



**Politecnico
di Torino**

POLITECNICO DI TORINO

Corso di Laurea Triennale in Ingegneria Gestionale
Classe L8 – Ingegneria dell'informazione

Software per la gestione ed ottimizzazione delle prenotazioni delle lezioni: Il caso di un'associazione per DSA

Relatore:

Prof. Fabrizio Lamberti

Candidato:

Gajoni Manuel

Anno Accademico: 2023/2024

Sommario

1. PROPOSTA DI PROGETTO.....	1
1.1 DESCRIZIONE DEL PROBLEMA PROPOSTO	1
1.2 DESCRIZIONE DELLA RILEVANZA GESTIONALE DEL PROBLEMA	1
1.3 DESCRIZIONE DEI DATA-SET PER LA VALUTAZIONE	2
1.4 DESCRIZIONE PRELIMINARE DEGLI ALGORITMI COINVOLTI	2
1.5 DESCRIZIONE PRELIMINARE DELLE FUNZIONALITÀ DELL'APPLICAZIONE SOFTWARE	3
2. DESCRIZIONE DEL PROBLEMA.....	4
2.1 INTRODUZIONE ASSOCIAZIONE	4
2.2 CASO IN ESAME.....	4
2.3: INPUT, OUTPUT E CRITICITÀ.....	10
2.4: VINCOLI DA RISPETTARE	12
3. DESCRIZIONE DEL DATA-SET	13
3.1 NECESSITÀ DI INPUT	13
3.2 DESCRIZIONE DATA-SET	13
3.3 DIAGRAMMA ER	16
4. STRUTTURE DATI E ALGORITMI UTILIZZATI.....	17
4.1 STRUTTURE DATI	17
4.2 ALGORITMI PRINCIPALI.....	17
4.2.1 LezioniDAO & ConnectDB	18
4.2.2 FXMLController.....	23
4.3 ALGORITMO RICORSIVO.....	29
4.3.1 Calcolo punteggio lezione.....	29
4.3.2 Calcola lezioni giorno.....	31
5. OUTPUT GRAFICO.....	38
5.1 ELEMENTI COMUNI	38
5.2 DESCRIZIONE INTERFACCIA GRAFICA.....	40
5.2.1 Gestisci Lezioni	40
5.2.2 Gestisci Studenti.....	43
5.2.3 Gestisci Tutor	44
5.2.4 Resoconto	45
5.3 OUTPUT GRAFICO	46
6. RISULTATI OTTENUTI E CONSIDERAZIONI FINALI	49
6.1 RACCOLTA DATI	49
6.2 DATI OTTENUTI.....	50
6.2.1 Lezioni non prenotabili.....	50
6.2.2 Media punteggio mensile.....	52
6.2.3 Numero di educatori interni.....	53
6.3 CONSIDERAZIONI FINALI.....	54
INDICE DELLE FIGURE	55

1. PROPOSTA DI PROGETTO

1.1 Descrizione del problema proposto

Lavorando in un'associazione che fornisce tra i vari servizi quello di tutoraggio e ripetizioni per gli studenti, dalle scuole Elementari all'Università, ho riscontrato come la gestione delle prenotazioni possa essere problematica, soprattutto se non gestita correttamente. È presente una richiesta altamente variabile da parte degli studenti che dipende dal periodo scolastico in cui ci si trova. I tutor, a loro volta, garantiscono delle disponibilità che possono variare da settimana a settimana e non garantiscono di poter seguire tutte le materie scolastiche. Ad oggi le prenotazioni sono gestite dal solo utilizzo di un foglio di calcolo di Excel, questo metodo presenta alcune problematiche date dalla difficoltà nel reperire informazioni su studenti, tutor e sulle lezioni svolte in passato. La mancanza di programmazione dovuta dalla compilazione del calendario, che viene fatta giornalmente, comporta un aumento del rischio di imprevisti e del tempo richiesto.

1.2 Descrizione della rilevanza gestionale del problema

Per questo problema è fondamentale la componente gestionale perché per poter migliorare l'efficienza delle prenotazioni è importante l'organizzazione nella fase iniziale dell'assegnazione della lezione al tutor. Il 'collo di bottiglia' del processo sono, appunto, i tutor per via della loro disponibilità variabile e l'impossibilità di seguire tutte le materie scolastiche quindi, la loro gestione diventa fondamentale per poter aumentare il numero di lezioni annue. Un altro problema che può essere risolto è la tempestività nel risolvere eventuali problemi di abbinamento tutor-studente. Lo scopo del progetto è quello di confrontare in 5/6 mesi i due metodi di prenotazione: quello utilizzato fino ad oggi dall'associazione e in modalità 'automatica' con 'CALCOLA LEZIONI GIORNO' con lo scopo di capire le reali funzionalità e limiti del metodo automatico.

1.3 Descrizione dei Data-Set per la valutazione

I Data-Set utilizzati saranno creati da zero grazie ai dati reali che mi verranno forniti dall'associazione. Il campione preso in esame comprenderà l'anno scolastico 2023/24 nel quale sarà preso un campione di circa 200/250 studenti e 15/20 tutor e tutte le lezioni prenotate durante l'anno solare. Il Data-Set sarà strutturato in modo da avere cinque tabelle: una in cui saranno memorizzati gli studenti registrati per l'anno accademico, una in cui saranno memorizzati i tutor reperibili, una per le lezioni prenotate, una per le disponibilità dei tutor ed infine una con l'elenco di tutte le materie che possono essere selezionate per prenotare una lezione.

1.4 Descrizione preliminare degli algoritmi coinvolti

Nell'applicazione saranno presenti quattro schede: Gestisci lezioni, Gestisci studenti, Gestisci tutor e Resoconto. La sezione più rilevante a livello di algoritmi è quella 'Gestisci lezione' nella quale ci saranno tre sezioni: 'Lezioni singole', 'Lezioni multiple' e 'Lezioni giornaliere'. Nella sezione 'Lezioni multiple' sarà possibile prenotare e gestire più lezioni contemporaneamente grazie a due algoritmi:

- **CALCOLA LEZIONI:** Se uno studente dovrà prenotare più di una lezione, sarà possibile svolgerlo in maniera automatica tramite questo pulsante. In un primo momento verranno aggiunte tutte le lezioni richieste, senza selezionare il tutor a cui affidare la lezione. In un secondo momento il programma calcolerà i migliori abbinamenti possibili con i tutor assegnando un punteggio, che è determinato grazie all'utilizzo di una tabella precedentemente stabilita in collaborazione con l'associazione. Lo scopo dell'algoritmo sarà quello di massimizzare il punteggio di ogni assegnazione per poter garantire il miglior servizio possibile. La preparazione del tutor non è il solo criterio che verrà analizzato perché, a parità di punteggio finale, sarà preferita la soluzione che permetta di impiegare il maggior numero di tutor già a contratto piuttosto che i liberi professionisti, che collaborano con l'associazione e che possono essere assunti 'a chiamata'. L'algoritmo avrà lo scopo di trovare il percorso di assegnazione Studenti-Tutor che garantisca il maggior punteggio possibile tenendo però conto dei vincoli sulla preparazione dei tutor (nessun accoppiamento sarà accettato se il suo punteggio sarà

inferiore a 1), sul numero di tutor già a contratto (massimizzare questo numero per aumentare i ricavi) e sulla presenza di un certificato come educatore.

- **CALCOLA LEZIONI GIORNO:** Questo programma è pensato per essere utilizzato per calcolare il calendario delle lezioni per una data selezionata oppure nei casi di ‘emergenza’ nei quali uno studente o un tutor chiedono un cambiamento e c’è la necessità di riorganizzare tutte le lezioni della giornata. Il principio di funzionamento è il medesimo di ‘CALCOLA LEZIONE’, una funzione ricorsiva dove il peso è calcolato con lo stesso metodo visto in precedenza. In questo caso per gli n studenti che hanno prenotato la lezione al giorno selezionato cerco quale degli m tutor che ha dato disponibilità fornisca, nel complessivo, la soluzione migliore. I vincoli e i punteggi utilizzati all’interno dell’algoritmo rimangono invariati.

