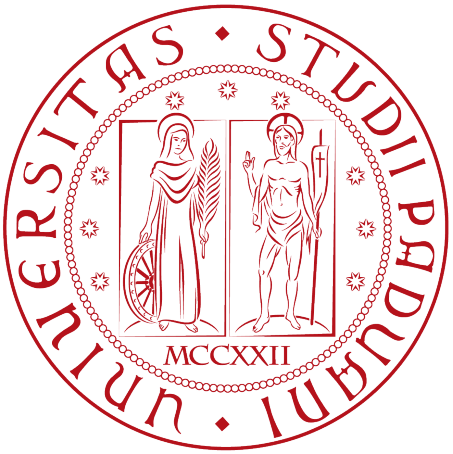
Università degli Studi di Padova

Dipartimento di Matematica "Tullio Levi-Civita"

Corso di Laurea in Informatica



Sviluppo di algoritmi euristici di ottimizzazione e applicazione alla pianificazione della produzione

Tesi di laurea triennale

Relatore

Prof. Luigi De Giovanni

Laureando

Riccardo Basso

Anno Accademico 2018-2019

Sommario

Il presente documento descrive il lavoro svolto durante il periodo di stage, della durata di circa trecentoventi ore, dal laureando Basso Riccardo presso l'azienda Ergon Informatica. Gli obbiettivi da raggiungere erano molteplici e sono descritti nei seguenti capitoli.
Il primo capitolo tratta dell'azienda e del suo rapporto con l'università.
Il secondo capitolo fornisce un'idea generale del progetto di stage, accompagnata dall'analisi dei rischi.
Il terzo capitolo racchiude il tracciamento dei requisiti e descrive le varie tecnologie utilizzate durante il progetto.
Il quarto capitolo parla in dettaglio del progetto di stage e di come è stato svolto. Racchiude una panoramica iniziale dell'architettura per poi soffermarsi in dettaglio sull'analisi del problema stesso e della sua soluzione. Si conclude con i risultati dei test effettuati.
Il quinto e ultimo capitolo racchiude le varie conclusioni raggiunte al termine dello stage.

Ringraziamenti

Innanzitutto, vorrei esprimere la mia gratitudine al Prof. Luigi De Giovanni, relatore della mia tesi, per l'aiuto e il sostegno fornitomi durante lo svolgimento del progetto e nella stesura dell'elaborato.

Vorrei ringraziare tutto il personale di Ergon Informatica con il quale ho avuto il piacere di lavorare durante questo periodo formativo, senza di loro non sarebbe stato possibile portare a termine questo percorso.

Ringrazio la mia famiglia, Irene e tutti coloro che mi hanno aiutato in questi anni a portare a termine questa avventura

Un ultimo ringraziamento speciale va a mio nonno Settimo, il quale più di tutti mi ha spronato ad iniziare questo percorso di studi ma che purtroppo non è più qui per godersi la fine.

Padova, Dicembre 2019 Riccardo Basso

Indice

Elenco delle figure

Elenco delle tabelle

Capitolo 1

Introduzione

1.1 L'azienda

Ergon Informatica viene fondata nel 1988 come società di Ingegneria Informatica per applicazioni gestionali dedicate.

La società, che all'inizio conta solo alcuni dipendenti, si sviluppa in maniera costante negli anni e oggi può vantare una posizione di tutto rispetto tra le aziende dello stesso settore.

I clienti iniziali hanno giocato un ruolo fondamentale nello sviluppo del software prodotto da Ergon; essi infatti appartenevano per la maggior parte all'ambito alimentare e l'esperienza di queste prime installazioni ha permesso di acquisire delle competenze interne altamente specializzate in questo settore.

1.1.1 Organizzazione aziendale

Attualmente fanno parte della stessa gestione tre società:

- Ergon Informatica S.r.l. che si occupa dello sviluppo software, in particolare del gestionale Ergdis;
- Ergon S.r.l. che si occupa dei servizi tecnologici;
- Ergon Servizi S.r.l. che si occupa dei servizi amministrativi, logistici e di marketing delle sopracitate.

1.1.2 Servizi aziendali

- **Help Desk**: Il servizio di assistenza è fornito tramite un'apertura di chiamata che può essere effettuata via web o telefonando presso la sede. Un servizio di monitoring interno permette di individuare gli eventuali ritardi nelle risposte e negli interventi risolutivi. Il sistema di qualità (ISO 9001), che Ergon ha adottato, individua un tempo massimo di risoluzione del problema. Il cliente, collegandosi all'area a lui riservata, ha la possibilità di monitorare tutte le sue chiamate e verificare, per quelle ancora aperte, il tempo previsto di chiusura;

- **Servizio di Monitoring**: Il servizio prevede l'installazione per ogni server di un certo numero di agenti. Un operatore di Ergon, da remoto, attraverso un programma di controllo, rileva i vari tipi di "alert" e, in accordo con il cliente, può intervenire se è necessario eliminare il problema. Alcuni esempi di alert possono essere: la memoria quasi esaurita, i salvataggi non effettuati;

- **Vendita e installazione Hardware**: Il servizio prevede la vendita, installazione e configurazione di qualsiasi tipo di apparato informatico direttamente presso la sede del cliente;

- **Radio Frequenza**: Il servizio prevede il supporto all'utilizzo di apparati radio per la comunicazione;

- **Sicurezza**: Il servizio prevede la vendita, installazione e aggiornamento costante dei servizi di sicurezza, affiliandosi a dei vendor internazionali;

- **Virtualizzazione**: Il servizio prevede la virtualizzazione dei server in modo da ottimizzare la performance dell'infrastruttura attraverso la creazione di server virtuali che sostituiscono quelli fisici. Ergon Informatica è specializzata nell'installazione del software di VMWARE e nel corso degli anni ha virtualizzato l'infrastruttura di quasi tutti i suoi clienti. Per il backup dei dati, in ambiente virtualizzato, utilizziamo il software di VEEAM;

- **Networking**: Il servizio prevede la progettazione, installazione e mantenimento di reti informatiche delle aziende;

- **Servizi Cloud**: Il servizio SLA (Service Level Agreement),servizio cloud offerto da Ergon Informatica viene erogato al cliente tramite un collegamento ai server che risiedono in un datacenter con elevati livelli di sicurezza ,può essere acquistato pagando un canone mensile per posto di lavoro; il cliente può quindi valutare in modo semplice i suoi costi e, grazie alla scalabilità del sistema, adattare in qualsiasi momento la struttura alle sue esigenze.

1.1.3 Ergdis

Ergdis è il sistema ERP progettato e sviluppato da Ergon Informatica S.r.l. L'insieme dei moduli proposti copre ogni aspetto della gestione aziendale:

- **Amministrazione e finanza**: È il prodotto che semplifica l'amministrazione aziendale, gestendo la contabilità e gli obblighi fiscali dell'azienda, offrendo analisi dettagliate sulla condizione creditizia e debitoria e fornendo un quadro in tempo reale della situazione contabile aziendale;
- **Controllo di Gestione**: È l'area applicativa del software Ergdis che supporta tutto il management nelle decisioni da prendere, promuovendo un'efficace supervisione della gestione interna dell'azienda. Permette un'analisi completa dei flussi economici che interessano la realtà aziendale, organizza i budget e i consuntivi, consente l'analisi dei costi di prodotto e rende funzionale l'organizzazione interna dell'impresa;
- **Area Acquisti**: È il prodotto che permette di sopperire ai bisogni di approvvigionamento della merce e dei servizi, attraverso un'efficace gestione degli ordini fornitori. Consente di ottimizzare l'intero processo relativo all'acquisto grazie ad un'amministrazione capillare delle disponibilità dei prodotti e dei documenti ad esso collegati. Infine garantisce il controllo dell'intero ciclo passivo: dall'ordine al fornitore fino alla ricezione della fattura;
- **Radio Frequenza**: Il servizio prevede il supporto all'utilizzo di apparati radio per la comunicazione;
- **Logistica**:È il prodotto che coordina l'insieme delle attività organizzative, gestionali e strategiche che regolano le movimentazioni di magazzino, consentendo la rintracciabilità della merce, monitorando le scorte dei prodotti, gestendo i documenti di carico e scarico. Grazie a questa suite di Ergdis l'utente potrà ottimizzare i propri costi e amministrare in maniera efficiente le operazioni che interessano l'entrata, l'uscita e la giacenza degli articoli;
- **Vendite**: È l'applicativo del software Ergdis realizzato per gestire l'area commerciale dell'azienda: dall'offerta iniziale fino alla fatturazione del prodotto, dagli ordini cliente alle campagne acquisti. Si compone di moduli software personalizzabili e modellabili secondo le esigenze, che semplificano i processi del business e migliorano l'utilizzo delle risorse aziendali. Grazie alla suite di programmi del Ciclo attivo sarà possibile velocizzare il passaggio dei dati e rendere automatici molti procedimenti prima manuali;
- **Produzione**: Questa suite di programmi offre un solido supporto nella pianificazione e nel controllo della produzione, ideale per chi necessita di analizzare con flessibilità le informazioni aziendali. Il prodotto consente infatti di gestire l'intero ciclo produttivo: dalla fase di programmazione, agli avanzamenti di produzione; dal versamento del prodotto finito al calcolo dei costi;
- **Web**: La suite di programmi consente di organizzare i propri negozi virtuali dando visibilità ai prodotti e gestendo le informazioni a questi legate, per un completo controllo delle esigenze aziendali;
- **Business Intelligence**: È il prodotto che consente di gestire graficamente le informazioni aziendali, effettuando analisi su tutti i dati gestionali, visualizzandoli

su griglie, dashboard e tabelle Pivot, in modo da renderli immediati e facilmente fruibili;

- **Qualità**: È il software che permette di garantire la qualità dell'azienda, attraverso un efficace sistema di controllo della soddisfazione dei clienti e della conformità dei prodotti venduti. Consente di realizzare le schede tecniche degli articoli, di creare dei questionari e di redigere i verbali delle riunioni, quest'ultime attività necessarie per chi possiede una certificazione di qualità. Garantisce infine l'assistenza ai clienti, grazie ad un software che guida l'utente nella risposta delle richieste e nella tempestiva risoluzione dei problemi;
- **Archiviazione Documentale**: L'utente può organizzare al meglio i propri documenti, rintracciandoli con facilità e snellendo il processo di ricerca informazioni. I moduli di quest'area permettono inoltre una migliore archiviazione del materiale in linea con le norme civili vigenti;
- **Gestione Archivi**:È il prodotto che favorisce un'ottima gestione dell'azienda attraverso la creazione di file e schede tecniche che codificano i prodotti, i clienti, le consegne e li organizzano secondo una struttura logica;
- **Pianificazione Consegne**: Permette di programmare e gestire le consegne, assegnandole ad un gruppo di mezzi, ottimizzando i percorsi compiuti dagli stessi nel rispetto dei vincoli definiti dal cliente.;
- **Area Mobile**: Proiettata anche verso il mondo mobile, Ergon ha ideato alcune app che lavorano in ambiente Android e permettono di gestire numerose attività dell'area commerciale dell'azienda;

1.1.4 Azienda e stage

Ergon Informatica propone già da diversi anni nuovi progetti di stage per gli studenti dell'Università degli Studi di Padova nella giornata di StageIT. Lo scopo di questa collaborazione è quello di creare un punto di incontro professionale con lo studente che si approccia al mondo del lavoro. Entrambe le parti ne traggono beneficio: lo studente si cimenta in progetti aziendali che arricchiscono le sue conoscenze e ampliano le competenze acquisite in ambito scolastico, l'azienda invece valuta il possibile inserimento nel team di sviluppo dello studente coinvolto.

Progetti StageIT

- **App Entrata Merce terminali wifi**: Il progetto si propone di effettuare un'analisi approfondita per lo sviluppo di un'applicazione per l'entrata merce in magazzino. Il candidato dovrà valutare se sia più opportuno riscrivere l'applicazione su WEB o client/server in VB .Net, effettuando un'analisi approfondita dei vantaggi e degli svantaggi delle due scelte. Nel caso di applicazione Web, essa dovrà essere creata con appoggio su server WEB come fosse un sito, nel caso di applicazione VB .Net dovrà essere usata tramite l'appoggio di un server Windows terminal server. Dopo aver scelto la strada più opportuna, il candidato procederà con lo sviluppo dell'applicazione. Questa dovrà interfacciarsi con database Informix ed effettuare le normali operazioni necessarie alla movimentazione del magazzino;
- **App Customer Service**: Realizzazione di una app per il customer service, che permetta agli utenti di effettuare le richieste di assistenza o di anomalia anche

dallo smartphone. L'attuale software gestisce i tickets via web, con apertura di chiamata direttamente dalla propria area riservata. Attraverso questa app si vuole dare la possibilità all'utente di visualizzare tutte le richieste di assistenza, così come di crearne di nuove, anche in mobilità;

- **App Gestione Note spese**: Progettazione e sviluppo di una app mobile che permetta di gestire le informazioni relative alle note spese dei collaboratori. L'applicazione sincronizzerà i dati presenti sul dispositivo con l'archivio di gestione dell'ERP proprietario;

- **Project Manager**: Il Project Manager di Ergon vuole essere uno strumento per la pianificazione delle risorse e la distribuzione del carico lavoro in azienda, con lo scopo di organizzare al meglio task e compiti dei singoli. Una migliore pianificazione contribuirà ad una riduzione degli sprechi dovuti ad errate valutazioni e ad una maggiore puntualità nell'organizzazione interna. Allo stesso modo risorse meglio impiegate permetteranno all'azienda di essere tempestiva nei confronti dei clienti, rispondendo con più precisione alle loro richieste. Su più ampia scala il software sarà di aiuto per comprendere un eventuale fabbisogno di personale e sarà un valido strumento di gestione per il responsabile tecnico;

- **Pianificazione della produzione**: Il candidato dovrà integrare e potenziare l'algoritmo esistente, che consente di pianificare la produzione di un periodo temporale, razionalizzando l'occupazione delle linee e rispettando i tempi di spedizione degli ordini clienti. Il risultato dell'elaborazione indicherà gli articoli da produrre per soddisfare gli ordini clienti, programmando in quale giorno e fascia oraria realizzare le quantità necessarie. Il software dovrà inoltre, in presenza di eventi accidentali quali rottura della linea, guasto momentaneo, ordini dell'ultimo momento etc, trovare la soluzione ottimale ricalcolando l'impiego delle stesse.

