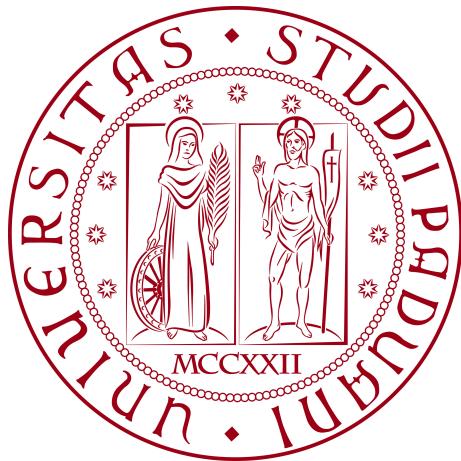


Università degli Studi di Padova

DIPARTIMENTO DI MATEMATICA “TULLIO LEVI-CIVITA”

CORSO DI LAUREA IN INFORMATICA



**Ottimizzazione della produzione in una sala
stampa 3D: programmazione a vincoli e Google
OR-Tools.**

Tesi di laurea triennale

Relatrice

Profssa. Losiouk Eleonora

Laureando

Mazzotti Matteo

Matricola 2068245

© Mazzotti Matteo, Dicembre 2025. Tesi di laurea triennale: “*Ottimizzazione della produzione in una sala stampa 3D: programmazione a vincoli e Google OR-Tools.*”, Università degli Studi di Padova, Dipartimento di Matematica “Tullio Levi-Civita”, Corso di Laurea in Informatica.

Ringraziamenti

Desidero esprimere la mia gratitudine alla professoressa Losiouk Eleonoraper l'aiuto e il sostegno che mi ha dato durante la stesura dell'elaborato.

Vorrei anche ringraziare, con affetto, i miei genitori per il loro sostegno, il grande aiuto e la loro presenza in ogni momento durante gli anni di studio.

Desidero poi ringraziare i miei amici per i bellissimi anni trascorsi insieme e le mille avventure vissute.

Padova, Dicembre 2025

Mazzotti Matteo

Sommario

L’azienda Spazio Dev dispone di una sala stampa 3D la cui pianificazione manuale della produzione richiede un notevole tempo operativo. Il presente lavoro descrive lo sviluppo di un sistema automatizzato, basato sulla programmazione a vincoli e sull’utilizzo di Google OR-Tools (CP-SAT), in grado di proporre all’operatore una coda di stampa ottimizzata secondo le buone pratiche aziendali. Il sistema, integrato con il software gestionale esistente, costituisce una soluzione concreta al problema della pianificazione manuale, contribuendo a migliorare la produttività della sala stampa e a ottimizzare l’impiego delle risorse.

Indice

Acronimi e abbreviazioni	ix
Glossario	x
1 Introduzione	1
1.1 L’azienda	1
1.1.1 Mugalab	1
1.2 Il progetto	2
1.2.1 L’idea	2
1.2.2 Motivazioni e problematiche attuali	3
1.3 Organizzazione del testo	3
2 Il contesto aziendale	5
2.1 Sviluppo software	5
2.1.1 Version Control System	5
2.1.2 Modello di lavoro agile	7
2.1.3 Issue Tracking System	8
2.1.4 Strumenti di comunicazione	9
2.2 Gestione della sala stampe	9
2.2.1 Ottimizzazione della coda di stampa	10
2.2.1.1 Orari di avvio della stampa	10
2.2.1.2 Cambi di materiale	10
2.2.1.3 Cambi ugello	11
3 Descrizione dello stage	12
3.1 Introduzione al progetto	12

INDICE

3.2 Analisi preventiva dei rischi	12
3.3 Pianificazione	13
4 Analisi dei requisiti	15
4.1 Casi d'uso	15
4.2 Tracciamento dei requisiti	16
4.3 Tabelle dei requisiti	17
5 Progettazione e codifica	19
5.1 Tecnologie e strumenti	19
5.2 Ciclo di vita del software	19
5.3 Progettazione	19
5.3.1 Namespace 1	19
5.4 Design Pattern utilizzati	20
5.5 Codifica	20
6 Verifica e validazione	21
7 Conclusioni	23
7.1 Consuntivo finale	23
7.2 Raggiungimento degli obiettivi	23
7.3 Conoscenze acquisite	23
7.4 Valutazione personale	24
7.5 Valutazione personale	24
Bibliografia	i
Sitografia	ii