1.5 Descrizione preliminare delle funzionalità dell’applicazione Software

L’applicazione è pensata per essere utilizzata dalla persona preposta alle prenotazioni delle lezioni e dal tesoriere dell’associazione per compilare il resoconto mensile riguardante i dipendenti. Nella sezione ‘Gestisci Lezioni’ sarà possibile prenotare, modificare o cancellare lezioni singole o multiple. Per la prenotazione di lezioni multiple sarà possibile scegliere se effettuarla in modalità ‘Manuale’, nella quale sarà l’operatore a scegliere gli abbinamenti migliori o in modalità ‘Automatica’ dove sarà il Software a calcolare gli abbinamenti Tutor-Studente che totalizzino il punteggio maggiore. Le sezioni ‘Gestisci studenti’ e ‘Gestisci Tutor’ sono dedicate all’inserimento e alla gestione dei dati degli studenti e tutor nel database. La sezione ‘Resoconto’ è dedicata alla stampa delle lezioni ed ore dei tutor (o studenti) nel mese selezionato.

2. DESCRIZIONE DEL PROBLEMA

2.1 Introduzione associazione

L'idea per lo sviluppo del Software nasce dalla mia esperienza lavorativa pregressa in un'Associazione del terzo settore che opera sul territorio di Torino, Grugliasco, Collegno e Rivoli. L'Associazione presa in esame supporta bambini, ragazzi e adulti nell'apprendimento per la costruzione di percorsi di successo in autonomia e percorsi educativi specifici per persone con disabilità. Sono specializzati nel sostegno rivolto a chi ha DSA o BES, disturbi specifici dell'apprendimento e bisogni educativi speciali: elaborano insieme agli studenti gli strumenti compensativi più efficaci per facilitarne l'apprendimento. Nello specifico offrono anche servizi di tutoraggio per ragazzi dalle scuole Elementari fino all'Università.

2.2 Caso in esame

Il servizio che ho preso in esame è quello dei tutoraggi per studenti, durante l'anno accademico 2023/24. Questo servizio è attivo dal lunedì al sabato per tutto l'anno tranne una settimana di chiusura che normalmente è individuata nella seconda settimana di agosto. Gli orari in cui è possibile prenotare una lezione sono '15-17' e '17-19' durante i giorni settimanali, mentre il sabato gli orari sono '9-11', '11-13' e '15-17', questi orari sono in vigore solo durante l'anno scolastico, mentre, durante il periodo estivo (giugno, luglio e agosto) sono disponibili tutti gli orari tutti i giorni dal lunedì al sabato. La suddivisione delle lezioni prenotate nel periodo in esame è la seguente:

	LUNEDI'	MARTEDI'	MERCOLEDI'	GIOVEDI'	VENERDI'	SABATO	%
9-11	4	9	8	16	16	286	11
11-13	13	20	23	28	27	287	13
15-17	238	244	239	248	226	19	39
17-19	239	243	237	275	184	0	37
TOTALE	494	516	507	567	453	592	

Tabella 1: Divisione numero di lezioni per giorno ed orario

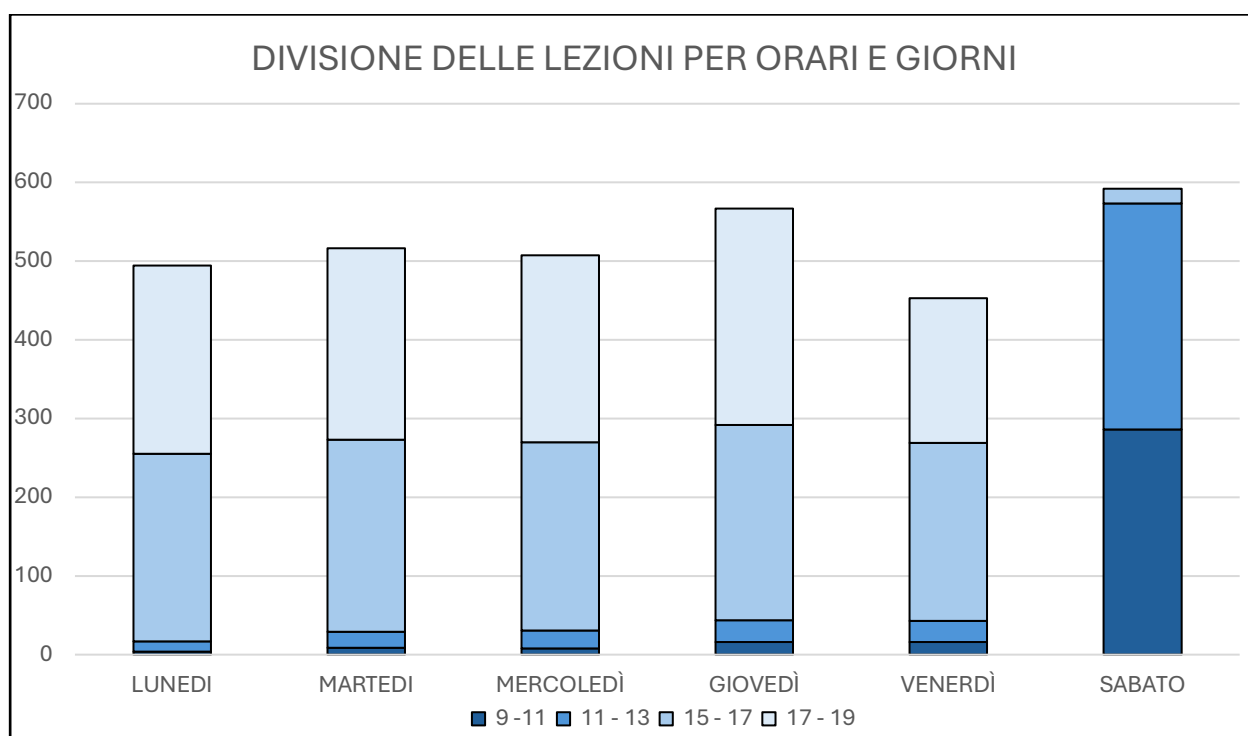


Tabella 2: Grafico per divisione delle lezioni per orari e giorni

Il servizio è garantito grazie all'impiego di tutor che svolgono le lezioni di sostegno scolastico a studenti singoli o in coppia. I tutor possono essere interni o esterni, i primi sono per la maggior parte educatori reperibili ad orari e giorni già prestabiliti ma sono anche coloro che si occupano delle altre attività dell'Associazione, esterne a quelle di tutoraggio, il che non garantisce la loro presenza durante tutti gli orari delle lezioni. In caso gli educatori a contratto non riuscissero a far fronte al numero di richieste delle lezioni si passerebbe a contattare tutor esterni, liberi professionisti che forniscono delle disponibilità ad inizio mese e che garantiscono di poter seguire determinate materie per il lavoro di tutoraggio con lo studente. Negli ultimi anni le richieste sono aumentate, quindi è normale che venga richiesto maggiormente il lavoro di tutor non a contratto, ad oggi quelli interni rappresentano circa il 30% della totalità dei tutor che abbiano lavorato nell'ultimo anno nell'associazione.

EDUCATORI	6	INTERNI	8
NON EDUCATORI	19	ESTERNI	17

Tabella 3: Suddivisione Tutor

TOTALI	3129	
INTERNI	1641	52%
ESTERNI	1488	48%

Tabella 4: Suddivisione Lezioni-Tutor

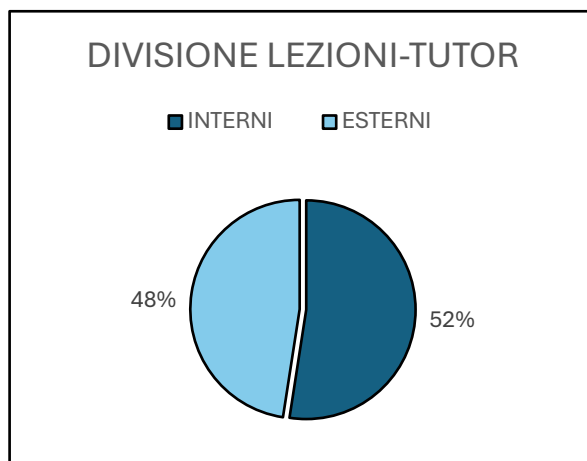


Tabella 5: Divisione in percentuale Lezioni-Tutor

Facendo riferimento all'anno accademico 2023/24 la richiesta è stata altamente variabile, concentrando la maggior parte delle prenotazioni nei mesi finali della scuola (marzo, aprile, maggio), questo fa sì che sia difficile prevedere con efficacia la domanda e, di conseguenza, assumere personale. La distribuzione delle lezioni è stata la seguente:

	SETTEMBRE	OTTOBRE	NOVEMBRE	DICEMBRE
NUMERO LEZIONI	131	264	344	229
%	4 %	8 %	11 %	7 %

Tabella 6: Divisione delle lezioni per mensilità (2023)