1.1.5 Progetto scelto: Pianificazione della produzione

Dopo un'attenta valutazione delle varie proposte sopracitate, anche a seguito di varie discussioni con i responsabili dell'azienda, ho voluto scartare tutti i progetti che includessero lo sviluppo di un'applicazione mobile o servizio web. Questa scelta deriva dal fatto che ,anche in ambito scolastico, ho avuto ampie possibilità di cimentarmi in questi campi e il mio desiderio era di mettermi alla prova in un progetto che richiedesse nuove competenze. D'altro canto il progetto di Project Manager non aveva attirato la mia attenzione. Il progetto di Pianificazione della produzione invece mi ha colpito sin da subito, richiedendo nuove competenze in ambito teorico, soprattutto per quanto riguarda tecniche Ricerca Operativa applicate ad algoritmi di pianificazione. Anche se è richiesto di ampliare e migliorare un applicativo già sviluppato ciò non mi ha creato problemi in quanto le nuove aggiunte sarebbero state atomiche dal punto di vista delle funzionalità e avrebbero portato ad un consistente miglioramento dell'applicativo una volta portato a termine il periodo di stage.

1.1.6 Convenzioni tipografiche

Vengono qui definite le regole tipografiche adottate nella stesura del testo:

- le abbreviazioni, gli acronimi e i termini che possono risultare ambigui o facenti parte del linguaggio tecnico o troppo specifico vengono definiti in modo approfondito alla fine del seguente documento in una sezione chiamata glossario;
- la prima occorrenza dei termini sopracitati viene identificata da un g a pedice come segue: glossario$_G$;

Capitolo 2

Descrizione dello stage

2.1 Panoramica del progetto

L'applicativo in questione è già in attività da circa un anno e ha il compito di pianificare la produzione settimanale degli ordini richiesti dalle varie aziende. Ogni azienda definisce un insieme di linee sulle quali è possibile produrre i prodotti e fornisce anche i giorni e gli orari di lavoro.
Con queste informazioni il precedente programma era in grado di fornire una pianificazione distribuita sull'arco della settimana, valutando quale fosse l'ordinamento migliore e quali ordini scartare se il tempo a disposizione non fosse stato sufficiente. Questa logica di pianificazione ometteva però il controllo della disponibilità di materie prime e/o semilavorati, la quale veniva considerata sufficiente a produrre tutti gli ordini richiesti.
Non veniva inoltre considerato il tempo di produzione degli eventuali semilavorati richiesti, e non si considerava nemmeno l'arrivo di materiali per il magazzino da parte dei fornitori. Quello che mi è stato richiesto dunque è l'integrazione delle parti sopracitate.

2.2 Specifiche tecniche del problema

Nel primo periodo di stage ho speso il tempo che avevo a disposizione studiando le componenti del problema che dovevo affrontare, ricavando un quadro generale del funzionamento logico della pianificazione della produzione. L'ostacolo principale si è rivelato essere la complessità in ambito lavorativo reale della pianificazione.
Ho dovuto spendere del tempo assieme al tutor interno per riuscire ad entrare nel contesto in cui avrei dovuto lavorare, facendomi spiegare le varie sfaccettature e le scelte implementative prese.
Di seguito voglio fornire un'idea del problema e di come avviene la pianificazione di un singolo prodotto finito.
Perché un prodotto venga pianificato è necessario che ci sia almeno una linea in grado di produrlo, una volta definita la linea si deve scegliere una sequenza in cui produrre l'articolo. Una sequenza definisce giorno, data di inizio e di fine e può comprendere un insieme di vincoli da soddisfare (quali lavaggi oppure ordinaria manutenzione), l'articolo verrà quindi inserito all'interno della sequenza con il relativo tempo di produzione. Ogni linea ha solitamente più sequenze nelle quali è possibile collocare l'articolo. Ogni sequenza può avere dei vincoli di dipendenza con altre sequenze.

2.2.1 Implementazione

Segue la descrizione della struttura delle principali classi che vengono impiegate nell'applicativo.

Linee

Indica l'insieme delle linee sulle quali è consentita la produzione degli ordini, ogni linea ha relativo costo orario e definisce l'insieme degli articoli che può produrre con annessa velocità di produzione. Ogni linea è così definita:

- **Codice Linea**: serve ad indicare su quale linea si vuole produrre l'articolo;
- **Info Articolo**: lista che indica quali articoli la linea può produrre con relativo tempo di produzione;
- **Info Vincolo**: lista che indica i vincoli presenti sulla linea, possono essere obbligatori o opzionali;
- **Sequenze Linea**: lista che indica quali sono le sequenze di produzione della linea;
- **Pianificazione**: lista che indica quali articoli e quali vincoli sono stati pianificati nella linea corrente;
- **Errori**: indica quali errori si sono presentati durante la pianificazione;
- **Calendario**: indica quali sono i giorni lavorativi con le eventuali pause.

Sequenze

Indica l'insieme delle sequenze di produzione presenti su ogni linea, nelle quali vengono inseriti gli ordini che sono stati pianificati. Ogni sequenza è così definita:

- **Codice Sequenza**: serve ad indicare su quale sequenza si vuole inserire l'articolo;
- **Giorno**: indica il giorno nel quale si vuole inserire l'articolo;
- **Ora inizio/Ora fine**: indica gli orari di inizio e fine della sequenza corrente;
- **Elementi**: lista che indica gli articoli e i vincoli appartenenti alla sequenza corrente.

Ordine

Indica come si presenta un ordine da produrre. Ogni ordine è così definito:

- **Codice Articolo**: indica il codice dell'articolo;
- **Tipo Articolo**: indica il tipo dell'articolo, può essere un prodotto finito, semilavorato o materia prima;
- **Codice Linea**: indica il livello più generale della gerarchia di classificazione di un prodotto;
- **Codice Settore**: indica il terzo livello di gerarchia di classificazione;

- **Codice Famiglia**: indica il secondo livello di gerarchia, definisce la famiglia del prodotto;
- **Codice SottoFamiglia**: indica il livello più caratterizzante della classificazione di un prodotto;
- **Quantità**: indica la quantità richiesta da produrre;
- **Data Spedizione**: indica la data di spedizione dell'articolo;
- **Ora spedizione**: indica l'ora di spedizione dell'articolo;
- **Data Consegna**: indica la data di consegna presso la sede del cliente;
- **Linea preferenziale**: indica la linea preferenziale di produzione dell'articolo;
- **Anno Ordine**: indica l'anno dell'ordine in questione;
- **Materie Prime**: lista che indica l'insieme delle materie prime richieste per la produzione dell'ordine;
- **Semilavorati**: lista che indica l'insieme dei semilavorati richiesti per la produzione dell'ordine;
- **Riferimento Ordine**: lista che indica l'insieme degli ordini ai quali l'ordine corrente fa riferimento per la produzione di semilavorati;

2.2.2 Problemi logici da affrontare

Di seguito sono presentate le problematiche che l'algoritmo deve affrontare per ottenere una corretta pianificazione della produzione.

Temporali

Qui vengono descritti tutti i problemi riguardanti i tempi di produzione.

- **Data Spedizione**: devono essere rispettate le date di spedizione e di consegna degli articoli da pianificare;
- **Tempo minimo alla consegna**: devono essere rispettate le date di inizio produzione nel caso di ordini con scadenza a breve termine;
- **Semilavorati**: i semilavorati di un ordine devono essere pianificati prima dell'ordine stesso;
- **Ordini Fornitori**: vanno considerate le date di arrivo delle materie prime da parte dei fornitori in modo da non scartare la produzione di ordini che sarebbero producibili;
- **Giorni Lavorativi**: devono essere rispettati i giorni lavorativi senza pianificare ordini al di fuori di essi;
- **Orario Lavorativo**: devono essere rispettati gli orari lavorativi senza eccedere dal monte ore impostato dall'azienda che esegue la pianificazione;
- **Sovrapposizione Sequenze**: ogni linea può produrre una singola sequenza per volta;
- **Sovrapposizione Articoli**: ogni sequenza può contenere un solo articolo senza sovrapporne degli altri nello stesso lasso di tempo.

Vincoli

Qui vengono descritti tutti i vincoli da rispettare.

- **Materie prime**: non è possibile produrre un ordine se non sono presenti sufficienti materie prime;
- **Vincoli Obbligatori**: devono essere pianificati tutti i vincoli obbligatori di ogni sequenza;
- **Vincoli Condizionati**: devono essere pianificati tutti i vincoli condizionati di ogni sequenza in base alle condizioni imposte;
- **Vincoli Opzionali**: devono essere pianificati i vincoli opzionali solo in caso ci sia una finestra temporale sufficiente altrimenti si lascia spazio alla produzione di articoli;
- **Vincoli Articolo**: devono essere rispettati i vincoli di produzione di ogni articolo, quali linea preferenziale o linea con maggiore velocità di produzione.

Scelte implementative

Dopo una discussione col tutor interno abbiamo scartato l'ultimo vincolo facoltativo FA2, presente nel piano di lavoro, il quale riguardava la ripianificazione degli articoli in caso di guasti o necessità aziendali. Optando per una semplice nuova esecuzione dell'applicativo con i nuovi dati interessati.

2.3 Obiettivi

Gli obiettivi da raggiungere durante il progetto sono elencati di seguito, fanno riferimento a quanto riportato nella versione definitiva del piano di lavoro, con qualche aggiunta o modifica in sede di sviluppo, previa discussione con il tutor aziendale.
Per ogni obiettivo è definito il grado di completamento: nullo, parziale, totale.

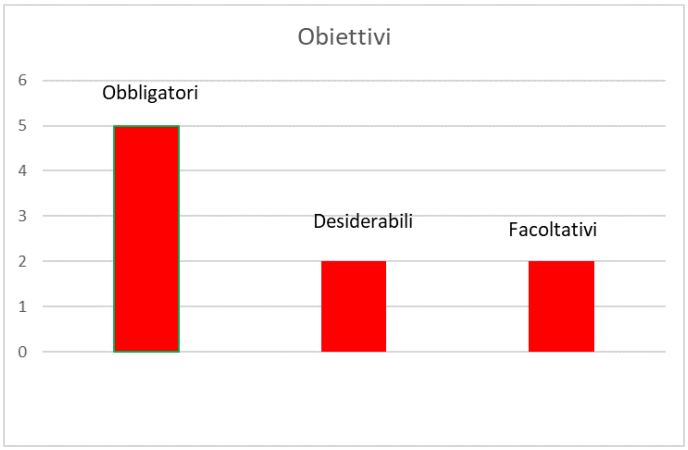


Figura 2.1: Requisiti

2.3.1 Requisiti obbligatori

- Ottimizzazione della pianificazione tenendo conto delle giacenze di magazzino: raggiungimento totale;
- Sviluppo di nuove strategie di scelta dell'Algoritmo Greedy e confronto dei risultati ottenuti in relazione alla funzione obiettivo: raggiungimento totale;
- Integrazione della Tabu Search con nuovi criteri di aspirazione e arresto: raggiungimento totale;
- Sviluppo di nuovi meccanismi di esplorazione del vicinato nella Tabu Search: raggiungimento totale;
- Acquisizione di competenze sull'utilizzo di algoritmi di Ricerca Operativa e applicazione in un caso di studio reale: raggiungimento totale.

2.3.2 Requisiti desiderabili

- Ottimizzazione della pianificazione tenendo conto degli ordini a fornitore presenti a sistema: raggiungimento totale;
- Ottimizzazione della pianificazione tenendo conto dei semilavorati: raggiungimento totale;

2.3.3 Requisiti opzionali

- Utilizzo di multithreading nelle fasi in cui è richiesta una maggiore capacità di calcolo: raggiungimento totale;
- Data una pianificazione inserita a sistema, eseguire una ripianificazione considerando vincoli dovuti a necessità aziendali dell'ultimo momento (anomalie, guasti, ordini urgenti, etc.): raggiungimento nullo.

Come accennato nella sezione precedente il requisito FA2 "Data una pianificazione inserita a sistema, eseguire una ripianificazione considerando vincoli dovuti a necessità aziendali dell'ultimo momento" è stato scartato a seguito di una riunione con il tutor aziendale.
Siamo quindi giunti alla conclusione che, in termini di efficacia, è preferibile una nuova esecuzione dell'algoritmo a partire da zero piuttosto che conservare gli attuali dati e procedere con una nuova rielaborazione. Questo rende anche più facile la definizione e l'aggiunta dei nuovi vincoli imposti, quali linee bloccate o sequenze non più attive, che potrebbero influenzare negativamente i precedenti dati inseriti.