Elenco delle figure

1.1	Logo dell'azienda Spazio Dev	1
1.2	Logo di Mugalab	2
2.1	Funzionamento Gitflow Workflow	6
2.2	Framework Scrum	7
2.3	Kanban board	9
4.1	Use Case 0: Scenario principale	15
6.1	Lorem	21

Elenco delle tavole

3.1	Tabella riassuntiva della pianificazione del periodo di stage.	14
4.1	Tabella del tracciamento dei requisiti funzionali.	17
4.2	Tabella del tracciamento dei requisiti qualitativi.	18
4.3	Tabella del tracciamento dei requisiti di vincolo.	18

Elenco dei codici sorgenti

5.1 Example of code	20
6.1 Fibonacci recursive	22

Acronimi e abbreviazioni

API Application Program Interface. [23](#)

SDK Software Development Kit. [23](#)

TSA Termine solo acronimo. [23](#)

UML Unified Modeling Language. [15](#)

Glossario

API In informatics, an API is a set of procedures available to programmers, typically grouped to form a toolkit for a specific task within a program. Its purpose is to provide an abstraction, usually between hardware and the programmer or between low-level and high-level software, simplifying the programming process. [3](#), [23](#)

Backlog Elenco ordinato e dinamico di elementi di lavoro (item) che rappresentano valore da realizzare. In Scrum, il Product Backlog, di responsabilità del Product Owner, raccoglie e priorizza le esigenze del prodotto; lo Sprint Backlog è il piano dei Developer per raggiungere lo Sprint Goal durante lo Sprint. [7](#)

Bugfix modifica del codice, della configurazione o dei dati volta a rimuovere un malfunzionamento (bug) e a ripristinare il comportamento atteso del sistema senza introdurre cambiamenti funzionali non richiesti. [6](#)

Bug In informatica, errore di funzionamento di un sistema o di un programma. [6](#)

Daily Scrum Evento giornaliero, time-box di 15 minuti, in cui i Developer ispezionano i progressi verso lo Sprint Goal e adattano il piano per le prossime 24 ore; non è una riunione di status per gli stakeholder. [7](#)

Feature Unità coerente di comportamento di un sistema che produce un beneficio osservabile per l'utente. [5](#)

Git branching model Un modello di branching Git è una strategia o un insieme di regole che definisce come i team devono creare, gestire e unire i branch in un repository Git, al fine di organizzare il flusso di sviluppo. [5](#)

Issue Unità di lavoro o ticket che rappresenta una richiesta, un bug, un miglioramento o un'attività da tracciare e gestire in un sistema di gestione del lavoro come Jira o GitHub Issues. Ogni issue dovrebbe includere contesto, criteri di accettazione e stato per facilitare la collaborazione. [8](#)

POC Il Proof Of Concept (POC) è l'allestimento di una demo prototipale del sistema o applicazione in sviluppo o in corso di valutazione. [3](#)

Product Owner Figura chiave di Scrum responsabile di massimizzare il valore del prodotto e del lavoro del team. Definisce e mantiene il Product Backlog, ne ordina gli elementi in base al valore e agli obiettivi, chiarisce i requisiti e accetta l'incremento completato. [7](#)

Scrum Master Servant leader del team Scrum. Promuove e supporta Scrum come definito nella Scrum Guide, facilita gli eventi Scrum, rimuove impegnamenti, tutela il team e aiuta l'organizzazione ad adottare pratiche agili. [7](#)

SDK A Software Development Kit (SDK) is a collection of development tools in one installable package, facilitating application creation by providing a compiler, debugger, and sometimes a software framework. SDKs are typically specific to a hardware platform and operating system combination. Many application developers use specific SDKs to enable advanced functionalities such as advertisements, push notifications, etc. [23](#)

Server di deploy Computer dedicato all'esecuzione in produzione di un'applicazione software. Ospita il codice distribuito, espone le relative API e garantisce risorse hardware e configurazioni adeguate per gestire il carico operativo previsto.. [13](#)