GENNAIO	FEBBRAIO	MARZO	APRILE	MAGGIO	GIUGNO	LUGLIO	AGOSTO
295	317	370	388	487	127	153	24
9 %	10 %	12 %	12 %	16 %	4 %	5 %	1 %

Tabella 7: Divisione delle lezioni per mensilità (2024)

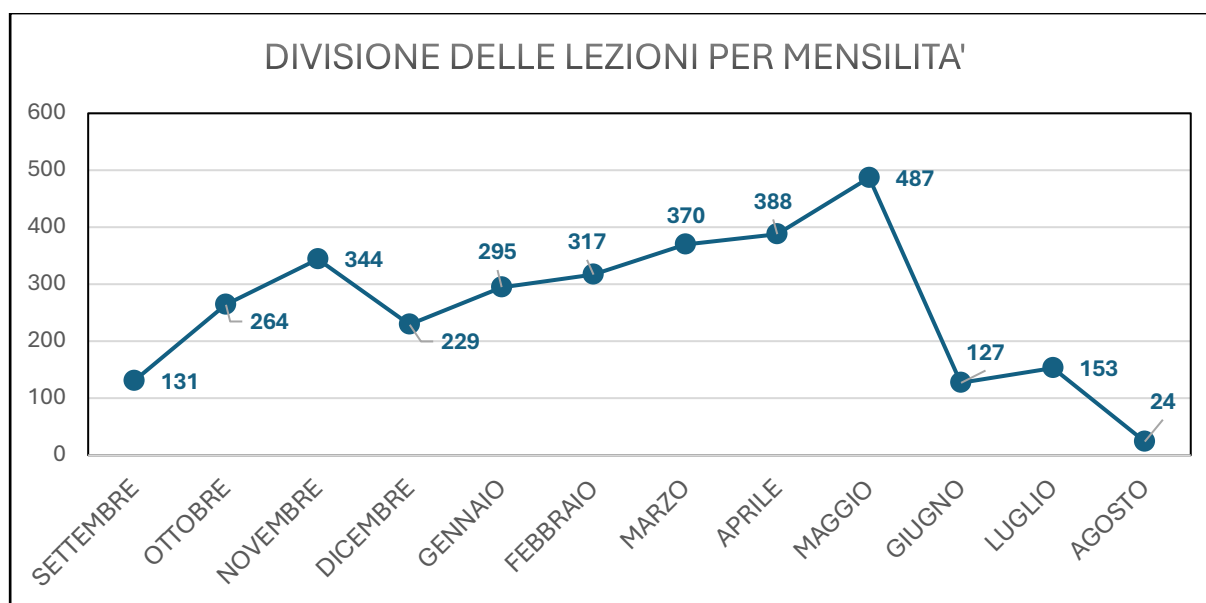


Tabella 8: Divisione lezioni per mensilità (A.A. 2023/24)

La richiesta variabile delle lezioni non è l'unico problema nella pianificazione del lavoro, essendo un servizio aperto al pubblico è difficile prevedere le caratteristiche degli studenti, come ad esempio la classe frequentata e la richiesta o meno di un percorso di formazione per DSA o BES. I dati raccolti sono i seguenti:

ELEMENTARI	36	15 %
MEDIE	74	32 %
SUPERIORI	115	49 %
UNIVERSITA'	9	4 %
TOTALE	234	

Tabella 9: Divisione studenti (A.A. 2023/24)

DSA SI	115	49 %
DSA NO	119	51 %

Tabella 10: Divisione studenti DSA o BES

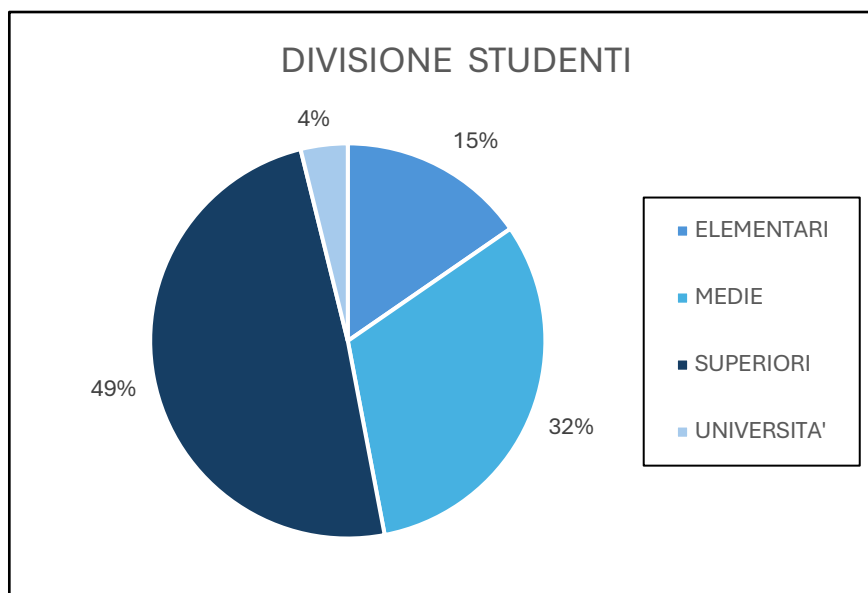


Tabella 11: Divisione studenti

Per ogni lezione è possibile richiedere una materia diversa che, nel caso di studenti delle scuole elementari o medie, non costituisce un problema in quanto tutti i tutor sono in grado di seguire tutte le materie, ma per studenti delle scuole di grado superiore può rappresentare un ostacolo. Le materie che sono state maggiormente richieste durante l'anno sono le seguenti:

COMPITI	1805	58 %
MATEMATICA	578	18 %
FISICA	147	5 %
INGLESE	124	4 %
CHIMICA	91	3 %
ECONOMICA	76	2 %
ITALIANO	57	2 %
ALTRE	241	8 %

Tabella 12: Divisione materie

Ad oggi per gestire le prenotazioni viene utilizzato un foglio di calcolo Excel in cui sono segnate le disponibilità future dei tutor, le lezioni svolte e prenotate suddivise nelle mensilità e, in alcuni casi, le materie svolte durante la lezione. Questo metodo non è ottimale perché è difficile tener conto di tutte le lezioni settimanali e degli abbinamenti migliori da fare data la grande quantità di richieste (circa cento lezioni ogni settimana) e data la poca flessibilità nel reperire informazioni dal software utilizzato, in quanto, questo strumento non permette di tener sotto controllo i percorsi degli studenti, le disponibilità dei tutor, le materie che sono in grado di seguire i tutor, i pagamenti delle lezioni e molti altri aspetti. Le prenotazioni vengono gestite per la maggior parte dei casi da una persona responsabile della segreteria, ma accettando prenotazioni sia in persona sia in modalità telematica è possibile che siano elaborate anche da altri dipendenti, per questo è importante che il software utilizzato sia di facile utilizzo e che possa fornire tutte le informazioni necessarie per una corretta prenotazione.

2.3: Input, Output e criticità

Per la gestione delle prenotazioni sono necessari diversi input, alcuni utili per registrare le informazioni necessarie per ottimizzare la ricerca degli abbinamenti ed altri funzionali alla prenotazione della lezione. Nel primo caso saranno fondamentali le informazioni relative ai tutor, come l'elenco delle materie che sono in grado di seguire, l'elenco delle disponibilità mensili e le informazioni generali sulla propria istruzione, mentre, per gli studenti, sono importanti le informazioni relative alla classe frequentata e alla presenza o meno di un certificato DSA o BES. Tutte queste informazioni sono inserite o modificate nelle finestre 'Gestisci Tutor' e 'Gestisci Studente' e saranno inserite durante la registrazione dell'utente o modificate in caso di necessità. Per prenotare la lezione verranno inserite le seguenti informazioni: nome e cognome studente, tutor, orario, giorno e materia trattata.

L'output generato dipenderà dalla scheda e dalle operazioni richieste, ad esempio nella scheda 'Gestisci studenti' saranno stampate le generalità relative allo studente come: nome, cognome, classe e l'elenco delle lezioni prenotate dalla data odierna in poi, la scheda 'Gestisci tutor' fornirà degli output del tutto simili a quelli visti per gli studenti, in quanto sarà possibile stampare le seguenti informazioni: nome e cognome del tutor, le informazioni relative alla sua formazione e l'elenco delle materie che garantisce di poter seguire.

Nella scheda 'Gestisci lezioni' sarà possibile reperire tutte le informazioni utili per la prenotazione delle lezioni come l'elenco dei tutor disponibili in un determinato giorno ed orario con il relativo punteggio di compatibilità, l'elenco delle disponibilità di un tutor specifico per poter eventualmente riprogrammare il giorno o l'orario nel caso in cui si voglia affidare la lezione al tutor ma quest'ultimo non sia disponibile nel giorno selezionato.