2.4 Divisione settimanale

La seguente sezione vuole descrivere come sono state suddivise le 320 ore previste per il periodo di stage, affiancando ad ogni settimana lavorativa i requisiti e gli obiettivi raggiunti.
Il periodo di stage inizia in data 09-settembre-2019 con termine ufficiale in data 01-novembre-2019 traslata al 03-novembre-2019 a causa del sostenimento di due prove d'esame.

- **Prima settimana**:
  - analisi del modulo software esistente;
  - funzionalità da realizzare;
  - studio della documentazione disponibile dell'algoritmo esistente.
- **Seconda settimana**:
  - analisi dei rischi;
  - stesura dell'analisi dei requisiti che comprende le nuove funzionalità da integrare nel software esistente;
  - inizio della stesura di documentazione a supporto dell'architettura utilizzata.
- **Terza settimana**:
  - Studio delle tecnologie aziendali necessarie allo sviluppo del modulo in particolare linguaggio di programmazione VB.NET$_G$, componenti DevExpress$_G$ e database Informix$_G$;
  - training$_G$ sulle nuove tecnologie da utilizzare con realizzazione di un software di prova;
  - studio di algoritmi e tecniche di Ricerca Operativa e Ottimizzazione Combinatoria;

  - riunioni col tutor aziendale per decidere le scelte implementative dell'algoritmo Greedy.

- **Quarta settimana**:

  - Algoritmo Greedy: sviluppo di nuove strategie di scelta del passo successivo adottato dall'algoritmo nella costruzione della soluzione del problema;
  - sviluppo procedura di gestione delle giacenze di magazzino.

- **Quinta settimana**:

  - sviluppo procedura di gestione dei semilavorati pianificati;
  - sviluppo procedura di gestione degli ordini fornitori.

- **Sesta settimana**:

  - integrazione della Tabu Search con nuovi meccanismi: criteri di aspirazione e arresto, variazione dei meccanismi di esplorazione del vicinato e adozione di tecniche di intensificazione e diversificazione.

- **Settima settimana**:

  - fase di test dei dati con valori reali forniti dai clienti di Ergon;
  - comparazione dell'algoritmo ultimato con il precedente algoritmo in uso;
  - verifica della bontà della soluzione in rapporto coi dati forniti dai clienti.

- **Ottava settimana**:

  - stesura della documentazione a supporto del prodotto sviluppato, con risalto sulle scelte implementative non banali.

Da sottolineare il fatto che le fasi di test sono state molteplici e non solo durante la settima settimana. Ad ogni nuova aggiunta veniva effettuato un controllo sul risultato prodotto dalla pianificazione con dei dati reali di supporto, tutto ciò per garantire la correttezza della logica del codice in aggiunta. In comune accordo con il tutor abbiamo deciso di iniziare dalle parti più complesse la nuova fase di implementazione, così da garantire una continua verifica ad ogni aggiunta delle funzionalità meno importanti.

2.5 Analisi dei rischi

Ho trovato importante individuare eventuali rischi che possono portare a problematiche in grado di far procedere a rilento la realizzazione del progetto di stage.
Di seguito viene presentata la tabella contenente i rischi preventivati durante la seconda settimana di stage. Ogni rischio possiede un codice identificativo, una breve descrizione affiancata dal suo rilevamento e relativo grado di rischio. Per ogni rischio è definito inoltre un piano di contingenza da seguire in caso di occorrenza del rischio.

Tabella 2.1: Tabella dei rischi di progetto

Codice Nome	Descrizione	Piano di contingenza	Grado di rischio
RO1 Problematiche di Ricerca Operativa	Dovendo svolgere uno studio individuale delle tematiche di Ricerca Operativa da affrontare, è possibile imbattersi in argomenti poco intuitivi o complicati da apprendere in solitaria.	Vengono fornite le dispense riguardanti il contesto in cui si andrà a lavorare e in caso di incomprensioni si farà affidamento sia sul tutor interno sia al proprio relatore.	Occorrenza: **Media** Pericolosità: **Alta**
RT1 Inesperienza tecnologica	Alcune delle tecnologie adottate sono nuove, pertanto è possibile incorrere in problemi durante lo svolgimento delle attività che le coinvolgono.	Viene fornita la documentazione ufficiale in modo da avere ampio supporto per qualsiasi lacuna di natura tecnica. Se ciò non dovesse bastare si farà affidamento al tutor interno o qualche membro delegato del team di sviluppo di Ergon.	Occorrenza: **Media** Pericolosità: **Media**
RT2 Scelte implementative	Non è detto che tutte le scelte prese in considerazione durante lo studio del problema portino ad una corretta soluzione o ad una esecuzione in tempi accettabili.	Verranno valutate, in caso di necessità, nuove strade da percorrere per la risoluzione del problema.	Occorrenza: **Bassa** Pericolosità: **Alta**

Capitolo 3

Progetto di stage

3.1 Analisi dei requisiti

Prima di iniziare a lavorare sul progetto, il tutor aziendale, mi ha fornito la documentazione esistente dell'applicativo, in particolar modo ho fatto riferimeto al manule utente creato in precedenza e all'analisi dei requisiti. Mi è stato espressamente chiesto inoltre di non eseguire alcuna modifica a livello grafico dell'applicativo in quanto la clientela era abituata alla versione corrente e non sempre è immediato addattarsi a cambiamenti di questo tipo.
Ho infine aggiunto due casi particolari che verranno evidenziati di seguito per quanto riguarda l'analisi dei requisiti esistente.

3.1.1 Casi d'uso

Per rappresentare l'insieme delle azioni comuni ad un singolo utente sono stati utilizzati i casi d'uso. Per descrivere i vari scenari ho mantenuto la precedente scaletta di informazioni così da dare una continuità al documento esistente. Ciascun caso d'uso sarà quindi rappresentato come descritto di seguito:

- **Identificatore** univo nel formato UC-[Codice];
- **Titolo** del caso d'uso;
- **Descrizione** generale del caso d'uso;
- **Attori** coinvolti: primari e secondari;
- **Precondizione** del caso d'uso;
- **Scenario principale** che il caso d'uso vuole modellare, identificando ogni azione che ne fa parte;
- **Postcondizione** del caso d'uso;
- **Estenzioni** dello scenario principale, evidenziata da una lettera maiuscola e inserita in un elenco.

Di seguito vengono rappresentate le modifiche apportate al documento esistente. Si fa riferimento allo standard UML 2.0 per la rappresentazione dei casi d'uso.

UC-3 Pianificazione della produzione

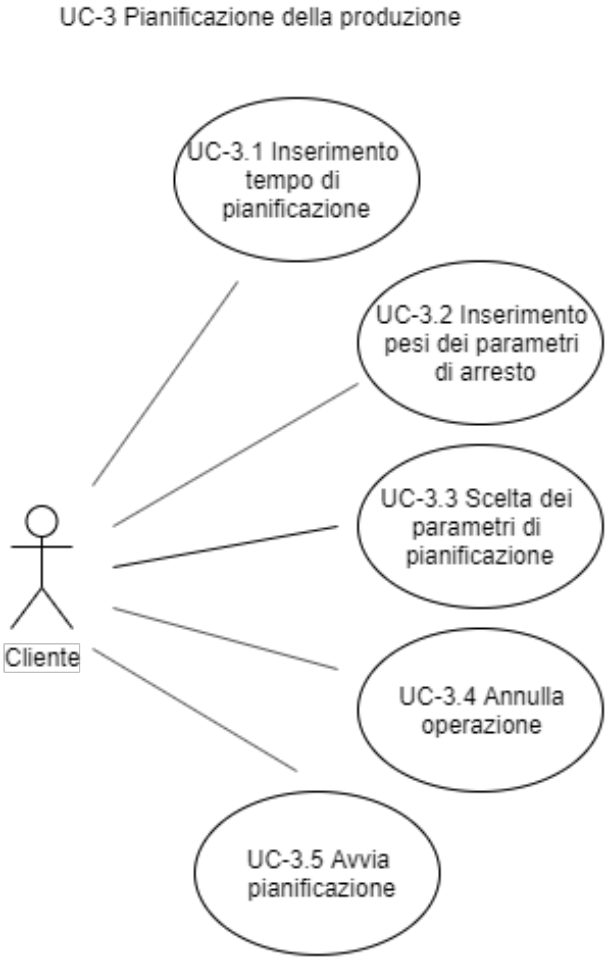


Figura 3.1: Use Case - UC-3

Attori: Cliente.
Descrizione: Il cliente esegue la pianificazione degli ordini selezionati.
Precondizione: Il cliente ha selezionato gli ordini che intende pianificare e i giorni in cui vuole farlo.
Scenario principale:

- Il cliente seleziona che ordini vuole pianificare;
- Il cliente preme il pulsante "Pianifica";
- Il cliente modifica i parametri della pianificazione come previsto dagli UC-3.1, UC-3.2, UC-3.3;
- Il cliente preme il pulsante "Avvia pianificazione".

Postcondizione: Nella finestra principale viene visualizzato il risultato della pianificazione.

UC-3.3 Scelta dei parametri di pianificazione

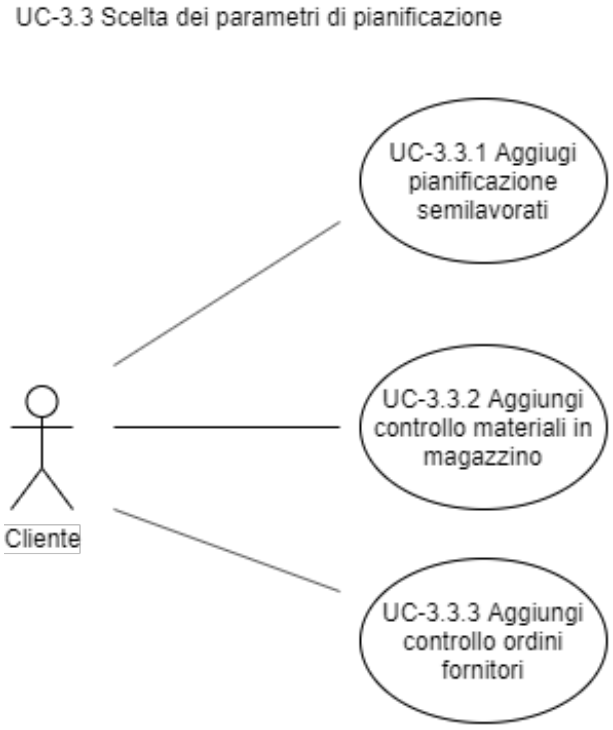


Figura 3.2: Use Case - UC-3.3

Attori: Cliente.
Descrizione: Il cliente seglie i parametri da considerare nella pianificazione.
Precondizione: Il cliente ha selezionato il pulsante "Metodo di pianificazione".
Scenario principale:

- Il cliente seleziona come effettuare le pianificazione;
- Il cliente può selezionare le varie aggiunte descritte dagli UC-3.3.1, UC-3.3.2, UC3.3.3;
- Il cliente preme il pulsante "Conferma".

Postcondizione: Si verrà ricondotti allo use case UC-3.3.

3.2 Tracciamento dei requisiti

Ogni requisito è composto dalla seguente struttura:

- **codice identificativo**: ogni codice identificativo è univoco e conforme alla seguente codifica:

  R[Importanza][Tipologia][Codice]

  Il significato delle cui voci è:

  - **Tipologia**: ogni requisito può assumere uno dei seguenti valori:
    - *F*: funzionale;
    - *P*: prestazionale;
    - *Q*: qualitativo;
    - *V*: vincolo.
  - **Importanza**: ogni requisito può assumere uno dei seguenti valori:
    - *O*: requisito obbligatorio: irrinunciabili per qualcuno degli stakeholder;
    - *D*: requisito desiderabile: non strettamente necessari ma a valore aggiunto riconoscibile;
    - *F*: requisito facoltativo: relativamente utili oppure contrattabili più avanti nel progetto.
  - **Codice**: è un identificatore univoco del requisito segue un ordine incrementale.

- **classificazione**: viene riportata l'importanza del requisito. Sebbene questa sia un'informazione ridondante ne facilita la lettura;

- **descrizione**: descrizione breve ma completa del requisito, meno ambigua possibile.