Sprint Time-box tipico del framework Scrum, della durata compresa fra uno e quattro settimane, durante il quale il team realizza un incremento potenzialmente rilasciabile seguendo uno Sprint Goal condiviso. Lo Sprint include pianificazione, sviluppo, revisione e retrospettiva. [8](#)

Sprint Planning Evento che apre lo Sprint in cui lo Scrum Team definisce lo Sprint Goal, seleziona gli elementi del Product Backlog da includere nello Sprint e pianifica il lavoro necessario nel Sprint Backlog. [7](#)

Sprint Retrospective Ultimo evento dello Sprint dedicato all'ispezione delle modalità di lavoro del team e all'individuazione di miglioramenti pratici da implementare nel prossimo Sprint. [7](#)

Sprint Review Evento alla fine dello Sprint per ispezionare l'incremento e adattare il Product Backlog. Coinvolge stakeholder e team per raccogliere feedback, rivedere i risultati e allineare i prossimi passi. [7](#)

Nome del termine Descrizione. [15](#), [23](#)

Version Control System Un sistema di versionamento (Version Control System o VCS) è uno strumento software che traccia e gestisce le modifiche apportate a un file o a un insieme di file nel tempo, permettendo di recuperare versioni precedenti e di collaborare con altri utenti. [5](#)

Capitolo 1

Introduzione

1.1 L’azienda

Spazio Dev S.r.l.¹ è una software house situata a Tombolo (PD) fondata da due soci, ad oggi l’azienda conta circa 17 dipendenti e si occupa di offrire servizi legati al mondo dello sviluppo web e dell’ottimizzazione dei processi industriali. L’azienda si occupa, nello specifico, di sviluppare siti web, e-commerce, software su misura e di integrare algoritmi che sfruttano l’intelligenza artificiale per ottimizzare la produzione o monitorare lo stato dell’azienda in tempo reale.



Figura 1.1: Logo dell’azienda Spazio Dev

1.1.1 Mugalab

Mugalab² è la divisione di Spazio Dev S.r.l. dedicata alla gestione della sala stampa 3D presente nella sede aziendale. Attraverso la tecnica della stampa additiva, Mugalab si occupa di:

- progettare e prototipare componenti per il settore industriale;

¹ Sito Spazio Dev. URL: <https://spaziodev.eu/>.

² Sito Mugalab. URL: <https://mugalab.com/>.

- progettare e prototipare componenti per dispositivi elettronici;
- disegnare e produrre oggetti ornamentali.



Figura 1.2: Logo di Mugalab

1.2 Il progetto

1.2.1 L'idea

L'idea di questo percorso di stage curriculare nasce dalla necessità dell'azienda di gestire in maniera efficiente la produzione delle componenti stampate in 3D. Attualmente lo stabilimento dispone di 12 stampanti e la pianificazione della stampa degli ordini di produzione viene effettuata manualmente da un singolo operatore, il quale si accerta di ottimizzare la produzione seguendo alcune buone pratiche che permettono di risparmiare tempo e aumentare l'efficienza della produzione. L'obiettivo principale dell'azienda è quindi quello di ideare un sistema automatizzato, il cui scopo è creare una coda di stampa ottimizzata a partire dagli ordini di produzione non ancora evasi completamente. Tale sistema dovrà integrarsi completamente con il sistema gestionale esistente, il quale viene utilizzato quotidianamente dal personale dell'azienda per gestire la coda di stampa e registrare gli ordini ricevuti dalle piattaforme utilizzate per la vendita delle componenti e degli oggetti ornamentali (Amazon, Shopify e vendita B2B). Questo progetto è destinato esclusivamente ad uso interno dell'azienda,

assumendo quindi il ruolo di *Proof Of Concept_G* della fattibilità di un sistema di schedulazione automatizzato e vincolato. Il codice potrà quindi essere ulteriormente ottimizzato e ampliato in modo tale da migliorarne l'efficienza e aumentare la qualità del risultato prodotto, oltre ad essere adattato a scenari d'utilizzo differenti da quello proposto.

1.2.2 Motivazioni e problematiche attuali

La pianificazione manuale della coda di stampa genera ritardi e inefficienze operative, con ricadute dirette sui tempi di consegna. Una schedulazione basata su un algoritmo di ottimizzazione consente di generare piani fattibili in modo più rapido e meno incline a errori, migliorando l'utilizzo delle stampanti e riducendo i tempi morti.