Per la sezione delle prenotazioni in automatico sarà restituito come output i migliori abbinamenti che il software sia riuscito a trovare, lasciando poi all'utente la possibilità di prenotarle oppure no.

La sezione 'Resoconto' restituirà diverse informazioni per il rendiconto delle lezioni tra cui: tutte le lezioni svolte dal tutor selezionato con la somma delle ore lavorative, tutte le lezioni realizzate da uno studente selezionato per il controllo dei pagamenti, le lezioni svolte nel mese suddivise per tutor e l'elenco delle lezioni prenotate dal giorno odierno fino alla fine del mese.

Queste informazioni saranno utili principalmente per il tesoriere dell'associazione per la rendicontazione e per la comunicazione ai tutor del proprio orario lavorativo, in quanto le

limitazioni date dal programma attualmente utilizzato fanno sì che la stesura del calendario delle lezioni avvenga il giorno prima e questo causi problemi organizzativi soprattutto per la reperibilità dei tutor esterni e un aumento del lavoro necessario per redigere il calendario dell'associazione.

Le criticità relative a questo progetto sono relative principalmente all'aspetto umano, in quanto capita sovente che i genitori o gli studenti stessi esprimano le proprie preferenze relative ad un tutor specifico o ad un metodo di studio, ad oggi queste richieste sono gestite dai referenti dell'associazione grazie alla loro esperienza. Per far fronte a queste richieste nel software sarà possibile, nel calcolo delle lezioni in modalità automatica, fissare le lezioni che non si vogliono ricalcolare in modo da lasciare prenotata una lezione anche se non è quella con il punteggio maggiore, lasciando libertà agli educatori di fare abbinamenti che reputano migliori secondo aspetti umani non rilevabili tramite il sistema di punteggio.

Un aspetto che è difficile da risolvere è quello del primo abbinamento, in quanto ad oggi, prima dell'iscrizione dello studente, viene svolto un colloquio con la famiglia per redigere una relazione sul ragazzo in modo da registrare non solo le sue generalità ma anche le sue carenze scolastiche, la sua personalità e la presenza di certificati DSA o BES in modo da capire quale possa essere l'abbinamento con il tutor migliore, in base anche alle caratteristiche del tutor e dal suo modo di impostare una lezione e di interagire con lo studente. Nei casi più particolari si preferisce sempre fare una lezione di prova con un educatore in modo da avere un resoconto più dettagliato sullo studente e poi eventualmente abbinarlo ai tutor più adatti.

Questi passaggi possono essere delicati e possono richiedere l'aiuto di personale specializzato quindi è lecito aspettarsi che i risultati ottenuti grazie alle prenotazioni in modalità automatica, soprattutto in alcuni casi, possano essere discordanti rispetto a quelli reali. Per sopperire a questa differenza la tabella dei punteggi è stata bilanciata in modo tale da avvantaggiare gli educatori di ruolo in caso di assegnazioni di studenti alla prima lezione con certificati DSA o BES, aggiungendo due punti in più rispetto ai tutor, di pari competenze scolastiche, ma senza un titolo da educatore.

Per fornire il miglior servizio possibile, per la ricerca dell'abbinamento migliore, si tiene conto del percorso svolto dallo studente ma anche della capacità del tutor di svolgere una lezione che possa essere adatta alle sue necessità e dalla sua competenza riguardante la materia svolta, andando così a preferire tutor con più esperienza o con maggiore maestria nell'insegnamento. Per il calcolo del punteggio automatico per una lezione non si tiene conto di questi fattori perché non esistono parametri oggettivi per preferire un tutor ad un altro nonostante entrambi abbiano

la stessa disponibilità a seguire la materia e conoscenza pregressa dello studente, ma si basano sull'empatia e sull'esperienza della persona preposta alla prenotazione della lezione. Per questo motivo ci attendiamo che nel calcolo automatico delle lezioni possano presentarsi più soluzioni con lo stesso punteggio.

2.4: Vincoli da rispettare

I vincoli da rispettare per compilare il calendario delle lezioni sono derivati dagli orari delle lezioni e dalla preparazione dei tutor: nel primo caso sarà possibile prenotare una lezione solo nel formato da due ore, può essere fatta un'eccezione per lezioni che si svolgano in orari extra-lavorativi ma non ne terremo conto per quest'analisi in quanto sono casi rari. Riguardo la preparazione dei tutor, non sarà possibile prenotare, in modalità automatica, lezioni che totalizzeranno un punteggio di zero, ovvero, uno studente frequentante le scuole superiori o università abbinato con un tutor che non ha le competenze per poter seguire la materia indicata. Gli studenti delle scuole elementari e medie troveranno sempre un tutor con un punteggio maggiore di zero, in quanto si dà per scontato che quest'ultimo sia in grado di poter seguire tutte le materie svolte nelle scuole di grado inferiore alle superiori grazie anche all'utilizzo di strumenti tecnologici forniti dall'associazione. Ad oggi è presente anche un limite sui banchi di lavoro ma non è stato preso in considerazione per lo sviluppo del software in quanto è attualmente in sviluppo un progetto per l'ampliamento dello spazio di lavoro e bisogna anche considerare che non tutti i banchi siano utilizzati esclusivamente per il servizio di tutoraggio, ma possono essere occupati per altri progetti, infatti risulta difficile stabilire a priori la disponibilità effettiva. Si lascia quindi all'operatore, in piena autonomia, la valutazione della disponibilità dei tavoli prima di prenotare una nuova lezione, come è stato fatto fino ad ora.

3. DESCRIZIONE DEL DATA-SET

3.1 Necessità di Input

Per l'analisi dei due metodi di prenotazione è fondamentale avere un database da cui partire in quanto serve per il calcolo del punteggio delle lezioni, per capire se il tutor ha già seguito in precedenza lo studente ed eventualmente aumentare il punteggio. Per questo i dati per compilare il database non saranno solo relativi all'anno attuale ma saranno relativi anche all'anno precedente (2023). Il data-set è stato compilato da zero quindi, prima di iniziare, è stato necessario uno studio preventivo per capire non solo cosa sarebbe stato necessario per lo sviluppo delle funzionalità presenti nel software ma anche cosa sarebbe stato possibile reperire, in quanto l'associazione non ha traccia di database relativi agli anni scolastici passati, né relativo alle lezioni né relativo agli studenti iscritti. Le informazioni reperibili sono quelle attraverso i fogli di calcolo Excel in cui sono registrati i calendari mensili con le lezioni prenotate da cui è stato possibile ricavare i nomi degli studenti iscritti, i tutor disponibili, le materie svolte, il giorno e l'orario. Da queste informazioni è stato possibile dedurre un elenco dei tutor che hanno lavorato nell'anno selezionato, un elenco delle lezioni svolte e un elenco di studenti in cui, in accordo con l'associazione, sono stati cambiati i nomi e cognomi per questioni di privacy.

3.2 Descrizione Data-Set

Il Data-Set è stato creato utilizzando il software '*Sequel Pro*' ed è strutturato in cinque tabelle: Disponibilità, Tutor, Studenti, Lezioni e Materie.

La tabella 'Tutor' è riempita con i seguenti dati:

- **id_Tutor:** È un identificativo univoco per i tutor.
- **nomeTutor:** Rappresenta il nome e cognome del tutor.
- **materieTutor:** Un elenco di materie che il tutor garantisce di seguire utilizzate per il calcolo dei punteggi per gli abbinamenti.
- **formazioneTutor:** Indica la formazione pregressa dell'educatore ma non influenza gli abbinamenti calcolati in modalità automatica, è utile nei casi in cui studenti

vogliono prenotare lezioni per materie di indirizzo, non comuni, per capire se qualche tutor abbia la formazione necessaria per garantire di fornire una lezione.

- **educatoreTutor:** Questo campo assume valori biunivoci ed indica se il tutor ha conseguito un titolo di studio che garantisca di seguire studenti con certificato DSA o BES e che li attesti come educatori.
- **Interno:** Indica se il lavoratore ha un contratto con l'associazione o se è un libero professionista.

La tabella relativa agli studenti contiene:

- **id_Stud:** È un identificativo univoco associato allo studente.
- **nomeStud:** Nome e Cognome dello studente.
- **classeStud:** Indica il livello scolastico dell'utente, può assumere i valori 'Elementare', 'Medie', 'Superiori' e 'Università'.
- **dsaStud:** È un valore biunivoco che indica se lo studente è in possesso di un certificato DSA o BES. Questo campo è un primo indicatore utile agli educatori per individuare e monitorare gli studenti con difficoltà nell'apprendimento, nella realtà però ci sono molti aspetti che devono essere attenzionati per la corretta diagnosi. Nei casi più delicati in cui è necessario l'affiancamento di un educatore sarà possibile indicarlo durante la prenotazione della lezione selezionando la materia 'Sostegno DSA'.