3.2.1 Requisiti funzionali

Tabella 3.1: Tabella dei requisiti funzionali

Requisito	Classificazione	Descrizione
RFO1	Obbligatorio	Il sistema permette l'inserimento di un nuovo vincolo di linea
RFO2	Obbligatorio	Il sistema permette la modifica dei dati di un vincolo di linea esistente
RFO3	Obbligatorio	Il sistema permette l'eliminazione di un vincolo di linea
RFO4	Obbligatorio	L'interfaccia permette l'eliminazione di una linea

Tabella 3.1: (continua)

Requisito	Classificazione	Descrizione
RFO5	Obbligatorio	Il sistema permette l'aggiunta di una nuova sequenza
RFO6	Obbligatorio	Il sistema permette la modifica dei dati di una sequenza esistente
RFO7	Obbligatorio	Il sistema permette l'eliminazione di un articolo di sequenza esistente
RFO8	Obbligatorio	Il sistema permette l'eliminazione di un vincolo di sequenza esistente
RFO9	Obbligatorio	Il sistema permette l'eliminazione di una sequenza
RFO10	Obbligatorio	Il sistema permette all'utente di scegliere un tempo massimo di esecuzione dell'algoritmo
RFO11	Obbligatorio	Il sistema permette all'utente di assegnare un peso all'importanza di parametri di produzione quali:pezzi prodotti e occupazione delle linee
RFD1	Desiderabile	Il sistema permette all'utente di scegliere se ricalcolare la pianificazione già esistente
RFO12	Obbligatorio	Il sistema permette di pianificare automaticamente la produzione dato un insieme di ordini e un intervallo di tempo
RFO1	Obbligatorio	Il sistema permette di scegliere i relativi criteri di pianificazione

3.2.2 Requisiti qualitativi

Tabella 3.2: Tabella dei requisiti qualitativi

Requisito	Classificazione	Descrizione
RQD1	Desiderabile	Il programma genera un file di Log_G di supporto al Debug_G
RQO1	Obbligatorio	Ogni scelta non banale effettuata è adeguatamente commentata nel codice
RQF1	Facoltativo	Fornire un manuale utente
RQF2	Facoltativo	Fornire un manuale sviluppatore

3.2.3 Requisiti prestazionali

Tabella 3.3: Tabella dei requisiti prestazionali

Requisito	Classificazione	Descrizione
RPD1	Desiderabile	L'applicativo deve fornire una soluzione entro il tempo stabilito dall'utente
RPD2	Desiderabile	L'applicativo deve fornire la soluzione nel modo più rapido possibile senza valutare ulteriori combinazioni se è stato raggiunto un determinato punteggio stabilito per la soluzione stessa
RPD3	Desiderabile	La fase di lettura dei dati deve essere eseguita in tempi inferiori ai 30 minuti

3.2.4 Requisiti vincolo

Tabella 3.4: Tabella dei requisiti prestazionali

Requisito	Classificazione	Descrizione
RVO1	Desiderabile	Rendere compatibile il programma con l'attualemodulo di pianificazione della produzione
RVO2	Desiderabile	Ottimizzare la pianificazione tenendo conto delle materie prime e dei semilavorati
RVO3	Desiderabile	Ottimizzare la pianificazione tenendo conto degli ordini dei fornitori inseriti a sistema
RVO4	Desiderabile	Ottimizzare la pianificazione rendendo possibile la pianificazione dei semilavorati
RVO5	Desiderabile	Il programma è sviluppato tramite il linguaggio diprogrammazione Vb.NET v.4.7.0
RVO6	Desiderabile	L'IDE$_G$ utilizzato è Microsoft Visual Studio$_G$ v.10.0.4
RVO7	Desiderabile	Le componenti grafiche sono basate sulle DevEx-press$_G$
RVO8	Desiderabile	I dati vengono salvati su un database Informix
RVO9	Desiderabile	I dati in input sono scritti su un file JSON
RVO10	Desiderabile	I dati in output sono scritti su un file JSON

3.2.5 Riepilogo dei requisiti

Tabella 3.5: Tabella del riepilogo dei requisiti

Tipo	Obbligatori	Desiderabili	Facoltativi
Funzionali	13	1	0
Qualitativi	1	1	2
Prestazionali	0	3	0
Di vincolo	10	0	0
Totali	24	5	2

3.3 Tecnologie e strumenti

Di seguito sono riportate tutte le tecnologie utilizzate durante lo sviluppo del progetto. Tali scelte sono state imposte da Ergon Informatica in quanto sono gli strumenti che vengono utilizzati dall'azienda per lo sviluppo dei progetti interni.
La scelta è inoltre guidata dal fatto che, per mantenere l'integrazione con le parti già esistenti dell'applicativo e il suo funzionamento tramite il software Ergdis, non si è potuto spostarsi su nuove tecnologie, magari anche solo a qualche versione successiva di esse.
Ergon Informatica si appoggia a questi strumenti in quanto sono forniti di un ottima documentazione di supporto, hanno un alto tasso di scalabilità e forniscono un supporto in tempo reale alla codifica.

Vb.NET

Figura 3.3: Logo Vb.NET

Linguaggio di programmazione che deriva da Visual Basic 6, è sviluppato da Microsoft e, al contrario dei precedenti, supporta il paradigma di programmazione orientata agli oggetti. Vb.NET consente lo sviluppo di applicazioni Windows, Web e per dispositivi mobili.
Come avviene con tutti i linguaggi basati su Microsoft .NET Framework, i programmi scritti in Vb.NET usufruiscono delle funzionalità di sicurezza e interoperabilità dei linguaggi.
Nel progetto viene utilizzato questo linguaggio per poter usufruire delle classi, le quali riducono significativamente le duplicazioni di codice e rendono l'intero sorgente chiuso alle modifiche e aperto agli ampliamenti.
Viene inoltre utilizzato in quanto è il linguaggio attualmente adottato dall'azienda per lo sviluppo e si presta bene al tipo di lavorazioni necessarie a portare a compimento lo sviluppo dell'applicativo.

Visual Studio 2010

Figura 3.4: Logo Visuali Studio 2010

Ambiente di sviluppo integrato sviluppato da Microsoft. La prima versione risale al 1997 con lo scopo di fornire un ambiente di sviluppo grafico ed ed integrato che aiutasse lo sviluppatore a gestire i progetti in maniera semplice, ma efficace, aumentandone quindi la produttività.
Microsoft ha incluso il supporto a differenti linguaggi di programmazione.
In Ergon Informatica è il principale ambiente di sviluppo sia per applicazioni desktop che mobile, di conseguenza è stato utilizzato anche durante lo sviluppo del progetto.

DevExpress

Figura 3.5: Logo DevExpress

Developer Express Inc. è una società di sviluppo software fondata nel 1998, inizialmente fornisce un insieme di controlli UI_G poi sviluppa diverse estensioni per le librerie grafiche. In particolare nel progetto ci si è appoggiati a questa tecnologia per velocizzare la creazione dell'interfaccia grafica sulla quale si basa il modo di rappresentare la soluzione che viene fornita, permettendo una chiara visualizzazione tabellare grazie ad una delle tante $estensioni_G$ utilizzabili.

JSON

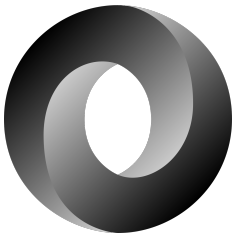


Figura 3.6: Logo JSON

Acronimo di *JavaScript Object Notation*, è un formato adatto all'interscambio di dati fra applicazioni client/server$_G$. Serve, in particolare, a fornire una struttura a dati interessati rendendoli interscambiabili tra diverse applicazioni senza che si renda necessaria la codifica e decodifica di quest'ultimi. Nel nostro caso si rende utile quando è necessario delegare l'esecuzione di operazioni sui dati prelavati da un terminale con accesso al database verso un terminale esterno. I dati raccolti dal database venivano trasferiti in un file JSON, il quale veniva inviato al successivo passo di esecuzione dell'algoritmo, al termine di ciò veniva restituita la soluzione sempre in formato JSON a chi ne aveva fatto richiesta e di conseguenza visualizzata.

Informix

Figura 3.7: Logo Informix

Fa parte della divisione DBMS$_G$ di IBM$_G$, è un sistema software progettato per consentire la creazione, la manipolazione e l'interrogazione efficiente di database. L'Informix server supporta il modello relazionale ad oggetti che permette ad IBM di offrire estensioni che supportano i tipi di dati che non sono una parte dello standard SQL$_G$. Le estensioni più usate sono quelle riguardanti le serie temporali e spaziale che forniscono entrambe supporto al tipo di dato ed estensioni linguistiche che permettono interrogazioni per un dominio specifico ad alte prestazioni e archiviazione efficiente per set di dati basati su serie temporali e dati spaziali. È il servizio database sul quale fa affidamento l'azienda essendo anche uno dei partner principali.

Capitolo 4

Progettazione e sviluppo

4.1 Progettazione

4.1.1 Architettura

Prima di descrivere la varie aggiunte che sono state apportate al software è necessario fornire un'idea dell'architettura sulla quale il progetto si basa.In questo modo si rendono più chiare molte delle scelte che sono state prese e il perchè altre sono state scartate. Di seguito è presentato uno schema ad alto livello di come sono strutturate le varie componenti che formano l'intero sistema. Come possiamo notare dalla figura sottostante, il progetto segue un flusso pressoché "ciclico", partendo dal recupero dei dati dal database e infine fornendo la soluzione.
I dati estratti dal database vengono convertiti in file JSON il quale verrà letto dall'algoritmo che ha il compito di trovare la soluzione, una volta fornita la soluzione essa viene ritradotta in un file JSON il quale viene letto e trasposto nell'interfaccia utente.

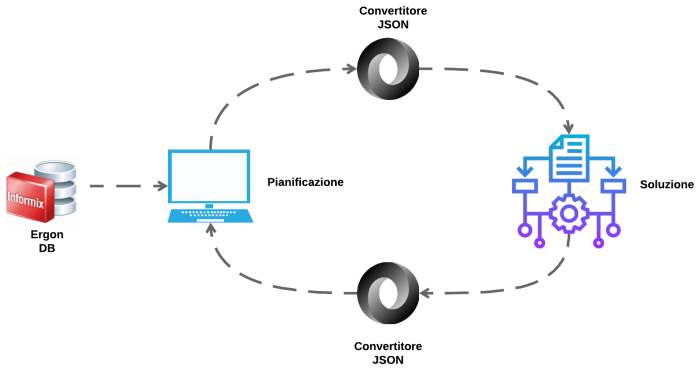


Figura 4.1: Architettura generale

4.1.2 Algoritmo

Il problema

Dovendo integrare delle nuove funzionalità al programma esistente, si è prima effettuata un analisi del problema esistente per quanto riguarda la pianificazione della produzione, in modo da poter comprendere alcune delle scelte euristiche$_G$ che sono state effettuate in seguito.
Il suddetto problema rientra nella categoria NP-hard$_G$ in quanto è riconducibile al problema del commesso viaggiatore$_G$. A tali problemi non è possibile far corrispondere una soluzione ottima ma, al massimo, la miglior approssimazione possibile di una soluzione ammissibile. In aggiunta al problema del commesso viaggiatore si devono considerare altri vincoli come:

- tempo massimo sul quale si vogliono pianificare gli ordini;
- numero di linee sulle quali si possono produrre determinati ordini;
- vincoli generali di sequenza che andremo a definire in seguito.

Questo è l'insieme di problematiche riguardanti la precedente fase del progetto, dove rientrava solo la pianificazione dei prodotti finiti. Con la richiesta di estendere tale progetto, il numero di vincoli da soddisfare cresce in proporzione al numero di ordini che si vogliono pianificare. In particolare vanno tenuti in considerazione i vincoli precedenti più l'aggiunta dei seguenti:

- pianificazione dei semilavorati antecedente ai prodotti finiti;
- materie prime e semilavorati presenti in quantità sufficienti per la produzione;
- verifica della disponibilità di materie prime e semilavorati in base agli ordini fornitori.

In conclusione, il problema che il progetto si pone di risolvere, consiste nel trovare una soluzione ammissibile, quanto più vicina all'ottimo, della pianificazione in base ai vincoli imposti.

La soluzione

Dovendo appoggiarsi all'algoritmo esistente, non è stato possibile eseguire uno stravolgimento completo del suo funzionamento, anzi si è cercato di inserire le nuove funzionalità in modo da integrarle il più possibile senza effettuare modifiche sostanziali alla struttura del codice esistente.
Per fare ciò si è fatto ampio uso di nuove funzioni e classi, per poter delegare, dalla base esistente, l'esecuzione di nuovi moduli di pianificazione.
La struttura di partenza sulla quale vengono integrati i nuovi moduli è composta da un iniziale fase di lettura dei dati dal database. Tali dati vengono convertiti in un file JSON che, per comodità di esecuzione, viene letto nuovamente dallo stesso programma, anche se l'dea di base sarebbe quella di delegare la sua lettura e successiva elaborazione dei dati ad un sistema centralizzato esterno all'azienda che fornisce i dati stessi.
Una volta ottenuti i dati in formato JSON, vengono inseriti nell'algoritmo tramite l'impiego di classi e strutture predisposte. Tali classi vengono utilizzate come parametri per la prima funzione che si pone come obiettivo quello di fornire una soluzione ammissibile nel minor tempo possibile ovvero l'Algoritmo Greedy$_G$.
Tale algoritmo ha il compito di valutare, in modo sequenziale, i vari ordini che necessitano di essere pianificati, se un ordine soddisfa i vincoli imposti allora verrà inserito

nella soluzione iniziale. Una volta ottenuta la soluzione iniziale, questa viene fornita alla successiva funzione di ottimizzazione, composta da un algoritmo di Tabu Search$_G$. Tale algoritmo ha il compito di eseguire un insieme di mosse specifiche in modo casuale, valutando di volta in volta la bontà della soluzione ottenuta dopo la sua esecuzione. Al termine di un numero prefissato di iterazioni, o dopo aver soddisfatto i criteri di aspirazione, verrà fornita la soluzione ottimizzata da visualizzare.
Visto che le scelte implementative non sempre possono essere chiare o si potrebbe pensare di utilizzare una metodologia diversa, viene illustrato di seguito il funzionamento degli algoritmi su cui fa affidamento la pianificazione della produzione.

Algoritmo Greedy
È il primo passo per ottenere una soluzione ammissibile del problema in questione. Tale algoritmo è sviluppato tenendo in considerazione che non è possibile effettuare dei "passi indietro", ovvero, una volta inserito un ordine in pianificazione tale ordine rimane fino alla fine dell'esecuzione. I parametri in ingresso di tale algoritmo sono i seguenti:

- ordini da pianificare;
- materie prime e semilavorati presenti in magazzino;
- materie prime e semilavorati presenti in ordini fornitori;
- linee di lavorazione con i rispettivi vincoli.