1.3 Organizzazione del testo

Il secondo capitolo descrive ...

Il terzo capitolo approfondisce ...

Il quarto capitolo approfondisce ...

Il quinto capitolo approfondisce ...

Il sesto capitolo approfondisce ...

Nel settimo capitolo descrive ...

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: *Application Program Interface_G*;

CAPITOLO 1. INTRODUZIONE

- i termini in lingua straniera o facenti parti del gergo tecnico sono evidenziati con il carattere *corsivo*.

Capitolo 2

Il contesto aziendale

2.1 Sviluppo software

Spazio Dev è un’azienda di piccole dimensioni, di recente fondazione e ancora in fase di crescita, di conseguenza il progetto è stato sviluppato con budget limitato e da una singola persona. I processi di sviluppo aziendali, pertanto, sono ancora in rapida evoluzione.

2.1.1 Version Control System

L’azienda utilizza Git¹ come *Version Control System*_G e Gitea² come piattaforma di hosting per i repository. Per quanto riguarda il *Git branching model*_G viene adottato il Gitflow workflow³, adattato nel seguente modo:

- il *branch* dedicato alle versioni stabili è denominato *main*;
- il *branch* dedicato alle versioni di sviluppo è denominato *develop*;
- i *branch* dedicati alle implementazioni di *feature*_G vengono denominati con lo schema *feature/*<nome_feature> dove il parametro <nome_feature> è una breve descrizione della *feature* che si sta codificando;

¹Sito ufficiale Git. URL: <https://git-scm.com/>.

²Sito ufficiale Gitea. URL: <https://about.gitea.com/>.

³Spiegazione completa Gitflow workflow. URL: <https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow>.

- i *branch* dedicati ai *bugfix*_G vengono denominati con lo schema *fix/<nome_bug>* dove il parametro *<nome_bug>* è una breve descrizione del *bug*_G che si sta risolvendo.
- i *branch* dedicati alla documentazione vengono denominati con lo schema *docs/<nome_documentazione>* dove il parametro *<nome_documentazione>* è una breve descrizione della modifica alla documentazione.

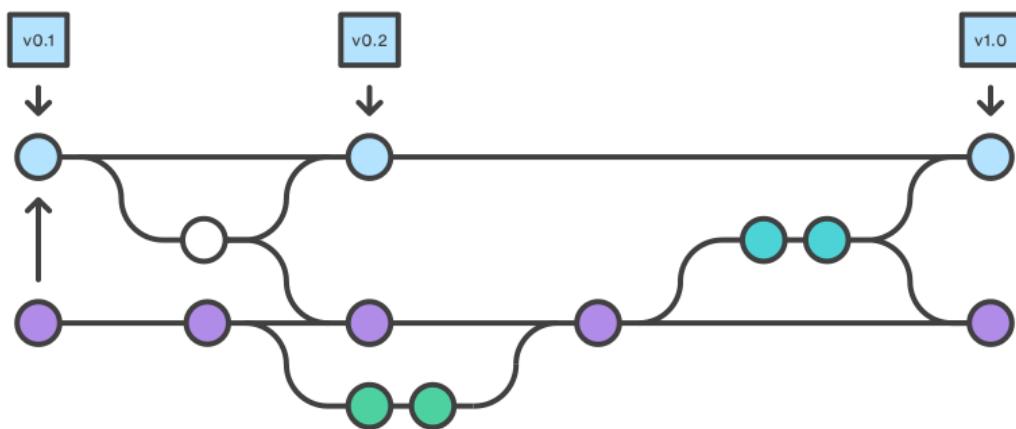


Figura 2.1: Funzionamento Gitflow Workflow

Viene inoltre adottato lo standard Conventional Commits 1.0.0⁴ per la scrittura dei messaggi di commit, in particolare ogni commit possiede un messaggio così formato: *<tipo>: <descrizione>*, dove il parametro *<tipo>* è:

- *fix* in caso di commit contenente *bugfix*;
- *docs* in caso di commit di aggiornamento della documentazione;
- *feat* in caso di commit contenente *feature*.

Non sono state definite policy di protezione dei rami, requisiti di revisione o regole di merge, tali aspetti potranno tuttavia essere definiti in futuro dall'azienda.