STUDENTE		
PK	id_Stud	INT
	nomeStud	VARCHAR
	classeStud	ENUM
	dsaStud	ENUM

Figura 1: Tabella Studente

TUTOR		
PK	id_Tutor	INT
	nomeTutor	VARCHAR
	materieTutor	VARCHAR
	formazioneTutor	VARCHAR
	educatoreTutor	ENUM
	interno	ENUM

Figura 2: Tabella Tutor

Nella tabella in cui vengono registrate le lezioni prenotate, sono contenuti i seguenti campi:

- **id_Lez:** È un identificativo univoco della lezione.
- **id_Stud & id_Tutor:** Indicano gli identificatori dello studente e del tutor che sono partecipi della lezione.
- **Data:** È un campo che contiene la data ed è nel formato 'DATE'.
- **Orario:** Indica l'orario della prenotazione e può assumere uno tra i valori '9-11', '11-13', '15-17' o '17-19'.
- **Materia:** Questo campo indica la materia su cui si baserà la lezione. Per evitare di avere materie discordanti fra di loro non sarà possibile inserire un nominativo da tastiera ma sarà possibile selezionarlo tramite un elenco a tendina.

Durante la fase di prenotazione di una lezione è fondamentale tener traccia delle disponibilità che sono state garantite dai tutor esterni per le mensilità, quest'ultime sono registrate nel database secondo la seguente struttura:

- **id_Disb:** Identificativo univoco delle disponibilità.
- **id_Tutor:** Indica l'identificativo del tutor a cui corrisponde la disponibilità.
- **Data & Orario:** Sono valori che indicano la data e l'orario della disponibilità, seguono le stesse proprietà dei campi presenti nella tabella 'Lezioni'.

All'interno della tabella 'Materie' è presente solo il campo '*nome*' nel quale sono elencati i nomi di tutte le materie selezionabili durante la prenotazione di una lezione.

DISPONIBILITA'		
PK	id_Disb	INT
	id_Tutor	INT
	Data	DATE
	Orario	ENUM

Figura 3: Tabella Disponibilità

LEZIONI		
PK	id_Lez	INT
	id_Stud	INT
	id_Tutor	INT
	Data	DATE
	Orario	ENUM
	Materia	VARCHAR

Figura 4: Tabella Lezioni

3.3 Diagramma ER

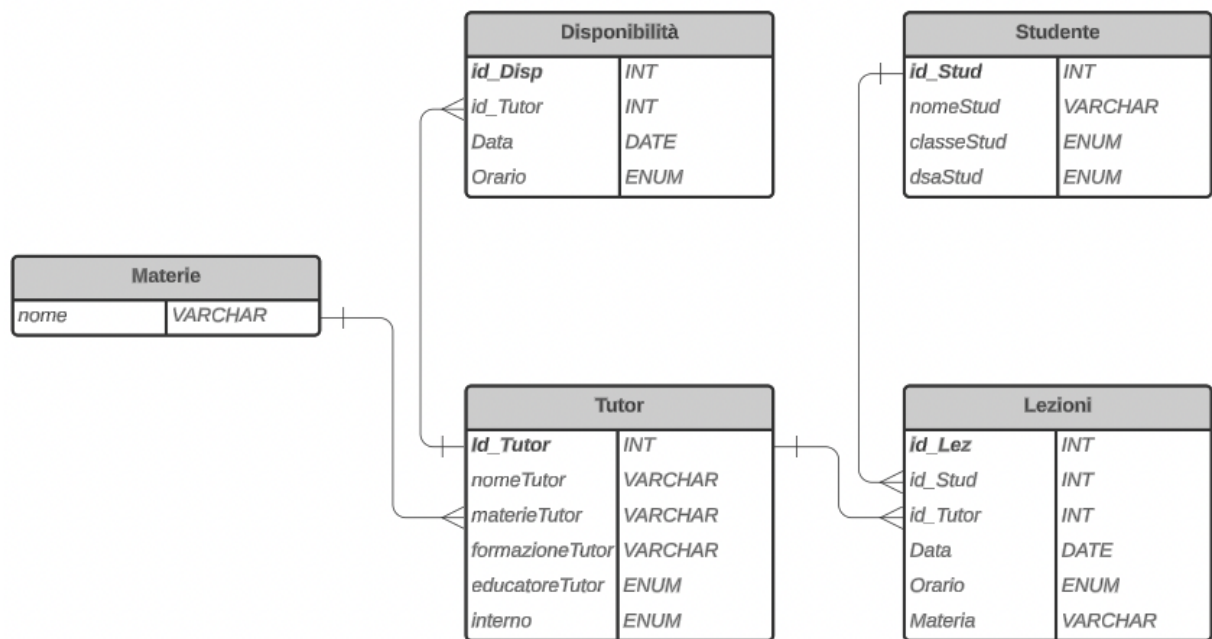


Figura 5: Diagramma Entità-Relazione

4. STRUTTURE DATI E ALGORITMI UTILIZZATI

4.1 Strutture dati

L'applicazione è stata realizzata in linguaggio Java e implementa il pattern MVC (Model View Controller) e il pattern DAO (Data Access Object), il progetto è composto da tre package utilizzati per separare il modello (Model), l'accesso al database (DAO) e l'interfaccia utente. Per realizzare l'interfaccia grafica in JavaFX è stato utilizzato il software '*SceneBuilder*'.

Il progetto si compone di tre package:

- ***ProvaGitt.Tesi:*** In questo package sono presenti tre classi, il *Main*, da cui viene eseguita l'applicazione, *Entry Point* e *FXMLController* nel quale sono definiti i metodi e i processi necessari per lo sviluppo del programma e che permettono la connessione tra il Model e l'interfaccia utente.
- ***ProvaGitt.Tesi.DAO:*** All'interno di questo package sono contenute le classi per il corretto collegamento e manipolazione del database utilizzato tra cui: *ConnectDB* che contiene i parametri necessari per l'accesso e garantisce la connessione, *LezioniDAO* è la classe in cui sono contenuti tutti i metodi utilizzati per estrarre tutte le informazioni dal data-set.
- ***ProvaGitt.Tesi.Model:*** Sono presenti cinque classi, quattro delle quali rappresentano gli oggetti visti in precedenza nel data-set, ovvero: *Disponibilità*, *Lezione*, *Studente* e *Tutor*. La classe *Model* contiene i metodi utilizzati per l'utilizzo dei dati raccolti grazie alla manipolazione del database e i metodi utili per le funzionalità di calcolo automatico dei punteggi delle lezioni e per l'algoritmo ricorsivo che permette all'applicazione di riorganizzare le prenotazioni delle lezioni giornaliere.

4.2 Algoritmi principali

Gli algoritmi più rilevanti per lo sviluppo sono quelli contenuti nel package *LezioniDAO* nel quale i metodi sono funzionali all'acquisizione delle informazioni contenute nel database, in Model dove saranno presenti modelli principalmente incentrati al calcolo ricorsivo per i metodi automatici e, infine, in *FXMLController* dove ci saranno i sottoprogrammi utili per lo svolgimento delle operazioni necessarie per l'interfaccia utente.

4.2.1 LezioniDAO & ConnectDB

Per l'accesso al database è stato utilizzato il framework HikaruCP, il suo funzionamento è basato sul concetto di connection pooling, che ha come scopo quello di ridurre il sovraccarico di gestione della connessione e la diminuzione delle attività di sviluppo per l'accesso ai dati, ad esempio la lettura/scrittura.

Le connessioni al database sono operazioni costose in termini di risorse ma che garantiscono risposte veloci soprattutto per l'acquisizione di informazioni specifiche ricavate da un data-set ampio.

