3.5.1 Tecnologie e linguaggi

La fase di codifica consiste nell'implementazione di tutte le componenti definite durante la fase di progettazione, applicando le tecnologie selezionate.

Per prima cosa ho iniziato la codifica della centralina Arduino, le tecnologie coinvolte sono state:

- **Arduino IDE e linguaggio C:** per la programmazione del microcontrollore ATMega presente nella scheda Arduino, ho usato l'IDE fornito da Arduino stesso e il linguaggio di programmazione richiesto è il C. Nonostante fossi già ad un buon livello di conoscenza di programmazione con linguaggio C, la trascrizione in codice di quanto progettato non è stata immediata, questo a causa delle diverse librerie da utilizzare, le cui funzioni andavano studiate nel comportamento.
- **Librerie Arduino:** ogni particolare componente elettronico disponeva di una specifica libreria per controllarne il funzionamento. Prima di iniziare la codifica finale ho dovuto studiare le funzioni specifiche di queste librerie che sarei andato ad utilizzare.
 - **SPI:** *Serial Peripheral Interface* è un protocollo per la comunicazione seriale sincrona usata per la comunicazione tra microcontrollori. Nello specifico ho utilizzato le funzionalità di questa libreria per la comunicazione tra Arduino Uno e il lettore di codici a barre;
 - **Ethernet:** la centralina è dotata di un modulo aggiuntivo che la equipaggia di una porta Ethernet. L'apposita libreria per l'utilizzo di questa porta è necessaria per la configurazione di rete;

La traduzione del progetto in codice in questo frangente non è stato per me difficoltoso: ho riscontrato, invece, maggiori problematiche durante la verifica del software, in seguito al caricamento sulla scheda Arduino. Grazie ad alcuni test, ho evidenziato che il codice si comportava saltuariamente in maniera anomala rispetto a quanto previsto, creando un flusso in output apparentemente casuale: ho scoperto in seguito che venivano creati dei caratteri casuali in uscita a causa di interferenze provocate dalla comunicazione seriale con il lettore di codici a barre. Il problema è stato risolto apportando alcune modifiche al software, in modo da ignorare questi caratteri dannosi. I test effettuati sono spiegati nella sezione **3.6.1**.

La seconda parte della codifica è stata incentrata sulla creazione del gestionale e le tecnologie utilizzate sono le seguenti:

- **PHP e Laravel:** per la codifica della componente back-end del gestionale ho utilizzato il linguaggio PHP con il framework Laravel, caratterizzato da un pattern di tipo Model-View-Controller. Per la gestione del Controller ho sfruttato il supporto fornito da CRUDBooster, un CRUD (Create, Read, Update, Delete) generator che, tramite un sistema a “scaffolding”, automatizza la creazione di oggetti e interfacce. Queste per me sono state tecnologie nuove

e hanno richiesto un periodo di studio individuale precedente alla fase di codifica.

- **Bootstrap:** è un framework front-end per la creazione di componenti lato client, integrato nel framework Laravel. Ho utilizzato la struttura messa a disposizione da Bootstrap per la creazione delle viste.

3.5.2 Segnalazione bug e relativa risoluzione

Non di rado accadeva durante la codifica di riscontrare alcuni bug, oppure di dover integrare alcune caratteristiche al sistema non previste nella progettazione. A questo proposito, durante la fase di progettazione, il team assieme al tutor, aveva concordato una procedura per la gestione di queste situazioni tramite la piattaforma di *GitLab*. Il componente del team che riscontrava un bug o riportava la richiesta di aggiungere nuove funzionalità doveva:

- Creare un nuovo *issue* fornendo un titolo breve ma esaustivo e una descrizione completa di tutti i dettagli allegando, se necessario, schermate o porzioni di codice. Chi avrebbe letto l'*issue* avrebbe dovuto essere in grado di interpretarlo senza dover approfondire con chi l'aveva creato;
- Selezionare un assegnatario, che avrebbe avuto l'onere di risolvere il bug, con la possibilità di assegnare l'*issue* anche a se stessi;
- Etichettare l'*issue* con una o più *labels*, per identificare il tipo di problema riportato; nel caso di un problema nella codifica andava riportata la label “bug”, oppure per l'aggiunta di nuove funzionalità “enhancement”;
- Eventualmente, era possibile creare un nuovo branch nel caso di aggiunta di nuove funzionalità isolate dal branch master;

3.5.3 Stampa 3D

Per completare il prodotto ho realizzato il primo prototipo di involucro con la stampante 3D.