L'ordine di funzionamento è quindi il seguente:

- prima scelta euristica, vengono ordinati i prodotti in base alla loro data di spedizione;
- il primo ordine in questione viene valutato;
- si verifica la presenza di materie prime, e qui abbiamo i seguenti casi:
  - le materie prime sono disponibili per produrre l'ordine, inserisco in pianificazione;
  - le materie prime non sono disponibili per produrre l'ordine, verifico l'arrivo di eventuali materie prime da parte dei fornitori, se ciò avviene inserisco in pianificazione;
  - non pianifico l'ordine.
- si verifica la presenza di semilavorati, e qui abbiamo i seguenti casi:
  - i semilavorati sono disponibili per produrre l'ordine, inserisco in pianificazione;
  - i semilavorati non sono disponibili per produrre l'ordine, provo ad eseguire la pianificazione dei semilavorati mancanti, se ciò avviene inserisco in pianificazione;
  - non pianifico l'ordine.
- ogni valutazione effettua un controllo se è possibile inserire l'ordine in base al tempo lavorativo rimanente;

- si ripetono le azioni sopracitate fino al termine della valutazione di tutti gli ordini da pianificare.

Si ricorda che non è necessario che tutti gli ordini richiesti siano pianificati, in quanto uno dei precedenti vincoli può non essere soddisfatto e, se tale ordine venisse inserito nella soluzione, quest'ultima sarebbe non ammissibile perché viola appunto almeno uno dei vincoli imposti. Inoltre, se si esegue un accurato controllo, si può notare che il risultato fornito al termine della sua esecuzione, in alcuni casi, ha un ampio margine di miglioramento. L'algoritmo Greedy ha come obiettivo quello di creare una soluzione iniziale nel modo più rapido possibile, in modo da ottenere una base di partenza per la successiva ottimizzazione eseguita dall'algoritmo Tabu Search.

Tabu Search

Una volta ottenuta la prima soluzione ammissibile, fornita dall'algoritmo Greedy, l'esecuzione si sposta sulla sua ottimizzazione tramite la Tabu Search. Questa particolare tecnica meta-euristica$_G$ ha lo scopo di fornire una soluzione migliore, più vicina all'ottimo, tramite l'esecuzione di un insieme di mosse ben definite in modo casuale. L'esecuzione di queste mosse in modo casuale porta ogni volta ad una nuova soluzione ammissibile, tale risultato viene poi confrontato con la soluzione precedente, se il risultato è migliorativo allora verrà considerato come nuova miglior soluzione e si utilizzerà per la successiva ottimizzazione, altrimenti tale risultato verrà semplicemente scartato e si prosegue con l'esecuzione della mossa successiva. Di seguito è presentato l'ordine di funzionamento di tale algoritmo:

- selezione casuale della mossa da eseguire tra le seguenti:
  - mossa aggiungi articolo;
  - mossa sposta articolo;
  - mossa scambia articoli;
  - mossa elimina articolo;
  - mossa sposta sequenza;
  - mossa scambia sequenza.
- validazione della mossa scelta;
- esecuzione della mossa scelta;
- valutazione della nuova soluzione in rapporto con la soluzione precedente;
- viene ripetuta l'esecuzione nell'ordine sopracitato fino al termine del numero massimo di iterazioni, del tempo imposto o del soddisfacimento di uno dei criteri di aspirazione.

Scelta della mossa

È utile entrare maggiormente nel dettaglio dei punti precedenti per avere un'idea di come si ottiene la soluzione ottimizzata.
Al punto uno, la scelta della mossa, viene eseguita inizialmente in modo casuale, sottolineo inizialmente in quanto, uno degli obiettivi che mi sono stati richiesti, riguardava l'impiego di alcuni metodi di diversificazione$_G$ o intensificazione$_G$. Tali metodi si applicano ad algoritmi di questo tipo per "forzare" la successiva mossa da scegliere,

quindi non rendendola più completamente casuale. Nel caso dell'intensificazione si punta a scegliere l'insieme di mosse che, ad esempio, nelle loro esecuzioni precedenti hanno fornito risultati migliori e quindi si pensa potranno fornire migliori risultati anche in futuro, in questo modo ci si concentra maggiormente su queste scartando le altre. Al contrario, la diversificazione, può decidere di eseguire le mosse che sono state meno eseguite in precedenza così da indirizzare i risultati verso strade mai percorse.
È proprio quest'ultimo caso che, a seguito di una discussione con il tutor aziendale, ho inserito nel programma esistente. Il suo funzionamento è semplice ma non banale: dopo l'esecuzione di metà delle iterazioni concesse alla Tabu Search entra in gioco tale meccanismo, ho deciso di considerare l'esatta metà così da poter ottenere inizialmente una soluzione meno forzata possibile, da tale momento in poi verrà assegnato un peso percentuale ad ogni mossa che diminuirà ad ogni esecuzione della stessa, aumentando quindi il peso delle rimanenti.
Per fare questo ho utilizzato un array di 100 elementi nel quale, inizialmente, ogni 16 posizioni contigue è presente una mossa specifica,rappresentata dal relativo numero, fino al suo totale riempimento. Di conseguenza ogni mossa avrà un peso percentuale di circa 16 e ora la scelta casuale non si effettua più scegliendo tra le 6 mosse ma scegliendo casualmente un elemento dell'array che corrisponde ad una mossa.
Nel caso in cui venga scelta la mossa *elimina articolo* corrispondente al numero 6, tale numero verrà sostituito con un altro numero casuale appartenente alle altre mosse restanti. In questo modo il peso percentuale di *elimina articolo* diminuirà mentre un altra mossa otterrà un relativo aumento. Si è scelto di utilizzare questo metodo per poter garantire comunque un alto grado di casualità, criterio su cui fa affidamento la tecnica di Tabu Search.

Validazione della mossa scelta
Anche in questo caso è utile approfondire come avviene la validazione di una mossa scelta in modo casuale. In particolare, in alcuni casi, possiamo direttamente scartare l'esecuzione della mossa perché siamo sicuri che porterà ad una soluzione non migliorativa.
Ogni algoritmo di Tabu Search fa affidamento su una relativa Tabu List$_G$, la quale non è altro che uno storico delle mosse eseguite in precedenza, le quali non dovranno essere rieseguite, almeno nell'immediato futuro.
Un criterio importante da tenere presente è la lunghezza che si sceglie di dare a tale lista, questo perché una lista troppo corta non garantisce che non vengano eseguite mosse utilizzate da poco tempo, mentre, una lista troppo lunga tende a creare troppa diversificazione delle soluzioni.
Nel mio progetto alla Tabu List è stata assegnata una lunghezza di 500, considerando anche il fatto di salvare sia la mossa che ha portato ad una nuova soluzione, sia la sua mossa opposta, in modo da non tornare casualmente alla soluzione precedente. La validazione di una mossa, in seguito, deve inoltre sottostare a dei vincoli che appartengono alla realtà sulla quale si basa il nostro applicativo. Ad esempio se viene considerata la mossa *aggiungi articolo* ma non abbiamo nessun articolo da aggiungere tale mossa viene scartata.
Con l'aggiunta della pianificazione dei semilavorati, i quali ricordiamo avere le stesse peculiarità dei prodotti finiti, sono stati aggiunti ulteriori controlli, in quanto non possiamo garantire l'eliminazione tramite la mossa *elimina articolo* di uno di questi senza eliminare anche l'articolo a cui fanno riferimento.
Perciò si è scelto di eseguire l'eliminazione solo sui prodotti finiti. Altri controlli da eseguire riguardano lo spostamento di articoli o sequenze, nel caso venga selezionato lo

spostamento di un semilavorato bisogna accertarsi che non venga inserito in pianificazione successivamente al relativo prodotto finito. Allo stesso modo è stato necessario aggiungere controlli sulla giacenza e sull'arrivo di merci dai fornitori, questo perché lo spostamento, l'eliminazione e l'aggiunta di articoli creava una forte oscillazione nei dati riguardanti i materiali da lavorazione e spesso riconduceva ad una soluzione peggiorativa.
Come si può intuire l'aggiunta di tutti questi vincoli ha minato la capacità di ottenere una soluzione nettamente migliorativa da parte della Tabu Search.

Esecuzione della mossa scelta
Una volta accertati che la mossa sia eseguibile si passa alla sua implementazione.
Ogni mossa è correlata ad un insieme di funzioni che hanno il compito di accertarsi che ad ogni passo della sua esecuzione ci si trovi all'interno di una soluzione ammissibile.
È facile riscontrare che, durante l'esecuzione di una di queste mosse, si possa terminare in una soluzione non ammissibile, in questo caso viene scartato il nuovo risultato e si torna alla soluzione ammissibile precedente.
Nel caso in cui siano soddisfatti tutti i vincoli imposti al termine dell'esecuzione di una mossa la soluzione ottenuta sarà ammissibile e sarà compito del prossimo passo dell'algoritmo accertarne la bontà.

Valutazione della soluzione
Una volta ottenuta un nuova soluzione è necessario valutarla per sapere se sia migliorativa o peggiorativa rispetto alla precedente.
Il confronto quindi viene eseguito calcolando il punteggio della nuova soluzione il quale è dato dalla seguente funzione:

$$\frac{\text{Pezzi pianificati}}{\text{Totale pezzi richiesti}} * \text{Peso del rapporto pezzi}$$

Qui otteniamo il punteggio di maggior spessore, dal quale si sottraggono i seguenti:

$$\frac{\text{Occupazione linee}}{\text{Massima occupazione linee}} * \text{Peso del rapporto occupazione}$$

$$\frac{\text{Tempo totale di pianificazione}}{\text{Massimo tempo a disposizione}} * \text{Peso del rapporto tempo}$$

Una volta ottenuto il relativo punteggio viene eseguito il confronto con la soluzione precedente e si sceglie la soluzione avente il punteggio più alto.
Questo non è l'unico metodo con il quale si sceglie la soluzione che riteniamo essere migliore.
Vengono impiegati dei criteri di aspirazione$_G$ i quali hanno il compito di valutare anche soluzioni che hanno punteggi inferiori alla precedente ma che potrebbero essere un buon punto di partenza per la valutazione di soluzioni successive.
Nel progetto è stato inserito un criterio di aspirazione con il seguente funzionamento: se la nuova soluzione ha un punteggio inferiore alla soluzione precedente, ma il numero di pezzi prodotti è superiore, allora se la differenza è inferiore al 2% la nuova soluzione

è considerata migliore. Questo perché lo scopo ultimo della pianificazione è riuscire a pianificare il maggior numero di ordini possibili.

4.1.3 Codifica

In questa sezione vengono illustrati i vari passi che ho seguito durante la fase di codifica del progetto, specificando i punti fondamentali di ognuno.

Il primo obiettivo che ho dovuto raggiungere riguardava l'esecuzione della pianificazione in base alle materie prime e semilavorati presenti in magazzino. Per fare ciò ho seguito i passi presentati di seguito:

Controllo giacenze

1. lettura dal database dei dati riguardanti le giacenze;
2. utilizzo di una funzione di esplosione distinta per ottenere la distinta base$_G$ dei prodotti finiti;
3. scrittura dei dati in un file JSON;
4. creazione di una classe volta a lavorare sulle giacenze;
5. creazione di una funzione volta al controllo della presenza di materiali per eseguire la pianificazione;
6. integrazione di tale funzione con gli algoritmi Greedy e Tabu Search;
7. rimozione dei materiali necessari a produrre un prodotto finito;
8. reintegrazione dei materiali in magazzino se uno o più articoli vengono eliminati dalla pianificazione.

Pianificazione semilavorati

1. utilizzo di una funzione di esplosione distinta per ottenere la distinta base$_G$ dei prodotti finiti;
2. scrittura dei dati in un file JSON;
3. riutilizzo della classe *Ordine* per i semilavorati;
4. ampliamento delle funzionalità della classe *Ordine* per tenere traccia dei semilavorati di ogni prodotto finito;
5. integrazione della pianificazione dei semilavorati nell'algoritmo Greedy, anticipando la loro produzione rispetto al prodotto finito;
6. inserimento di controlli ausiliari per impedire spostamenti ed eliminazioni durante l'esecuzione dell'algoritmo di Tabu Search.

Di seguito è mostrato come si presenta la pianificazione restituita dall'applicativo dopo la pianificazione dei semilavorati.

Codice Articolo	Descrizione Articolo	UM	Giacenza al 21/10/2019	MARTEDÌ 22/10/2019 Cod. Linea Prod.	MARTEDÌ 22/10/2019 Linea Produzione	MARTEDÌ 22/10/2019 Disponibile	MARTEDÌ 22/10/2019 Pianificata	MARTEDÌ 22/10/2019 Pianificata da Ordine	MARTEDÌ 22/10/2019 Su Pianif. Oraria
F78	FARINA "00" PASTA ORO 10X1KG	KG	100,000			100,000	500,000		50,000
I42	CORIANDOLI CACAO X 1,5KG	KG	11.000,000			11.000,000			0,000
I43	PASTICCINI COCCO GLASSATI X 1,5KG	KG	0,000			0,000			0,000

Codice Articolo	Descrizione Articolo	UM	MERCOLEDÌ 23/10/2019 Disponibile	MERCOLEDÌ 23/10/2019 Pianificata	MERCOLEDÌ 23/10/2019 Pianificata da Ordine	MERCOLEDÌ 23/10/2019 Su Pianif. Oraria	MERCOLEDÌ 23/10/2019 Ordini Clienti	MERCOLEDÌ 23/10/2019 Giacenza
F78	FARINA "00" PASTA ORO 10X1KG	KG	600,000			0,000	0,000	600,000
I42	CORIANDOLI CACAO X 1,5KG	KG	11.000,000	400,000	400,000	400,000	0,000	11.400,000
I43	PASTICCINI COCCO GLASSATI X 1,5KG	KG	0,000	400,000	400,000	400,000	0,000	400,000

Figura 4.2: Pianificazione semilavorati

Dall'immagine precedente vediamo che *F78* è un semilavorato,di cui abbiamo 100kg in giacenza, richiesto dai prodotti finiti *I42* e *I43*. Sappiamo che i due prodotti finiti in questione richiedono 300kg del semilavorato *F78* ciascuno, e vediamo che il martedì ne sono pianificati esattamente 500kg per garantire la produzione di *I42* e *I43* il mercoledì.