In seguito, è riportato il codice per l'accesso al database, contenuto in *ConnectDB*:

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import com.zaxxer.hikari.HikariConfig;
import com.zaxxer.hikari.HikariDataSource;

public class ConnectDB
{
    private static String jdbcURL="jdbc:mariadb://localhost/SpecialmenteTu";
    private static HikariDataSource ds;
    public static Connection getConnection()
    {
        if (ds == null) {
            HikariConfig config = new HikariConfig();
            config.setJdbcUrl(jdbcURL);
            config.setUsername("user");

            //imposta password del tuo DB
            config.setPassword("password");

            // configurazione MySQL
            config.addDataSourceProperty("cachePrepStmts", "true");
            config.addDataSourceProperty("prepStmtCacheSize", "250");
            config.addDataSourceProperty("prepStmtCacheSqlLimit","2048");
            ds = new HikariDataSource(config);
        }
        try {
            return ds.getConnection();
        } catch (SQLException e) {
            e.printStackTrace();
            return null;
        }
    }
}
```

Per questione di sicurezza sono stato cambiati i valori di username e password per l'accesso al database SQL. All'interno di *LezioniDAO* viene utilizzato il linguaggio di programmazione SQL per interrogare il database e ricavare le informazioni richieste.

I metodi principali per l'interrogazione del database sono i seguenti:

- Inserimento nuovo Tutor
-

```
public boolean InserisciTutor (Tutor t)
{
    String sql = "INSERT IGNORE INTO `SpecialmenteTu`.`Tutor`
    (`id_Tutor`, `nomeTutor`, `materieTutor`,
    `formazioneTutor`, `educatoreTutor`, `interno`) VALUES(?,?,?,?,?,?,?)";
    boolean returnValue = false;

    try {
        // ...
    }
}
```

- Cancellazione Tutor
-

```
public boolean CancellaTutor (int id)
{
    String sql = "DELETE FROM Tutor AS t "
    + "WHERE t.`id_Tutor` = ?";
    boolean returnValue = false;
    try {
        // ...
    }
}
```

Questi metodi di inserimento e cancellazione dei tutor sono assimilabili a quelli utilizzati per gli studenti, le lezioni e le disponibilità.

- Conteggio del numero di identificativi
-

```
public Integer getContoTutor ()
{
    final String sql = "SELECT MAX(t.`id_Tutor`) AS c "
                      + "FROM Tutor AS t";
    int tot = 0;
    try {
        Connection conn = ConnectDB.getConnection();
        PreparedStatement st = conn.prepareStatement(sql);
        ResultSet rs = st.executeQuery();

        if (rs.next())
        {
            tot = rs.getInt("c");
        }

        // ...
    }
}
```

Per l'aggiunta di un nuovo tutor è necessario assegnargli un identificativo univoco che è rappresentato da un numero crescente che parte da 1. Visto che gli utenti possono essere rimossi dal database è necessario estrarre quello che è il numero di identificativo maggiore per poter sapere quale sarà quello successivo da assegnare.

Il metodo di conteggio è utilizzato anche per gli identificativi di studenti, lezioni e disponibilità.

- Richiesta elenco tutor
-

```
public List <Tutor> getTutor ()
{
    final String sql = "SELECT t.`id_Tutor`, t.`nomeTutor`,
                        t.`materieTutor`, t.`formazioneTutor`, t.`educatoreTutor`,
                        t.`interno`"
                      + "FROM Tutor AS t";
    List<Tutor> tutor = new ArrayList<Tutor>();

    // Creazione di un oggetto della classe Tutor

    // ...
}
```

- Richiesta lezioni prenotate per tutor
-

```
public List <Lezione> getResocontoLezioniPrenotateTutor (int DayY, int
id, int anno)
{
    final String sql = "SELECT l.`id_Lez`, l.`id_Stud`, l.`Data`, l.`Orario`,
l.`Materia`, DAY(l.`Data`) AS g, MONTH (l.`Data`) AS m, YEAR (l.`Data`) AS
a, DAYOFYEAR(l.`Data`) AS d, WEEKDAY (l.`Data`) AS w "
+ "FROM Lezioni AS l "
+ "WHERE l.`id_Tutor` = ? AND DAYOFYEAR(l.`Data`) >= ? AND
YEAR(l.`Data`) = ? "
+ "ORDER BY l.`Data`, l.`Orario`";

    List<Lezione> lezioni = new ArrayList<Lezione>();
    try
    {
        Connection conn = ConnectDB.getConnection();
        PreparedStatement st = conn.prepareStatement(sql);
        st.setInt(1, id);
        st.setInt(2, DayY);
        st.setInt(3, anno);
        ResultSet rs = st.executeQuery();
        while (rs.next()) {
            // Inizializzazione delle lezioni e aggiunta a 'lezioni'
        }
        conn.close();
        return lezioni;
    } catch (SQLException e) {
        e.printStackTrace();
        throw new RuntimeException(e);
    }
}
```

- Richiesta tutor disponibili in data selezionata
-

```
public List <Integer> getTutorDisponibili(int DayY, int anno, String ora)
{
    final String sql = "SELECT d.`id_Tutor` "
+ "FROM Disponibilita AS d "
+ "WHERE YEAR(d.`Data`) = ?
AND DAYOFYEAR(d.`Data`)= ? AND d.`Orario` = ? "
+ "GROUP BY d.`id_Tutor`";
    List<Integer> tutor = new ArrayList<Integer>();

    // ...

}
```

- Richiesta lezioni tutor-studente già effettuate
-

```
public List <Lezione> getLezioniTutorStudenteFatte (int idStud, int idTut,
int DayY, int anno)
{
    final String sql = "SELECT l.`id_Lez`, l.`id_Stud`, l.`id_Tutor`,
l.`Data`, l.`Orario`, l.`Materia`, DAY(l.`Data`) AS g, MONTH
(l.`Data`) AS m, YEAR (l.`Data`) AS a, DAYOFYEAR(l.`Data`) AS d,
WEEKDAY (l.`Data`) AS w "
+ "FROM Lezioni AS l "
+ "WHERE l.`id_Stud`= ? AND l.`id_Tutor` = ?
AND((DAYOFYEAR(l.`Data`)< ? AND YEAR(l.`Data`) = ?) OR
(YEAR(l.`Data`) < ?)) "
+ "ORDER BY l.`Data`, l.`Orario`";

    List<Lezione> lezioni = new ArrayList<Lezione>();

    // ...

}
```

- Richiesta lezioni prenotate nel giorno selezionato
-

```
public List<Lezione> getLezioniGiorno (int DayY, int anno)
{
    final String sql = "SELECT l.`id_Lez`, l.`id_Stud`, l.`id_Tutor`,
l.`Data`, l.`Orario`, l.`Materia`, DAY(l.`Data`) AS g,
MONTH(l.`Data`) AS m, YEAR (l.`Data`) AS a, DAYOFYEAR(l.`Data`)
AS d, WEEKDAY (l.`Data`) AS w "
+ "FROM Lezioni AS l "
+ "WHERE DAYOFYEAR(l.`Data`) = ? AND YEAR(l.`Data`)= ? "
+ "ORDER BY l.`Data`, l.`Orario`";

    List<Lezione> lezioni = new ArrayList<Lezione>();

    // ...

}
```

Questo metodo è utilizzato principalmente per ottenere le lezioni prenotate in un giorno selezionato, per la stampa del calendario giornaliero e per salvare una lista di lezioni da utilizzare per il programma ricorsivo per il calcolo delle lezioni giornaliere.

4.2.2 FXMLController

Il Controller ha il compito di collegare il livello Model, dove sono contenuti tutti i metodi con rilevanza algoritmica, e l'interfaccia utente, che permette all'utilizzatore di visualizzare le finestre di output e di utilizzare il software. Di seguito sono riportati i metodi principali presenti in FXMLController:

- Aggiungi Studente
-

```
void doAggiungiStudente4(ActionEvent event)
{
    this.Area4.clear();
    try {
        String name =
        this.txtNomeStud4.getText().substring(0,1).toUpperCase() +
        this.txtNomeStud4.getText().substring(1,this.txtNomeStud4.getText().length())
            .toLowerCase();
        String surname =
        this.txtCognomeStud4.getText().substring(0,1).toUpperCase() +
        this.txtCognomeStud4.getText().substring(1,this.txtCognomeStud4.getText().length()).
        toLowerCase();
        String nome = name.replaceAll(" ", "")+" "+surname.replaceAll(" ", "");
        // Controllo se lo studente è già stato aggiunto
        if(model.trovaIdStudente(model.getStudenti(), nome ) != -1)
        {
            this.Area4.setText("STUDENTE GIA' INSERITO");
            return;
        }
        // Controllo il corretto inserimento dei dati
        // ...
        Studente s = new Studente (model.getContoStudenti()+1, nome, classe,dsa);
        if(model.InserisciStudente(s))
        {
            this.Area4.setText("STUDENTE AGGIUNTO CORRETTAMENTE! ");
            return;
        }else {
            this.Area4.setText("ERRORE NELL'AGGIUNTA DEL TUTOR");
            return;
        }
    }
    catch(NumberFormatException e)
    {
        this.Area4.setText("ATTENZIONE!\n IMPOSSIBILE AGGIUNGERE
        STUDENTE");
    }
    catch(RuntimeException e)
    {
        this.Area4.setText("ERRORE DI CONNESSIONE AL DATABASE");
    }
}
```

- Aggiungi Tutor
-