La procedura per la stampa 3D di un modello, come già sperimentato durante lo studio preliminare, consisteva di tre passaggi strettamente sequenziali:

- **Pre-riscaldamento e caricamento del materiale:** tramite il pannello di controllo della stampante, dovevo portare in temperatura il piano di stampa a 60 °C e l'ugello di estrusione a 200 °C. Successivamente andava caricato il filamento di materiale che avrei utilizzato per la creazione del modello: le temperature selezionate erano adatte per l'utilizzo di un materiale chiamato PLA (acido polilattico).

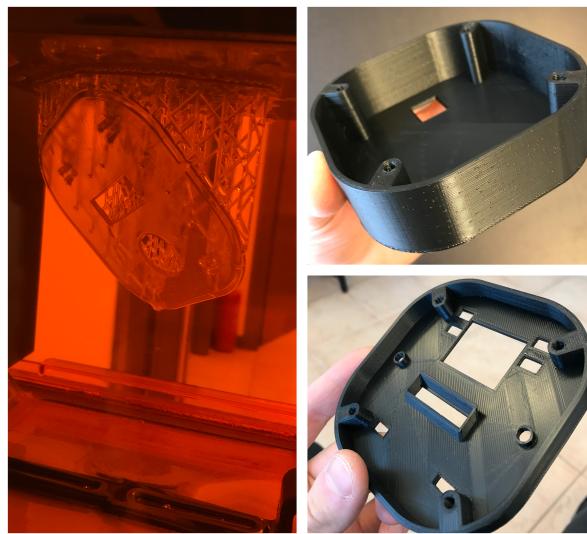


Figura 3.12: I primi prototipi di involucro stampati in 3D.

Con l'involucro a disposizione ho potuto procedere con l'assemblaggio finale di tutte le componenti elettroniche, terminando così la prima versione del prodotto. A codifica completata, ho concluso con la preparazione dei test di sistema e successivamente con la loro esecuzione.

3.5.4 Redazione dei manuali

I requisiti richiedevano anche la redazione dei manuali, necessari all'utente e al personale interno, per un corretto utilizzo del sistema:

- **Manuale d'uso utente:** questo manuale è stato scritto per guidare l'utente finale all'utilizzo sia della piattaforma online per la gestione degli utenti, che per l'utilizzo del sistema di apertura porta;
- **Manuale di installazione:** breve guida per l'installatore dove sono riportate le istruzioni per la corretta installazione delle componenti elettriche quali centralina e lettore esterno;
- **Manuale di configurazione:** questo è un manuale ad uso interno, il suo scopo è di guidare il tecnico durante l'assemblaggio delle componenti elettroniche e la configurazione software, operazioni previste prima della consegna al cliente;

Per la scrittura dei manuali ho utilizzato L^AT_EX, un linguaggio di markup utilizzato per la stesura di testi. Ho effettuato questa scelta per alcuni motivi in particolare:

- **Buona conoscenza del linguaggio:** ho spesso usato questo linguaggio per la stesura dei documenti di alcuni progetti universitari, per cui disponevo già di una buona conoscenza nel suo utilizzo;

- **Indici e riferimenti:** facilità, flessibilità ed efficienza nella creazione degli indici e dei riferimenti incrociati; occupandosi solamente del contenuto, questo tipo di strutture vengono generate automaticamente;
- **Gestione versionamento:** si integra perfettamente con software per il controllo della versione (Git, SVN, ...) permettendo di gestire facilmente le versioni del documento, come se fosse un software. La sua struttura permette inoltre a più utenti di lavorare contemporaneamente a diverse parti del documento.

3.6 Verifica e validazione

3.6.1 Test

Al fine di collaudare e validare il prodotto ho previsto diversi test di unità e di sistema per verificare la correttezza di porzioni di codice e la loro integrazione nel sistema completo. I test di unità, a differenza di quelli di sistema, sono stati eseguiti anche durante la fase di codifica, in modo da segnalare ed eliminare tempestivamente la presenza di eventuali bug.