⁴Documentazione ufficiale Conventional Commits 1.0.0. URL: <https://www.conventionalcommits.org/en/v1.0.0/>.

2.1.2 Modello di lavoro agile

L'azienda adotta il framework Scrum⁵ come modello di lavoro agile⁶, in particolare:

- Il ruolo di *Product Owner*_G viene ricoperto dai titolari dell'azienda i quali si occupano di creare e gestire il *backlog*_G del prodotto, fornire indicazioni al team su quali feature implementare e decidono le scadenze dei rilasci del prodotto;
- Il ruolo di *Scrum Master*_G viene ricoperto dai titolari dell'azienda, i quali si occupano di pianificare *Sprint Planning*_G, *Daily Scrum*_G, *Sprint Review*_G e *Sprint Retrospective*_G;
- Il ruolo del team di sviluppo viene ricoperto dagli sviluppatori, i quali vengono divisi in gruppi e assegnati ai progetti in fase di sviluppo.

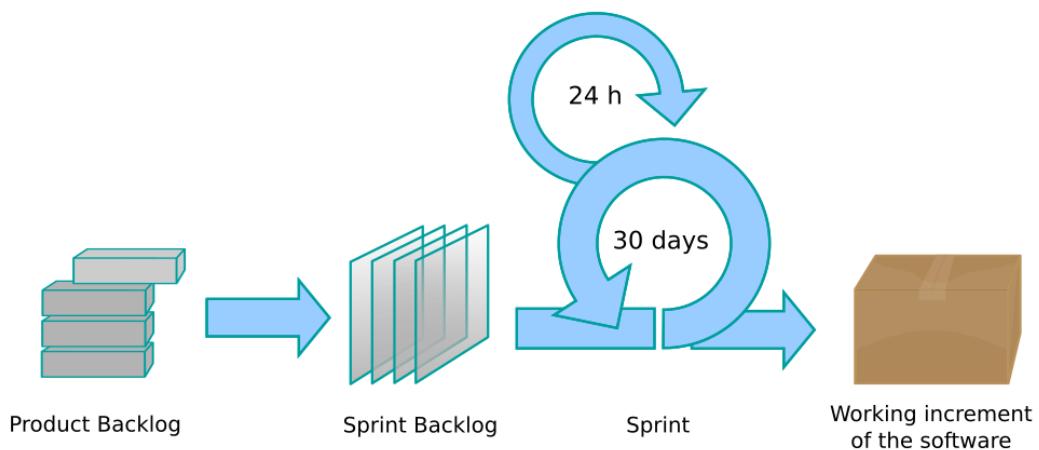


Figura 2.2: Framework Scrum

⁵Framework Scrum. URL: [https://it.wikipedia.org/wiki/Scrum_\(informatica\)](https://it.wikipedia.org/wiki/Scrum_(informatica)).

⁶Manifesto Agile. URL: <http://agilemanifesto.org/iso/it/>.

2.1.3 Issue Tracking System

L’azienda utilizza Plane⁷ come sistema di tracciamento delle attività. Le *issue* sono organizzate su una board in stile Kanban composta da cinque colonne:

- *Backlog*: raccolta iniziale delle attività proposte, non ancora pianificate;
- *Todo*: attività prioritarie selezionate per l’esecuzione;
- *In Progress*: attività in lavorazione;
- *In Review*: attività concluse e in fase di verifica;
- *Done*: attività completate.

Ogni *issue* è inoltre associata a un livello di priorità tra i seguenti:

- *Critical*: attività richiedenti intervento immediato;
- *High*: richieste ad alto impatto o con scadenza molto ravvicinata;
- *Medium*: *bugfix* non critici che vengono inseriti nella pianificazione ordinaria degli *Sprint*;
- *Low*: miglioramenti minori non prioritari.

La combinazione di colonna Kanban e priorità rende evidente sia lo stato di avanzamento sia l’urgenza.

⁷Sito ufficiale Plane. URL: <https://plane.so/>.



Figura 2.3: Kanban board

2.1.4 Strumenti di comunicazione

Le comunicazioni interne avvengono interamente tramite Telegram⁸: ogni progetto dispone di un gruppo dedicato in cui titolari e sviluppatori condividono aggiornamenti, documentazione e decisioni operative in tempo reale. Questo canale unico permette di evitare dispersioni e di mantenere traccia delle richieste provenienti dai clienti.