```
void doAggiungiTutor6(ActionEvent event)
{
    this.Area6.clear();
    try {
        String name =
this.txtNomeTutor6.getText().substring(0,1).toUpperCase() +
this.txtNomeTutor6.getText().substring(1,this.txtNomeTutor6.getText().length
()).toLowerCase();
        String surname =
this.txtCognomeTutor6.getText().substring(0,1).toUpperCase() +
this.txtCognomeTutor6.getText().substring(1,this.txtCognomeTutor6.getText().
length()).toLowerCase();
        String nome = name.replaceAll(" ", "")+" "+surname.replaceAll(" ", "");

        if(model.trovaIdTutor(model.getTutor(), nome) != -1)
        {
            this.Area6.setText("TUTOR GIA' INSERITO");
            return;
        }
        boolean educatore = false;
        boolean interno = false;

        if(this.CheckEducatoreYes6.isSelected())
        {
            educatore = true;
        }
        if(this.CheckInternoYes6.isSelected())
        {
            interno = true;
        }

        Tutor t = new Tutor (model.getContoTutor()+1, nome, Materia6, form,
educatore, interno);
        // Aggiorno tutti i menù a tendina presenti nelle interfacce
        if(model.InserisciTutor(t))
        {
            this.Area6.setText("TUTOR AGGIUNTO CORRETTAMENTE! ");
            this.BoxTutor1.getItems().removeAll(Tutor);
            this.BoxTutor5.getItems().removeAll(Tutor);
            this.BoxTutor7.getItems().removeAll(Tutor);
            this.Tutor = model.getTutor();
            this.BoxTutor1.getItems().addAll(Tutor);
            this.BoxTutor5.getItems().addAll(Tutor);
            this.BoxTutor7.getItems().addAll(Tutor);
            this.Materia6.clear();
            return;
        }else {
            this.Area6.setText("ERRORE NELL'AGGIUNTA DEL TUTOR");
            return;
        }
    }
    // Controllo le eccezioni
    // ...
}
```

- Prenotazione lezione singola

```
void doPrenotaLezione1(ActionEvent event)
{
    this.Area1.clear();
    try {
        // Elaboro le stringhe del nome e cognome
        // ...
        Studente stud = model.getStudianteByNome(nome);

        // Controllo che i dati inseriti dall'utente siano corretti
        // ...

        List <Integer> id = model.getTutorDisponibili(d.getDayOfYear(),
            d.getYear(), orario);
        if(id.isEmpty())
        {
            this.Area1.setText("NESSUNA DISPONIBILITA' DEL TUTOR ");
            return;
        }

        if(id.contains(tutor.getIdTutor()))
        {
            int tot = model.getContoLezioni()+1;
            Calendar cal = Calendar.getInstance();
            cal.set(Calendar.YEAR, d.getYear());
            cal.set(Calendar.MONTH, d.getMonthValue() - 1);
            cal.set(Calendar.DAY_OF_MONTH, d.getDayOfMonth());
            java.sql.Date date = new java.sql.Date(cal.getTimeInMillis());
            Lezione lez = new Lezione (...);
            List <Lezione> lezioni =
            model.getResocontoLezioniPrenotateTutor(d.getDayOfYear(),
            tutor.getIdTutor(), d.getYear());
            for(int i=0; i<lezioni.size(); i++)
            {
                if((lezioni.get(i).getOrario().equals(lez.getOrario())) &&
                (lezioni.get(i).getDayY() == d.getDayOfYear()) &&
                (lezioni.get(i).getAnno() == d.getYear()))
                {
                    this.Area1.setText("IL TUTOR NON E' DISPONIBILE");
                    return;
                }
            }
            // Controllo che la lezione rispetti i criteri per la prenotazione
            // ...

            if(model.InserisciLezione(lez))
            {
                this.Area1.setText("LEZIONE PRENOTATA CON SUCCESSO! ");
                return;
            }else {
                this.Area1.setText("ERRORE NELLA PRENOTAZIONE DELLA LEZIONE");
                return;
            }
        }
    }
}
```

- Verifica disponibilità
-

```
void doVerificaDisponibilita1(ActionEvent event)
{
    this.Area1.clear();
    try {
        // Elaboro le stringhe del nome e cognome
        // ...

        Studente stud = model.getStudenteByNome(nome);

        // Controllo che i dati inseriti dall'utente siano corretti
        // ...

        List <Integer> id = model.getTutorDisponibili(d.getDayOfYear(),
            d.getYear(), orario);
        if(id.isEmpty())
        {
            this.Area1.setText("NESSUNA DISPONIBILITA' DI TUTOR ");
            return;
        }
        List <Tutor> tutor = new ArrayList <Tutor> ();
        for(int i=0; i<id.size(); i++)
        {
            Tutor t = model.trovaTutor(model.getTutor(), id.get(i));
            tutor.add(t);
        }

        List <Lezione> l = new ArrayList <Lezione> ();
        int conto = model.getContoLezioni();
        for(int i=0; i<tutor.size(); i++)
        {
            Calendar cal = Calendar.getInstance();
            cal.set(Calendar.YEAR, d.getYear());
            cal.set(Calendar.MONTH, d.getMonthValue() - 1);
            cal.set(Calendar.DAY_OF_MONTH, d.getDayOfMonth());
            java.sql.Date date = new java.sql.Date(cal.getTimeInMillis());
            Lezione lez = new Lezione (conto+1, stud.getIdStudente(),
            tutor.get(i).getIdTutor(), date, orario, materia, d.getDayOfMonth(),
            d.getMonth().getValue(), d.getYear(),
            d.getDayOfYear(), d.getDayOfWeek().getValue());
            l.add(lez);
        }

        // Stampo un elenco di tutor disponibili con il proprio punteggio
        this.Area1.appendText("ELENCO TUTOR DISPONIBILI: \n" );
        for(int i=0; i<l.size(); i++)
        {
            this.Area1.appendText("Tutor: "+ model.trovaTutor(model.getTutor(),
            l.get(i).getIdTutor()) + "PUNTI:"+model.calcolaPunteggio(l.get(i))+
            "\n" );
        }

        // Controllo le eccezioni
        // ...
    }
}
```

- Calcolo lezioni multiple
-

```
List <Lezione> Lezioni2 = new ArrayList <Lezione> ();