Test del codice di Arduino

Effettuare dei test di unità direttamente sulla scheda Arduino avrebbe dato vita ad un processo ripetitivo e lento composto da:

- Modifica del codice;
- Compilazione e caricamento sulla scheda Arduino;
- Osservazione del comportamento e trarre ipotesi sul suo funzionamento, positivo o negativo che sia;
- Ripetere i passaggi.

Per questo motivo ho deciso di testare la qualità del codice scritto facendolo eseguire direttamente al computer. I test di unità che ho previsto avevano la seguente struttura:

1. Per prima cosa ho isolato la parte di codice di interesse, rendendola eseguibile come programma esterno;
2. Ho composto una previsione di output anche relazionata all'eventuale input fornito all'unità;
3. Nel caso in cui la porzione di codice effettuasse chiamate a funzioni di librerie esterne, ho creato dei mock-up che avrebbero simulato il comportamento di tali procedure, in modo da testare esclusivamente la qualità del codice scritto da me;

4. Dopo l'esecuzione del test, se l'output corrispondeva alle attese si procedeva al caricamento del test sulla scheda Arduino, altrimenti si modificava il codice in modo da correggerlo;
5. Se il comportamento del codice eseguito sulla scheda Arduino era ancora corretto, il test terminava, altrimenti si procedeva identificando e correggendo la natura del comportamento anomalo.

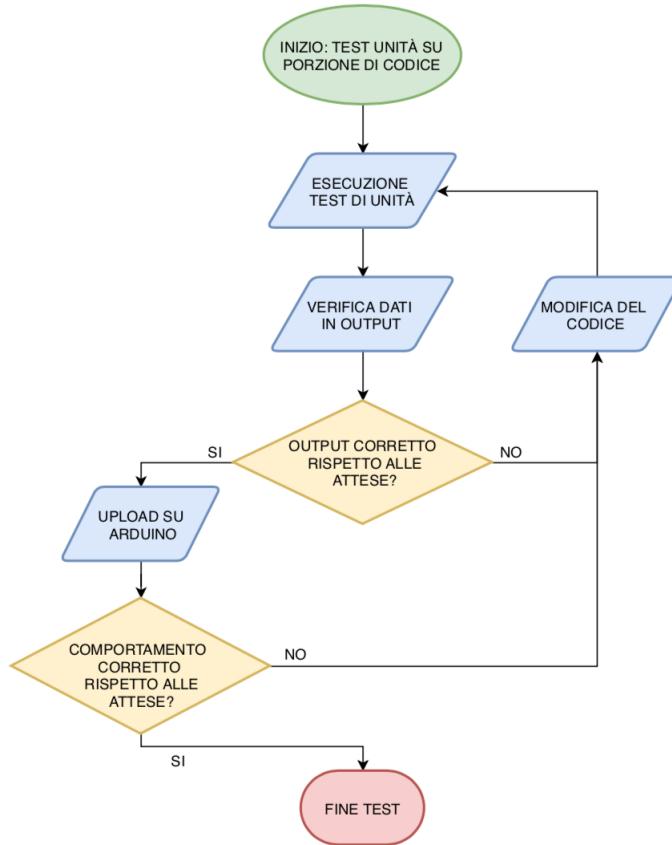


Figura 3.13: Diagramma rappresentante l'esecuzione di un test di unità.

Con questa procedura ho riscontrato, come atteso, che circa nel 90% dei casi la porzione di codice aveva lo stesso comportamento sia nell'esecuzione sul computer che sulla scheda Arduino. In questo modo ho ridotto i tempi ottimizzando il processo di test.

Test gestionale utente

Parallelamente ad Arduino, e quindi alla centralina, dovevo eseguire dei test anche sulla piattaforma online di gestione degli accessi. Anche in questo caso ho previsto alcuni test di unità sul codice PHP che gestisce la parte dinamica dell'applicazione. Per farlo ho utilizzato il framework di testing PHPUnit, incluso nel framework Laravel; ho isolato singolarmente le funzionalità implementate oggetto di test e,

per ognuna di queste, scritto degli snippet per simularne il funzionamento sia in condizioni normali che anomale.