Le riunioni operative vengono organizzate con cadenza variabile, in funzione delle necessità del progetto o delle scadenze concordate con i clienti. Quando emergono nuove priorità o si avvicina un rilascio, i titolari convocano incontri mirati per allineare il team sugli obiettivi, discutere eventuali impedimenti e definire le attività da pianificare nel successivo *sprint*.

2.2 Gestione della sala stampe

Spazio Dev possiede attualmente molteplici punti di ingresso degli ordini relativi alla stampa 3D dei componenti, in particolare:

⁸ *Telegram*. Sito ufficiale Telegram. URL: <https://telegram.org> (visitato il giorno 23/10/2025).

- La vendita di vasi portafiori e complementi d'arredo viene effettuata online e sulla piattaforma e-commerce Amazon;
- Le produzioni personalizzate e B2B vengono gestite manualmente.

Per la gestione di tali ordini e della coda di stampa viene utilizzato un gestionale sviluppato dall'azienda stessa, denominato "Idrotech Manager".

2.2.1 Ottimizzazione della coda di stampa

Per gestire in maniera efficiente l'evasione degli ordini l'azienda applica diverse ottimizzazioni alla coda di stampa, in modo tale da sfruttare nel miglior modo possibile le capacità di produzione. Tali procedure riguardano in particolare gli orari di avvio della stampa, la gestione dei cambi di materiale e dei cambi di ugello.

2.2.1.1 Orari di avvio della stampa

La gestione degli orari di avvio della stampa risulta fondamentale per ottimizzare al meglio la produzione. Ciò è dovuto al fatto che ogni stampante, quando termina la produzione di un piatto, necessita di un intervento manuale da parte di un operatore per la rimozione del prodotto finito e l'avvio della stampa successiva. L'idea alla base è quindi quella di pianificare tutte le stampe brevi durante l'orario lavorativo (in modo che sia sempre possibile avviare una nuova produzione), le stampe più lunghe vengono invece avviate poco prima della fine della giornata per sfruttare al meglio i periodi di tempo in cui non c'è personale presente all'interno dello stabilimento.

2.2.1.2 Cambi di materiale

La coda di stampa può essere ottimizzata riducendo al minimo i cambi di materiale (o colore) tra una stampa e quella successiva. Questo perchè ogni volta viene richiesto di cambiare il colore del filamento o il materiale utilizzato la stampa viene interrotta e un operatore deve intervenire manualmente per la sostituzione del filamento. Per semplicità, l'azienda assume che il tempo necessario per cambiare un filamento sia di 5 minuti.

2.2.1.3 Cambi ugello

Ogni articolo possiede uno specifico ugello che viene utilizzato per cambiare lo spessore della stampa. Come per i cambi di materiale anche il cambio ugello richiede un intervento manuale da parte di un operatore. Per semplicità, l'azienda assume che il tempo necessario per cambiare un filamento sia di 15 minuti.

Capitolo 3

Descrizione dello stage

3.1 Introduzione al progetto

Come descritto nella introduzione (1) il progetto codificato durante il periodo di stage curricolare è uno schedulatore automatico che ottimizza e riordina la coda di stampa a partire dagli ordini di produzione presenti all'interno del software gestionale Idrotech Manager. Tale schedulatore deve tenere conto delle ottimizzazioni descritte nella sezione dedicata (2.2), oltre a doversi integrare con il gestionale in utilizzo dal personale.

3.2 Analisi preventiva dei rischi

La prima fase dello stage curricolare è stata dedicata all'analisi preventiva dei rischi. Tale procedura ha lo scopo di delineare i possibili rischi a cui si può andare incontro durante lo sviluppo del progetto, oltre a trovare delle possibili soluzioni per mitigare tali problematiche.

1. Errata scelta delle tecnologie

Descrizione: una scelta errata delle tecnologie da utilizzare per la codifica del progetto potrebbe portare ad un risultato inutilizzabile o non abbastanza performante.

Soluzione: coinvolgimento del responsabile nella scelta e occupare il primo periodo per validare le idee proposte.