Controllo ordini fornitori

1. lettura dal database i dati riguardanti gli ordini fornitori;
2. scrittura dei dati in un file JSON;
3. creazione di una classe volta a lavorare sugli ordini fornitori;
4. ampliamento della funzione dedita al controllo delle giacenze;
5. integrazione di tale funzione con gli algoritmi Greedy e Tabu Search;
6. inserimento di controlli ausiliari per evitare di ottenere soluzioni non ammissibili durante l'esecuzione di entrambi gli algoritmi.

Pianificazione ordini da magazzino

1. lettura di ordini da inserire in pianificazione direttamente dal magazzino;
2. scrittura dei dati in un file JSON;
3. riutilizzo della classe *Ordine* per gli ordini da magazzino;
4. utilizzo di tutti i vincoli precedenti per effettuare la pianificazione.

Criteri di Diversificazione/Intensificazione

In questa fase ho studiato come ottenere il massimo dall'esecuzione dell'algoritmo Tabu Search.
Si è infine scelto di utilizzare una tecnica di diversificazione come spiegato nella sezione precedente.

4.2 Verifica e validazione

La fase di verifica si è protratta durante tutta la fase di sviluppo del progetto. Ho scelto delle tecniche di analisi statica, in quanto non era previsto un piano di test da effettuare sui risultati ottenuti.
Il fatto di non eseguire dei test automatici sui risultati ottenuti deriva anche dal fatto che i dati utilizzati corrispondano a dei dati reali, per i quali sarebbe stato più complicato codificare un insieme di test automatici e avrebbe sottratto un considerevole ammontare di tempo allo sviluppo del codice applicativo.

4.2.1 Analisi statica

Ho fatto ampio uso dello strumento di debug fornito da Visual Studio 2010 e ho creato una lista di controllo tramite breakpoint$_G$ in modo da poter valutare, durante l'esecuzione, lo stato del software ad ogni punto critico. Ciò mi ha permesso di individuare efficacemente gli errori a livello logico che si sono presentati durante la varie fasi di sviluppo. Ho proseguito poi con un controllo effettivo sulla soluzione presentata a video dal programma, la quale veniva valutata a campione su determinati ordini specifici, in modo da ottenere un riscontro reale della bontà della soluzione stessa.

4.2.2 Analisi dinamica

Ho fatto uso di più funzioni con il compito di attestare che la soluzione corrente sia ammissibile, in particolare ogni funzione aveva il compito di verificare che determinati vincoli fossero rispettati. Ad esempio, dopo aver inserito il controllo sulle materie prime e semilavorati, verificavo che la rimozione di queste due componenti combaciasse con quanto risultava presente in magazzino. Ho effettuato altri controlli sul tempo di pianificazione rimanente e vincoli di linea imposti dalle relative aziende. Si sottolinea che queste funzioni vengono richiamate solo in punti ben definiti nel codice, in quanto non sarebbe utile richiamare la stessa funzione di controllo se i vincoli che attesta non sono stati modificati nella soluzione. Nel caso in cui una di queste funzioni accertasse l'inammissibilità della soluzione corrente veniva lanciata un eccezione, la quale riportava il vincolo non era stato soddisfatto.

4.2.3 Validazione

Al raggiungimento di ogni stato rilevante dell'applicativo eseguivo un collaudo. Questo aveva il compito di accertare che i risultati ottenuti fossero in linea con quanto ci si aspetta dalla pianificazione reale. Per fare ciò eseguivo l'applicativo con i dati reali di una specifica azienda, dei quali si era giunti ad ottenere una soluzione relativamente buona. Una volta confrontate le due soluzioni evidenziavo le eventuali discrepanze e passavo ad una verifica accurata di quanto ottenuto. Seguiva poi una discussione con il tutor interno per accertarsi che tali scostamenti derivassero dalle nuove aggiunte e non fossero errori logici del programma.
Una volta accertati che la soluzione ottenuta fosse ammissibile e coerente veniva salvata e impiegata nelle successive fasi di validazione e collaudo.

4.3 Risultati dei test

Di seguito vengono riportati i risultati sui test effettuati durante le varie fasi di sviluppo del progetto. Sono presentati in forma grafica così da garantire un immediata comprensione.
I dati utilizzati per ottenere questi risultati fanno parte degli ordini, delle giacenze di magazzino e degli ordini fornitori di uno specifico cliente di Ergon Informatica.
Ogni test è stato effettuato sullo stesso ammontare di ordini distribuito in una singola settimana lavorativa, questo per garantire una continuità tra i vari test eseguiti.
L'insieme dei dati utilizzati sono i seguenti:

- Numero ordini = 273;
- Numero semilavorati = 183;
- Numero giacenze = 525;
- Numero ordini fornitori = 57.

Si precisa che non tutti gli ordini devono essere pianificati e che vengono presi in considerazione solo i dati riguardanti la settimana corrente di pianificazione. Ogni grafico è presentato in due diverse versioni, una tramite grafico a barre per confrontare le diverse misurazioni e una tramite grafico a dispersione per evidenziare l'andamento della soluzione.

4.3.1 Pianificazione prodotti finiti

Qui di seguito viene rappresentato l'andamento temporale dell'esecuzione dell'algoritmo Greedy e Tabu Search al variare del numero di iterazione massime consentite.

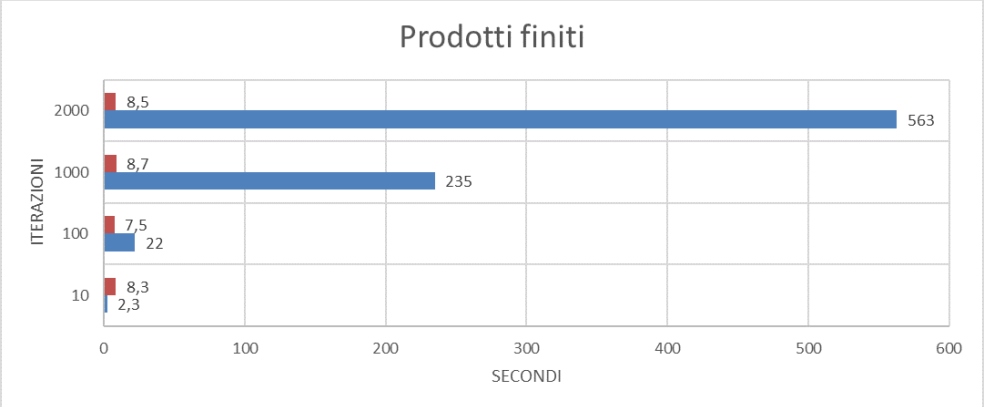


Figura 4.3: Grafico prodotti finiti

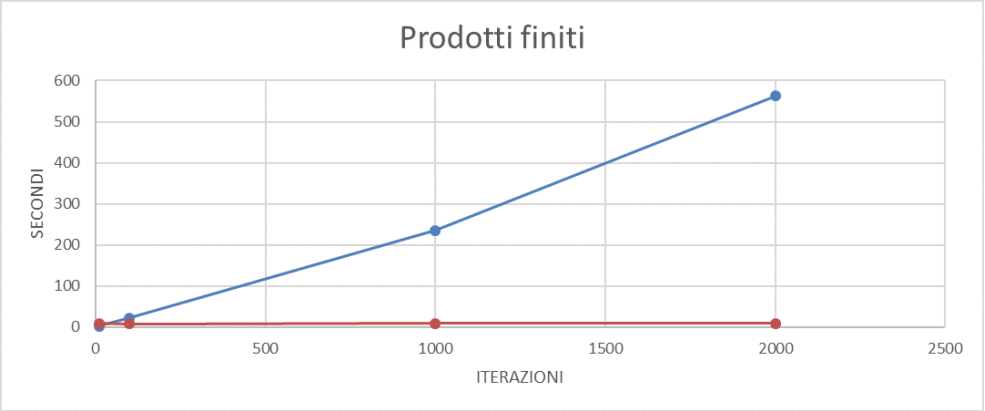


Figura 4.4: Grafico prodotti finiti

Come possiamo notare il tempo di esecuzione dell'algoritmo Greedy non varia in quanto non è dipendente dal numero di iterazioni ma solo dal numero di ordini.
Per quanto riguarda il tempo di esecuzione dell'algoritmo Tabu Search vediamo un aumento in proporzione al variare del numero di iterazioni. Di seguito viene mostrato il miglioramento in percentuale ottenuto dall'algoritmo di ottimizzazione.

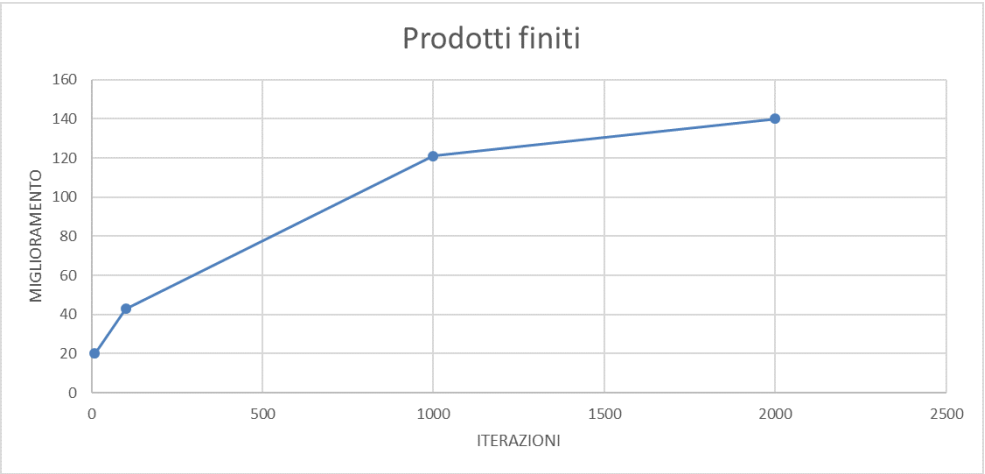


Figura 4.5: Grafico miglioramento prodotti finiti

Dopo vari test eseguiti posso affermare che, considerando anche il tempo impiegato, il numero di iterazioni ideale si assesti tra le 1000 e 2000 iterazioni. Superate le 2000 iterazioni il miglioramento assume un ordine asintotico ed evidenzia solo uno spreco di tempo di esecuzione.

4.3.2 Pianificazione semilavorati

Di seguito sono rappresentati i risultati dei test eseguiti dopo il primo importante avanzamento del progetto ovvero la pianificazione dei semilavorati.

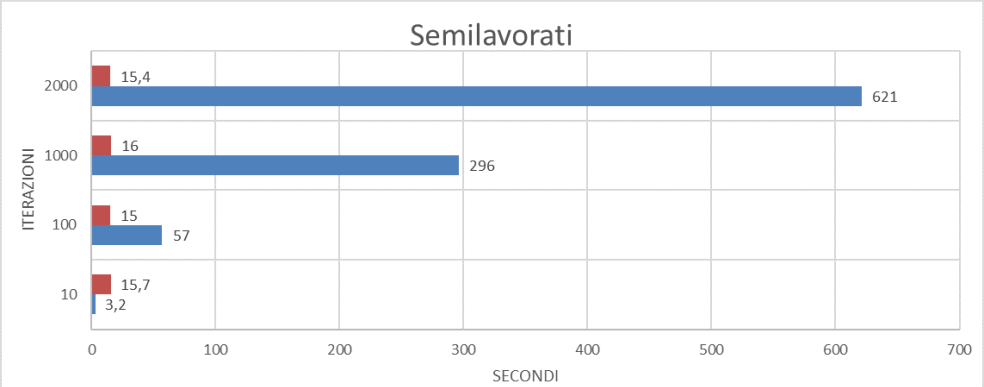


Figura 4.6: Grafico semilavorati

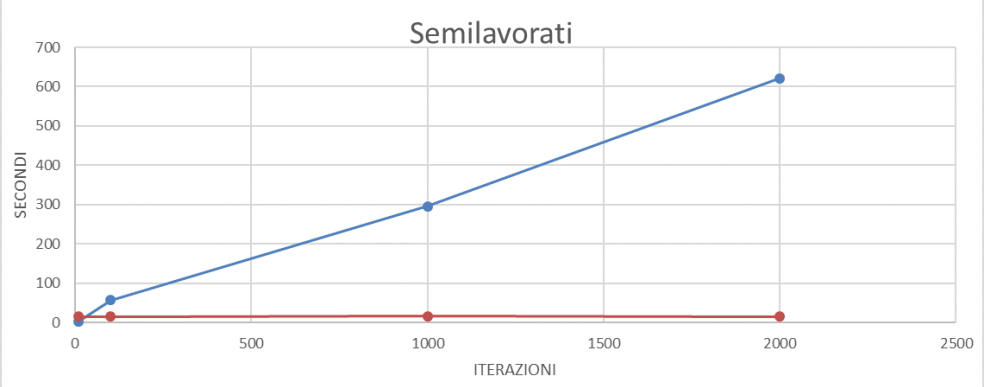


Figura 4.7: Grafico semilavorati

La prima cosa da notare è l'aumento del tempo di esecuzione da parte dell'algoritmo Greedy, il quale necessita ora di un elevato numero di controlli aggiuntivi. La funzione stessa è stata resa ricorsiva$_G$ in modo da riutilizzare il codice al suo interno per eseguire la pianificazione dei semilavorati. Questo ha permesso anche di creare una relazione diretta tra il semilavorato e il suo prodotto finito, il tutto tramite una relazione padre figlio creata appositamente nella classe. In compenso l'algoritmo di Tabu Search non ha subito un aumento considerevole del tempo di esecuzione ma ha subito un calo drastico nella capacità di ottimizzazione della soluzione.