Laravel mette inoltre a disposizione un API di semplice utilizzo per effettuare chiamate HTTP e verificare l'output di tali chiamate. Ho utilizzato queste funzionalità per testare l'applicazione in tutte le sue parti, simulando l'utilizzo da parte di un utente, ma ho anche testato la funzione che gestisce le chiamate HTTP da parte della centralina, simulando l'invio di alcune chiavi e analizzando la risposta del server.

Per l'occasione ho creato un file di configurazione dedicato alla fase di test del tipo `.env.testing`, e un database separato, già popolato con dati utili.

Di seguito uno dei test eseguiti per analizzare l'efficacia delle chiamate HTTP dalla centralina Arduino verso il server:

```

1 public function httpRequestArduino()
2 {
3     $data = [
4         'codice_chiave' => "klgfrfzteqcabzawpqrft",
5         'id_varco' => "1",
6         'Host' => "fabkey.testing.com",
7         'User-Agent' => "arduino-ethernet",
8         'X-Authorization-Token' =>
9             ↵ "2c0bc4ab56c59ce277697c53bc6e7860",
10        'X-Authorization-Time' => "12345"
11    ];
12    $response = $this->json('POST', '/api/verifica_accesso', $data);
13    $response->assertStatus(200);
14    $response->assertJson(['status' => true]);
15    $response->assertJson(['return_message' => "#1#"]);
}

```

Il codice sopra riportato simula l'invio di una chiave cifrata, corrispondente ad un utente con permesso di accesso. Se il test fallisce significa che i valori di risposta ottenuti non corrispondono alle attese; al contrario, se il test termina correttamente, l'unità di codice è stata verificata con successo.

Riepilogo finale

Al termine dell'esecuzione dei test ho potuto trarre delle conclusioni rispetto alla copertura di codice per quanto riguarda i test di unità e alla copertura dei requisiti per i test di sistema.

I test di unità mi sono tornati utili per poter dimostrare la correttezza e la completezza del codice scritto, analizzando porzioni di codice, nella maggior parte dei casi identificate da procedure.

Di seguito i risultati di tali test:

Applicazione	Test eseguiti	Test superati	Copertura
HWFabKey	18	18	69%
Gestionale accessi front-end	24	24	72%
Gestionale accessi back-end	33	33	78%

Tabella 3.3: Risultati dei test di unità.

I test di sistema, invece, mi hanno permesso di verificare la copertura dei requisiti, confrontando quelli definiti durante la fase di analisi con quelli sviluppati al termine del progetto. Di seguito i risultati:

Test eseguiti	Requisiti definiti	Requisiti soddisfatti
11	31	29

Tabella 3.4: Risultati dei test di sistema.

Collaudo finale e consegna

Al seguito di tutti i test, io e il tutor abbiamo organizzato un collaudo finale convocando il cliente che avrebbe effettuato la prima installazione del prodotto. Il collaudo finale si è svolto simulando tutte le operazioni previste e dimostrando la copertura completa di tutti i requisiti richiesti in fase di analisi. La dimostrazione si è conclusa senza imprevisti e con la soddisfazione del cliente, inoltre io, il tutor e il cliente abbiamo firmato un verbale di collaudo che constatava la fine dello sviluppo del progetto e dava la possibilità di procedere con la prima installazione.

Capitolo 4

Valutazione retrospettiva

4.1 Obiettivi raggiunti

Breve bilancio sugli obiettivi raggiunti in rapporto a quelli preventivati.

4.2 Valutazione formativa

Valutazione di quanto effettivamente appreso durante l'esperienza di stage.

4.3 Considerazioni personali

Analisi sulla distanza (o vicinanza) tra il corso di studi (e quindi le conoscenze apprese durante lo studio) e il mondo lavorativo incontrato durante lo stage, mettendo in evidenza le lacune che si sono dovute colmare per completare il tirocinio. Consigli al corso di studi sulla base dell'esperienza personale.

Appendice A

Appendice A

Citazione

Autore della citazione

Bibliografia