2. Sicurezza nella integrazione del modulo

Descrizione: la comunicazione tra il modulo da sviluppare e il gestionale esistente deve rispettare degli standard minimi di sicurezza, in modo che non tutti possano accedere alle *API* esposte.

Soluzione: dedicare parte del tempo allo studio su come rendere più sicuro l'accesso alle rotte *API*.

3. Risultati di ordinamento insoddisfacenti

Descrizione: il modulo potrebbe restituire risultati non soddisfacenti, con conseguenti inefficienze nella produzione.

Soluzione: validare assieme al tutor interno le logiche implementate in maniera periodica, testare il modulo con dataset realistici.

4. Prestazioni dello schedulatore

Descrizione: il modulo potrebbe impiegare troppo tempo per ottimizzare la coda di stampa.

Soluzione: utilizzare un *server di deploy* con buone prestazioni, velocizzare l'esecuzione diminuendo, ad esempio, il tempo disponibile per la ricerca di una soluzione ottima.

5. Rispetto delle scadenze

Descrizione: le tempistiche dello stage curricolare potrebbero non essere sufficienti per avere un risultato concreto e utilizzabile.

Soluzione: svolgere una pianificazione (3.3) accurata del periodo di tempo a disposizione e convalidare il raggiungimento degli obiettivi.

3.3 Pianificazione

Oltre all'analisi preventiva dei rischi (3.2) è stata svolta una pianificazione accurata di tutte le fasi del periodo di stage. Questo, oltre a mitigare il rischio "Rispetto delle scadenze" (5) serve a determinare i contenuti da revisionare al termine di ogni *sprint*.

CAPITOLO 3. DESCRIZIONE DELLO STAGE

Settimana	Obiettivi	Ore
1	Analisi dei requisiti del progetto, scelta e inizio studio delle tecnologie.	40
2	Lettura della documentazione, studio del gestionale e progettazione delle integrazioni.	40
3	Inizio della codifica del progetto.	40
4	Codifica del progetto.	40
5	Codifica del progetto.	40
6	Codifica del progetto.	40
7	Codifica del progetto, validazione logiche implementate.	40
8	Test dell'algoritmo con dataset realistici.	20

Tabella 3.1: Tabella riassuntiva della pianificazione del periodo di stage.

Capitolo 4

Analisi dei requisiti

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)G* 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 *Nome del termineG* 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.

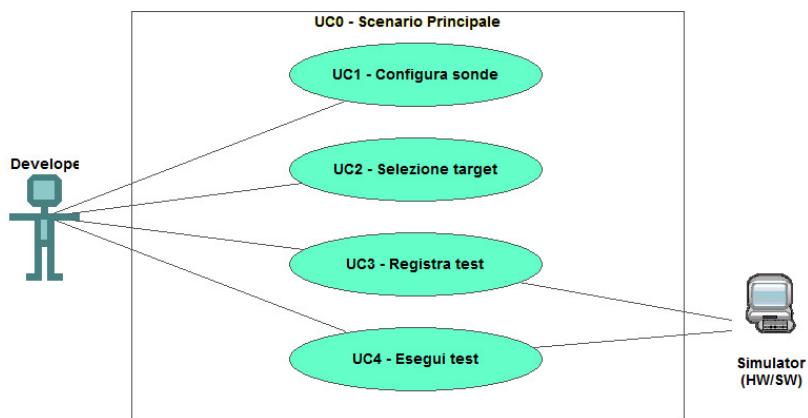


Figura 4.1: Use Case 0: Scenario principale

UC0: Scenario principale

Attori Principali: Sviluppatore applicativi.

Precondizioni: Lo sviluppatore è entrato nel plugin 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.

UC1: Gestione Utente

Attori Principali: Amministratore, Utente Registrato.

Precondizioni: L'utente deve essere autenticato nel sistema.

Descrizione: L'utente può gestire le informazioni del proprio profilo.

Postcondizioni: Le modifiche vengono salvate nel sistema.

Scenario Alternativo: Se l'utente non è autenticato, visualizza un messaggio di errore.

UC2: Creazione Prodotto

Attori Principali: Amministratore.

Precondizioni: L'amministratore ha effettuato l'accesso al sistema.