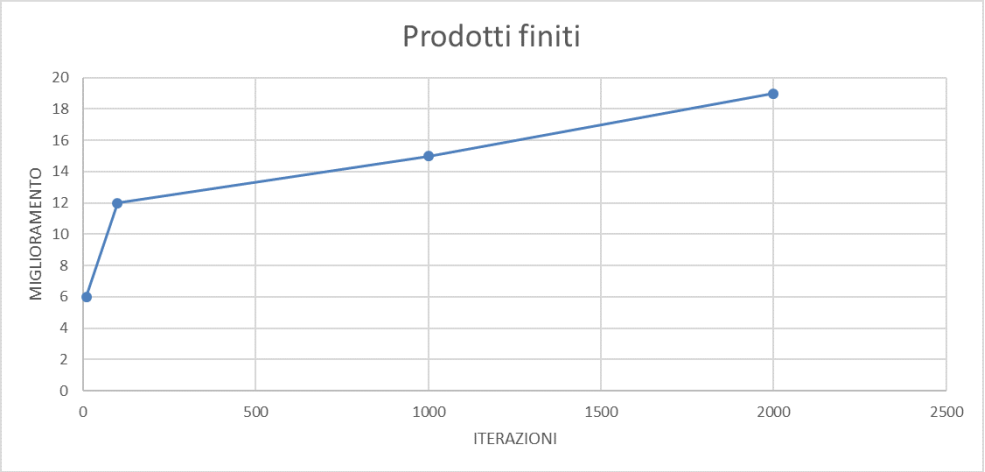


Figura 4.8: Grafico miglioramento semilavorati

Come possiamo notare ora, al termine delle 2000 iterazioni, il fattore di miglioramento percentuale si aggira attorno 19 percentile. Questo è dovuto al fatto che l'algoritmo si trova difronte a un elevato numero di controlli ausiliari prima di poter eseguire una mossa. Si prenda ad esempio la mossa di eliminazione, non può essere effettuata su un semilavorato per questioni logico/reali in quanto non sarebbe più possibile pianificare il suo prodotto finito relativo. Allo stesso modo non è possibile spostare la pianificazione di un semilavorato successivamente al suo prodotto finito e viceversa non è possibile pianificare il prodotto finito prima dei suoi semilavorati. Ciò è la causa della perdita di capacità di ottimizzare la soluzione ottenuta dall'algoritmo Greedy.

4.3.3 Controllo ordini fornitori

In aggiunta alla pianificazione dei semilavorati è stato aggiunto il controllo sull'arrivo di materiale da parte dei fornitori.
Ciò permette di non scartare a priori la possibilità di pianificare alcuni ordini che, a causa dell'assenza di materiali,non verrebbero considerati. In questo modo vengono considerati pianificabili tutti gli ordini per i quali è presente almeno un ordine fornitore che soddisfa la quantità di materiali necessari.

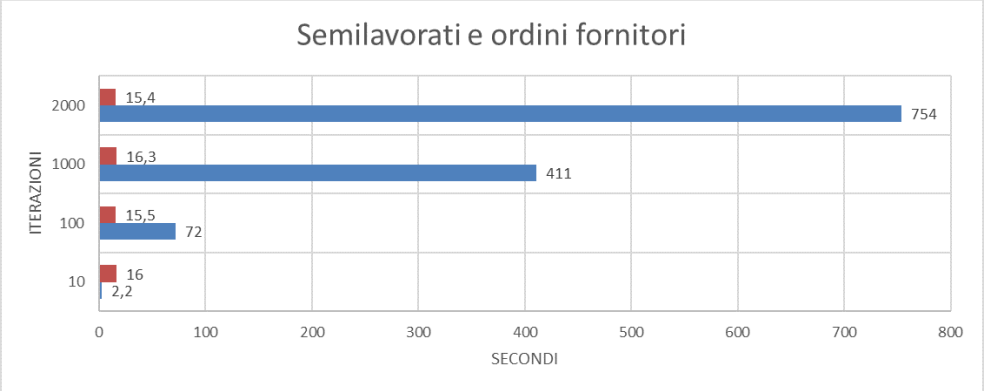


Figura 4.9: Grafico ordini fornitori

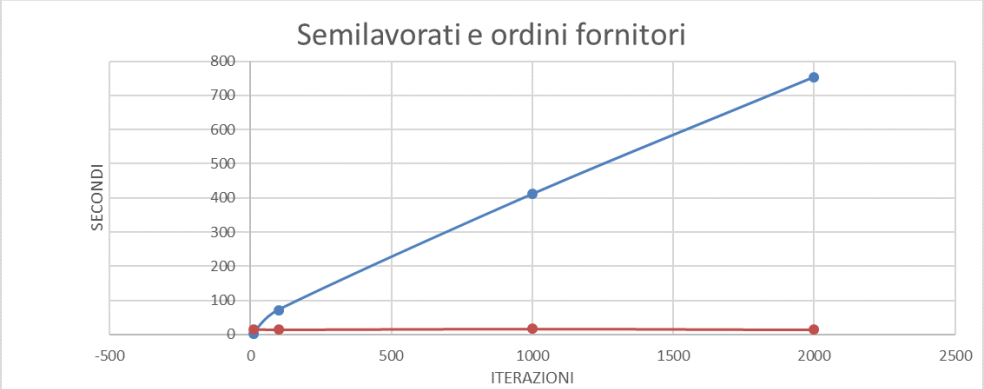


Figura 4.10: Grafico ordini fornitori

Come si può notare il tempo di esecuzione dell'algoritmo Greedy non è aumentato in quanto i controlli di questo tipo sono molto più veloci rispetto alla fase precedente. Per quanto riguarda la Tabu Search notiamo anche in questo caso un aumento di tempo al variare del numero di iterazioni, ma rimanendo comunque entro limiti accettabili.

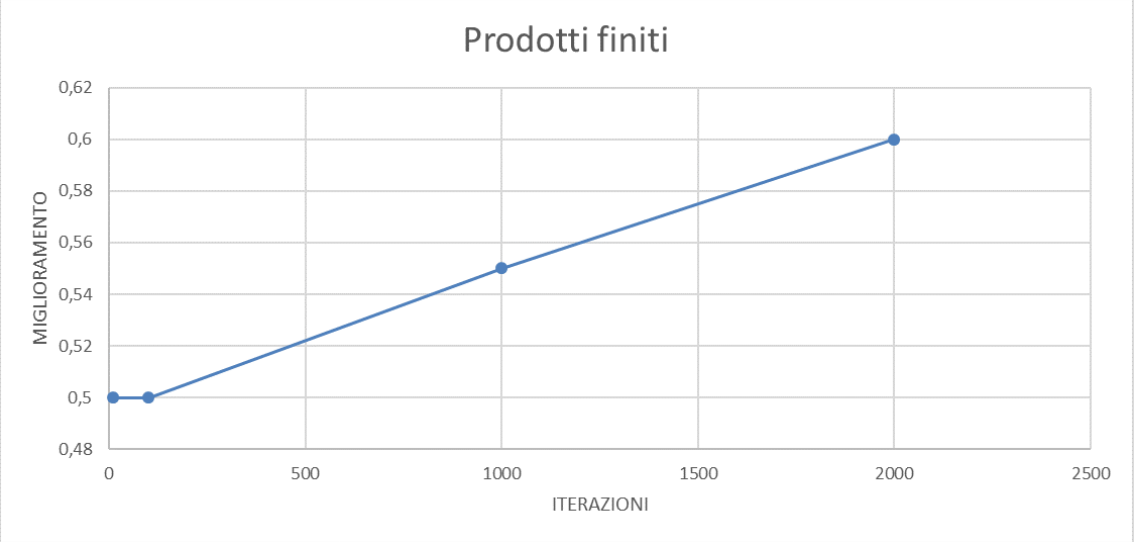


Figura 4.11: Grafico miglioramento ordini fornitori

Da questo grafico si evince la netta incapacità della Tabu Search di ottenere una soluzione migliore partendo dalla soluzione fornita dall'algoritmo Greedy.
Questo avviene perché, in questo caso particolare, molti ordini sarebbero pianificabili per quanto riguarda i materiali a disposizione (cosa non realizzabile per le fasi precedenti) e questo va a diminuire nettamente il risultato del rapporto

$$\frac{\text{Pezzi pianificati}}{\text{Totale pezzi richiesti}} * \text{Peso del rapporto pezzi}$$

dove in *Totale pezzi richiesti* vengono considerati tutti gli ordini producibili.
Inoltre avendo un grandissimo numero di ordini producibili le mosse eseguite dalla Tabu Search non fanno altro che sostituire ordini con altri ordini quindi non può ottenere un miglioramento significativo.

4.3.4 Criteri di diversificazione

Mi è stato richiesto inoltre di applicare dei criteri di diversificazione per quanto riguarda la Tabu Search.
Dopo l'implementazione di tale tecnica ho eseguito i test su dei dati specifici, in quanto non avrei ottenuto grandi risultati basandomi sulle ultime aggiunte del progetto.

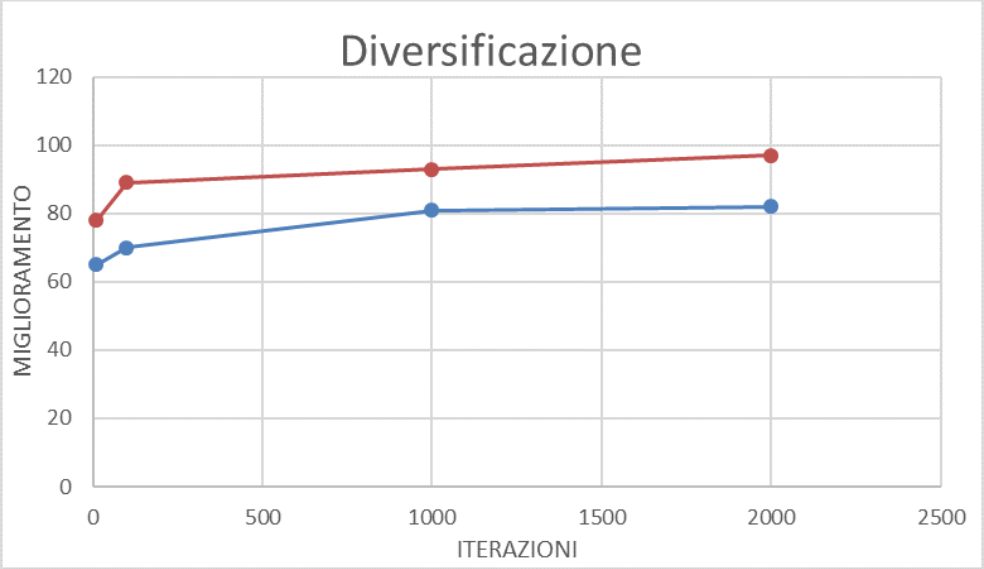


Figura 4.12: Grafico diversificazione

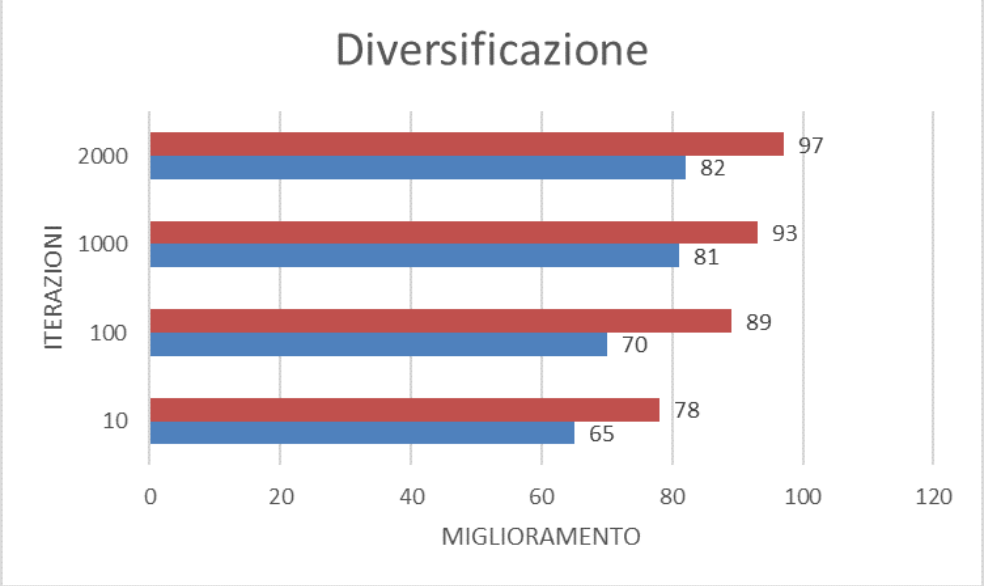


Figura 4.13: Grafico diversificazione

Come possiamo notare dai grafici sopra, vi è un aumento del 20% circa, in ogni stato di esecuzione.
Questo evidenzia l'efficacia dei criteri di diversificazione utilizzati. Inoltre, esaminando le soluzione fornite pre e post inserimento dei criteri, si può notare che questi ultimi prediligono la pianificazione di ordini aventi un maggior numero di pezzi, tralasciando molti ordini aventi un quantitativo di pezzi inferiore alla media.
Tali dati verranno presi in considerazione da Ergon Spa per decidere, in base alle esigenze dei clienti, quale delle due politiche di pianificazione adottare.

Capitolo 5

Conclusioni

5.1 Consuntivo delle tempistiche

Questo capitolo racchiude l'intero andamento del progetto di stage, passando per tutte le sue fasi.
La pianificazione oraria ha subito alcuni cambiamenti durante la durata dello stage. Questo perché mi è stato chiesto di dare maggiore priorità ad alcuni elementi ritenuti più importanti per l'azienda rispetto ad altri aspetti secondari che potevano essere colmati in altri modi. In particolare la parte di stesura di manuale utente e sviluppatore, considerata opzionale, è stata tralasciata per dare maggior peso alla parte di test e validazione del prodotto. Inoltre si è scelto, assieme al tutor interno, di utilizzare come appoggio i commenti a supporto del codice piuttosto che i manuali sopracitati. Anche la fase di stesura dell'analisi dei requisiti ha subito una variazione oraria, in quanto l'aggiunta dei casi d'uso da me proposti ha ricevuto riscontri positivi ed è proseguita senza intoppi fin da subito.
Un lieve aumento lo si nota sulla fase di studio delle tecnologie utilizzate, in quanto dovevo familiarizzare con strumenti che non avevo mai utilizzato, quali DevExpress e vb.NET, la loro padronanza ha richiesto più del dovuto.
Infine, come anticipato, la fase di test e validazione ha subito un aumento in quanto, l'accertamento della bontà della soluzione richiedeva un considerevole ammontare di tempo sia da parte mia che da parte del tutor interno. Questo tempo ha inoltre incluso la fase di controllo statica del codice durante la stesura delle nuove unità e controlli dinamici durante l'esecuzione dell'applicativo.

Tabella 5.1: Ore effettive

Ore preventivate	Descrizione	Ore effettive
40	Analisi del modulo software esistente, funzionalità da realizzare e documentazione disponibile dell'algoritmo esistente	40
12	Analisi dei requisiti e stesura della relativa documentazione	4(-8)
16	Studio delle tecnologie aziendali necessarie allo sviluppo del modulo	20(+4)
40	Studio di algoritmi e tecniche di Ricerca Operativa e Ottimizzazione Combinatoria	40
128	Sviluppo delle componenti: • Sviluppo procedura di gestione degli ordini fornitori • Sviluppo procedura di gestione delle giacenze di magazzino • Sviluppo procedura di gestione dei semilavorati pianificati • Integrazione dei vincoli del problema di ottimizzazione con i nuovi parametri • Algoritmo Greedy: sviluppo di nuove strategie di scelta del passo successivo adottato dall'algoritmo nella costruzione della soluzione del problema • Integrazione della Tabu Search con nuovi meccanismi: criteri di aspirazione e arresto, variazione dei meccanismi di esplorazione del vicinato e adozione di tecniche di intensificazione e diversificazione	132(+4)
60	Validazione e test	80(+20)
24	Stesura della documentazione del prodotto sviluppato	4(-20)

5.2 Consuntivo dell'analisi dei rischi

Si sono verificati due dei tre casi preventivati nella fase di analisi dei rischi.
Il primo riguarda il rischio *RT1-Inesperienza tecnologica* con il quale mi sono scontrato nelle prime fasi del progetto. Il secondo riguarda *RT2-Scelte implementative* il quale ha richiesto maggior tempo di verifica durante la codifica.

Tabella 5.2: Consuntivo dell'analisi dei rischi

Descrizione	Soluzione
Le nuove tecnologie da utilizzare si sono verificate più complicate del previsto, causato anche dal fatto di dover ampliare un progetto già esistente con diverse funzionalità avanzate già implementate	Ho fatto uso della documentazione a supporto dell'applicativo e dell'aiuto del tutor interno che mi dava indicazioni ad ogni passo di esecuzione dell'algoritmo
Le varie scelte implementative spesso conducevano a strade senza uscita, dalle quali era necessario ricominciare con una nuova progettazione del codice	Tramite l'aiuto del tutor interno e delle conoscenze ormai acquisite sono sempre riuscito a trovare tecniche alternative per portare al termine il progetto

5.3 Soddisfacimento dei requisiti e raggiungimento degli obiettivi

Tutti i requisiti obbligatori sono stati soddisfatti, i quali formavano le fondamenta del progetto di stage ed erano considerati di fondamentale importanza per l'azienda. Si è raggiunta la totalità dei requisiti desiderabili, i quali davano lustro alle funzionalità dell'applicativo e ne accrescevano l'insieme di funzionalità. I due requisiti facoltativi non soddisfatti riguardano la stesura dei manuali, convertita in aggiunta di commenti durante la codifica.

Tabella 5.3: Requisiti soddisfatti

Stato	Obbligatori	Desiderabili	Facoltativi
Soddisfatti	24	5	0
Non soddisfatti	0	0	2

Per quanto riguarda gli obiettivi opzionali ne è stato scartato uno in seguito ad un colloquio con il tutor aziendale, secondo il quale non valeva la pena di sovraccaricare il programma con dei dati ridondati per eseguire una ripianificazione. Si è scelto quindi di pianificare partendo da zero con i nuovi dati. Di seguito sono riportati in grafico i requisiti e il loro Soddisfacimento.
Gli altri obiettivi sono stati ampiamente raggiunti durante le rispettive fasi dello sviluppo dell'applicativo.

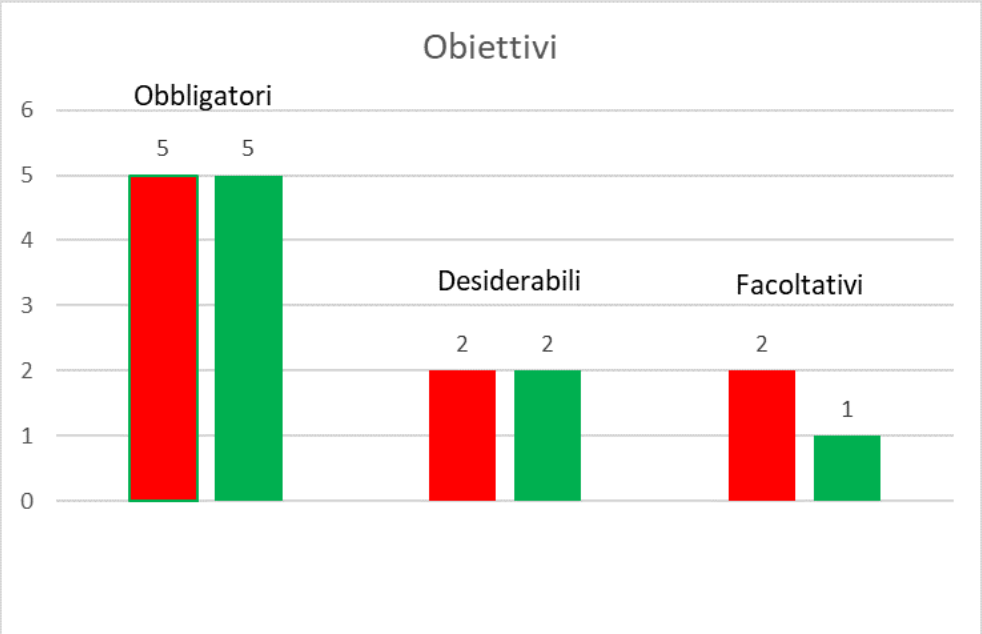


Figura 5.1: Obiettivi raggiunti

5.4 Conoscenze acquisite

Durante il percorso di stage ho acquisito nuove conoscenze sia in ambito tecnico che di sviluppo sottostando ad esigenze reali.

5.4.1 Vb.NET, DevExpress e Informix

Sono le tecnologie che più ho utilizzato durante lo svolgimento dello stage, e mi sono servite durante tutto l'arco di codifica e testing.
Essendo la prima volta che mi interfacciavo con queste tecnologie ho riscontrato alcuni problemi in partenza, dovuti appunto all'inesperienza e al fatto che il mio progetto si integrava ad uno esistente. La base teorica comunque è riconducibile a qualsiasi altro linguaggio di programmazione visto anche in ambito universitario per quanto riguarda Vb.NET.
Per quanto riguarda Informix si sono rese sufficienti le conoscenze teoriche e pratiche conseguite durante il percorso di studi, si è resa necessario qualche intervento da parte del tutor aziendale in casi particolari di formattazione delle tabelle che andavo ad utilizzare.
DevExpress ha avuto un ruolo marginale rispetto ai precedenti ma è comunque servito ad interpretare correttamente le funzionalità dell'interfaccia grafica sulla quale dovevo apportare le modifiche richieste. Al termine dello stage posso affermare che ho acquisito la piena padronanza di queste tecnologie non solo a livello superficiale ma anche andando più in profondità, altre conoscenze derivano dal fatto di prolungare il loro utilizzo in futuro.

5.4.2 Metodo di lavoro

È la prima volta che mi pongo dinanzi ad un progetto di tali dimensioni in solitaria, ciò mi ha permesso di imparare a gestire le varie scadenze e consegne entro dei limiti imposti da un vero e proprio mondo del lavoro.
Inizialmente ho leggermente sottovalutato le ore di studio del problema esistente, non per negligenza ma perché non mi è parso chiaro sin da subito l'entità del problema che dovevo affrontare, fino a quando non ho effettivamente iniziato a lavorarci.
Ciò ha inizialmente rallentato la fase di sviluppo in quanto dovevo porre frequenti domande al tutor anche per aiutarmi ad entrare nell'ottica della pianificazione eseguita sui generi alimentari di cui tratta l'azienda.
Ciò mi ha concesso di capire quanto sia importante chiarire ogni dubbio prima di iniziare a lavorare sulla soluzione del problema, così da non causare un effetto cascata in seguito. Nelle fasi successive, grazie a tale esperienza, non ho avuto problemi al di fuori di questioni legate a vincoli reali di pianificazione ed è stato possibile proseguire senza rallentamenti.

5.4.3 Scadenze

Anche in questo caso ho dovuto riadattare il mio regime di lavoro in base alle scadenze imposte.
Avendo già avuto a che fare con problemi di scheduling$_G$ del lavoro, mi sono affidato ad un calendario che ho compilato dal primo giorno di stage, nel quale inserivo i vari obiettivi da raggiungere settimanalmente e giornalmente. Con questo sono riuscito a garantire un ottimo flusso di esecuzione delle operazioni da portare al termine in modo

lineare ed efficacie. Nel caso di problematiche esterne ho valutato dei possibili tempi di stallo nel planning$_G$ così da avere un margine di scarto in ogni settimana lavorativa.

5.5 Università e mondo del lavoro

Ho sempre avuto la tendenza a valutare il mondo scolastico completamente separato dal mondo del lavoro. Questo perché alcune conoscenze tendono a fermarsi al livello teorico senza però scendere nel dettaglio e quindi mancanti della parte pratica. Al contrario nel mondo del lavoro ho sempre pensato che sia la pratica a far da padrone e che quindi risultasse sì utile il grado di preparazione teorica, ma che non avesse la stessa rilevanza in termini di capacità richieste ad un qualsiasi lavoratore. Dopo questo stage posso invece affermare che il grado di conoscenza teorico è di gran lunga superiore al grado di conoscenza pratico ed è un elemento fondamentale che un qualsiasi buon programmatore deve avere.
Durante il progetto ho dovuto spaziare tra diversi campi che vanno dall'Ingegneria del Software alla Ricerca Operativa, passando per altre decine di conoscenze che se non avessi frequentato l'Università non avrei padroneggiato. Inoltre ho valutato importantissima la capacità di saper imparare nuovi concetti e acquisire informazioni, grazie al metodo di studio affinato durante il percorso di studi, la quale supera di gran lunga la capacità di saper semplicemente utilizzare un qualsiasi strumento di sviluppo o simili.
In sostanza sono fermamente convinto che anche un percorso di studi triennale fornisca le competenze necessarie e sufficienti per approcciarsi in modo corretto al mondo del lavoro.

5.5.1 Valutazione personale

L'intero progetto di stage è stato svolto sotto la supervisione del tutor aziendale che però non ha partecipato attivamente allo sviluppo del progetto, anzi ha cercato di lasciarmi quanto più spazio possibile cosicché possa trovare da solo una soluzione ai problemi che, molto probabilmente, dovrò riaffrontare in futuro. Ovviamente era sempre disponibile in caso di problematiche che andassero al dì la della mia comprensione in materia di politiche aziendali e anche nel caso in cui dovessi affrontare dei problemi implementativi di spessore.
Il fatto di doversi gestire con le tempistiche e le varie scadenze ha accresciuto molto la mia competenza di giudizio sulle mie reali capacità. Inoltre il rapporto coi colleghi mi ha aiutato ad entrare, soprattutto mentalmente, nell'ideologia lavorativa condivisa in azienda e anche questo mi ha permesso di crescere a livello personale.
In conclusione ritengo il periodo di stage un fattore di crescita personale di fondamentale importanza per uno studente universitario. Ti permette di capire se il tuo percorso di studi è stato utile e se effettivamente è il lavoro che intendi fare una volta concluso il proprio percorso.
Lo ritengo fondamentale in un corso di studi come il nostro e sono d'accordo che sia obbligatorio per conseguire il titolo di studio.

Bibliografia