Descrizione: L'amministratore può aggiungere un nuovo prodotto al catalogo.

Postcondizioni: Il nuovo prodotto viene aggiunto con successo.

Scenario Alternativo: Se i campi obbligatori non sono compilati, visualizza un messaggio di errore.

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, dove ogni requisito è identificato con il carattere **R**, è così strutturato:

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.

4.3 Tabelle dei requisiti

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

Tabella 4.1: Tabella del tracciamento dei requisiti funzionali.

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

Tabella 4.2: Tabella del tracciamento dei requisiti qualitativi.

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

Tabella 4.3: Tabella del tracciamento dei requisiti di vincolo.

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

5.3.1 Namespace 1

Descrizione namespace 1.

5.4 Design Pattern utilizzati

5.5 Codifica

Blocco di codice in C

```
#include <stdio.h>
int main() {
    print("Hello, world!");
    return 0;
}
```

Codice 5.1: Example of code

Capitolo 6

Verifica e validazione



Figura 6.1: Lorem

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna,

vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

 Lorem ipsum:

```
def recur_fibo(n):
    if n <= 1:
        return n
    else:
        return(recur_fibo(n-1) + recur_fibo(n-2))

nterms = 10
if nterms <= 0:
    print("Please enter a positive integer")
else:
    print("Fibonacci sequence:")
    for i in range(nterms):
        print(recur_fibo(i))
```

Codice 6.1: Fibonacci recursive

 Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Capitolo 7

Conclusioni

7.1 Consuntivo finale

Esempio di aggiunta di un termine con glossario e acronimo:

 Lorem *Software Development Kit (SDK)_G* ipsum dolor.

 Nel successivo utilizzo, apparirà solo l'acronimo:

 Lorem *SDK_G*.

 Nel caso si voglia invece mettere solo il termine per esteso, si può usare:

 Lorem *Software Development Kit_G*.

7.2 Raggiungimento degli obiettivi

Esempio di termine con solo acronimo

 Lorem *Termine solo acronimo (TSA)_G*, ipsum dolor sit amet

 termine costruito senza acronimo: Lorem *Nome del termine_G*, ipsum dolor
 sit amet

7.3 Conoscenze acquisite

 Lorem Ipsum dolor Lorem *Application Program Interface (API)_G*

 Lorem Ipsum dolor Lorem *Application Program Interface_G*

 Si può consultare il file *glossary_acronyms.tex* per alcuni esempi.

7.4 Valutazione personale

7.5 Valutazione personale

Bibliografia

Testi

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

Articoli

Einstein, Albert, Boris Podolsky e Nathan Rosen. «Can Quantum-Mechanical Description of Physical Reality be Considered Complete?» In: *Physical Review* 47.10 (1935), pp. 777–780. DOI: [10.1103/PhysRev.47.777](https://doi.org/10.1103/PhysRev.47.777).

Sitografia

Documentazione ufficiale Conventional Commits 1.0.0. URL: <https://www.conventionalcommits.org/en/v1.0.0/> (cit. a p. 6).

Framework Scrum. URL: [https://it.wikipedia.org/wiki/Scrum_\(informatica\)](https://it.wikipedia.org/wiki/Scrum_(informatica)) (cit. a p. 7).

Framework Scrum. Licenza: CC BY-SA 4.0; l'immagine SVG originale è stata convertita in PNG per l'inclusione nel documento. URL: https://commons.wikimedia.org/wiki/File:Scrum_process.svg (visitato il giorno 23/10/2025).

Manifesto Agile. URL: <http://agilemanifesto.org/iso/it/> (cit. a p. 7).

Sito Mugalab. URL: <https://mugalab.com/> (cit. a p. 1).

Sito Spazio Dev. URL: <https://spaziodev.eu/> (cit. a p. 1).

Sito ufficiale Git. URL: <https://git-scm.com/> (cit. a p. 5).

Sito ufficiale Gitea. URL: <https://about.gitea.com/> (cit. a p. 5).

Sito ufficiale Plane. URL: <https://plane.so/> (cit. a p. 8).

Spiegazione completa Gitflow workflow. URL: <https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow> (cit. a p. 5).

Telegram. Sito ufficiale Telegram. URL: <https://telegram.org> (visitato il giorno 23/10/2025) (cit. a p. 9).