void doCalcolaLezioniMultiple2(ActionEvent event)
{
    this.Area2.clear();
    if(this.Lezioni2.isEmpty())
    {
        this.Area2.setText("INSERIRE UNA LEZIONE PRIMA DI CALCOLARE LE LEZIONI");
        return;
    }
    List <Lezione> migliori = new ArrayList <Lezione> ();
    for(int i=0; i<Lezioni2.size(); i++)
    {
        List <Integer> id =
        model.getTutorDisponibili(Lezioni2.get(i).getDayY(),
        Lezioni2.get(i).getAnno(), Lezioni2.get(i).getOrario());
        if(id.isEmpty())
        {
            this.Area2.setText("NESSUNA DISPONIBILITA' DI TUTOR PER IL
            GIORNO: "+Lezioni2.get(i).getData()+" E ORARIO:"
            +Lezioni2.get(i).getOrario());
            return;
        }
        int max = 0;
        Lezione migliore = null;
        for(int j=0; j<id.size(); j++)
        {
            Lezione l = new Lezione (-1, Lezioni2.get(i).getIdStudente(),
            id.get(j), Lezioni2.get(i).getData(),
            Lezioni2.get(i).getOrario(), Lezioni2.get(i).getMateria(),
            Lezioni2.get(i).getGiorno(), Lezioni2.get(i).getMese(),
            Lezioni2.get(i).getAnno(),
            Lezioni2.get(i).getDayY(), Lezioni2.get(i).getNumGiorno());
            if(model.calcolaPunteggio(l)>max)
            {
                if(model.isCorrettaLezione(l, migliori))
                {
                    max = model.calcolaPunteggio(l);
                    migliore = l;
                }
            }
        }
        if(max != 0)
        {
            migliori.add(migliore);
        } else {
            this.Area2.setText("NESSUNA LEZIONE PRENOTABILE ");
            return;
        }
    }
    Lezioni2 = migliori;
}
```

Questo metodo è utilizzato per il calcolo automatico delle migliori lezioni possibili se si vuole prenotare più lezioni nello stesso momento, è utile perché l'associazione ha varie promozioni attive che garantiscono sconti se si acquista e si prenota in singola data un blocco di lezioni (cinque o dieci). La prenotazione non avviene in automatico ma il calcolo delle migliori lezioni verrà stampato a schermo e, se approvato da un educatore, potrà essere confermato tramite il pulsante "PRENOTA", in seguito saranno cancellati gli oggetti presenti nella lista '*Lezione2*'.

4.3 Algoritmo ricorsivo

L'ottenimento di un elenco di prenotazioni in modalità automatica si basa su un algoritmo ricorsivo con una ricerca di tutte le combinazioni possibili (backtracking), assegnando ad ognuna di esse un punteggio, l'obiettivo è quello di massimizzare i punti totali pur rispettando alcuni vincoli imposti dall'associazione.

Per esigenze logistiche, la tabella dei punteggi è suddivisa in tre parti: la prima riguardante studenti che frequentano le scuole elementari o medie, la seconda relativa a scuole superiori e università, mentre l'ultima è pensata per gli studenti che hanno bisogni dell'apprendimento specifici e richiedono l'aiuto di un educatore. Tutte le tabelle saranno divise in due parti, la prima riguardante uno studente con certificato DSA, mentre la seconda riguarda studenti che ne sono sprovvisti.

4.3.1 Calcolo punteggio lezione

Nella tabella relativa a studenti frequentanti scuole elementari o medio non è riportata la voce 'Materia' perché si assume che il tutor possa seguire tutte le materie svolte, grazie anche a strumenti tecnologici forniti e all'aiuto dei colleghi. Per questo motivo non è plausibile che sia impossibile assegnare un tutor con uno studente, assegnandogli quindi un punteggio pari a zero.

	Già Seguito	Tutor Educatore	PUNTEGGIO
Studente DSA	No	No	1
	No	Sì	2
	Sì	No	2
	Sì	Sì	3
Studente non DSA	No	No	1
	No	Sì	1
	Sì	No	2
	Sì	Sì	2

Tabella 13: Punteggio scuole Elementari-Medie

Riguardo a studenti delle scuole superiori o università entra in gioco anche il fattore 'Materia' che impone già un primo vincolo da rispettare, in quanto, se il tutor non garantisce di poter seguire una determinata materia richiesta dallo studente sarà assegnato un punteggio pari a zero, rendendo la prenotazione non valida. Il vincolo dello studente DSA viene meno quando un tutor ha già seguito con successo il medesimo studente, questo perché generalmente in seguito ad una lezione, con un abbinamento nuovo, viene effettuata una breve riunione con un educatore al fine di evidenziare possibili problematiche riscontrate.

I tutor che hanno già seguito uno studente per la stessa materia richiesta per la lezione hanno un punteggio maggiore rispetto agli altri in quanto, come da direttiva dell'associazione, si vuole dare continuità al percorso degli studenti con i propri referenti.

In seguito, è riportata la tabella:

	Materia	Già Seguito	Educatore	PUNTEGGIO
Studente DSA	No			0
	Sì	No	No	1
	Sì	No	Sì	2
	Sì	Sì	No	3
	Sì	Sì	Sì	3
	Sì	Si, seguito per materia	No	4
	Sì	Si, seguito per materia	Sì	5
Studente non DSA	No			0
	Sì	No	No	2
	Sì	No	Sì	2
	Sì	Sì	No	3
	Sì	Sì	Sì	3
	Sì	Si, seguito per materia	No	4
	Sì	Si, seguito per materia	Sì	4

Tabella 14: Punteggio scuole Superiori-Università

Una particolare attenzione è riposta sugli studenti che hanno bisogni dell'apprendimento specifici, per i quali, deve essere stipulato insieme ai genitori un percorso educativo specifico riguardante tutto l'anno scolastico. Non è possibile, sia in modalità manuale che automatica, assegnare una lezione in cui è segnalato come materia "*Sostegno DSA*" ad un tutor che non ha una qualifica come educatore.

SOSTEGNO DSA	
Tutor Educatore	PUNTEGGIO
No	0
Sì	1

Tabella 15: Punteggio Sostegno DSA

4.3.2 Calcola lezioni giorno

Il metodo ricorsivo principale *cerca* è contenuto nel package '*Model*' e utilizza in input una lista di lezioni già prenotate in cui è indicato un tutor provvisorio per ciascuna (*tutte*), una lista di lezioni che rappresentano una possibile soluzione (*parziale*), un contatore (*cont*) che rappresenta il vincolo di lunghezza della soluzione finale ed una lista con il numero identificativo dei tutor disponibili per la lezione (*id*).

I passaggi seguiti dal metodo sono i seguenti:

1. Controlla la dimensione della soluzione parziale, se è uguale alla dimensione della di lista contenente tutte le lezioni prenotate allora verifica se ha un punteggio totale maggiore rispetto alla soluzione migliore trovata in precedenza (*isMeglio*) e se rispetta tutti i parametri richiesti (*isCorretto*) , in caso affermativo la lista parziale viene salvata come la migliore trovata, in caso negativo si esce dal metodo senza alterare nulla.
2. Nel caso in cui la dimensione della soluzione parziale sia minore rispetto a quella generale, si procede con un ciclo iterativo nel quale, se non sono stati aggiunti precedentemente, verranno aggiunti uno per volta tutti i tutor disponibili. In seguito all'inserimento di un tutor nuovo, viene creato un nuovo oggetto *lezione* e successivamente viene richiamato il metodo ricorsivo *cerca*, per poi eseguire un'operazione di backtracking e rimuovere la lezione appena aggiunta.

- Metodo ricorsivo 'cerca'
-

```
public void cerca(List <Lezione> parziale, int cont, List <Lezione> tutte,
                 List <Integer> id)
{
    if(cont == tutte.size())
    {
        // Controllo se parziale è migliore della soluzione
        if((isMeglio(parziale, LezMigliori) == true) &&
            (isCorretto(parziale) == true))
        {
            if(isNonZero(parziale) == true)
            {
                LezMigliori = new ArrayList <> (parziale);
                punteggio = this.calcolaPunteggioTot(LezMigliori);
            }
        }
    }
    for(int i=0; i<tutte.size(); i++)
    {
        for(int j=0; j<id.size(); j++)
        {
            if( (!isTutorAggiunto(parziale, id.get(j), tutte.get(i).getOrario()))
            && (!isLezioneAggiunta(parziale, tutte.get(i).getIdLezione())))
            {
                Lezione l = new Lezione (tutte.get(i).getIdLezione(),
                tutte.get(i).getIdStudente(), id.get(j), tutte.get(i).getData(),
                tutte.get(i).getOrario(), tutte.get(i).getMateria(), tutte.get(i).getGiorno(),
                tutte.get(i).getMese(), tutte.get(i).getAnno(), tutte.get(i).getDayY(),
                tutte.get(i).getNumGiorno());
                if(this.calcolaPunteggio(l)>0)
                {
                    parziale.add(l);
                    cerca(parziale, cont+1, tutte, fisse, id);
                    parziale.remove(parziale.size()-1);
                }
            }
        }
    }
}
```

Il metodo ricorsivo viene richiamato nel package model nel metodo 'getSoluzioneMigliore' che riceve in input la lista di tutte le lezioni prenotate nella data selezionata e restituisce come output la lista di lezioni che costituiscono la soluzione migliore. Nell'eventualità in cui al proprio interno sia contenuta almeno una lezione che non ha un tutor disponibile che possa svolgerla, quindi nel caso di punteggio pari a zero, il metodo restituirà una lista vuota e l'errore verrà segnalato come output quando l'utente cercherà di calcolare le lezioni in maniera automatica grazie al bottone "CALCOLA LEZIONI GIORNALIERE".

- Calcolo delle soluzioni migliori
-

```
public List <Lezione> getSoluzioneMigliore (List <Lezione> tutte , List
<Lezione> fisse)
{
    List <Lezione> totale = new ArrayList <Lezione> ();
    this.LezMigliori = new ArrayList <Lezione> ();
    List <Lezione> parziale1 = new ArrayList <Lezione> ();
    List <Lezione> tutte1 = new ArrayList <Lezione> ();
    // Ripeto la creazione delle liste per altre tre volte

    // Suddivido le lezioni in base al rispettivo orario
    for(int i=0; i<tutte.size(); i++)
    {
        if(tutte.get(i).getOrario().equals("9-11"))
        {
            tutte1.add(tutte.get(i));
        }
        if(tutte.get(i).getOrario().equals("11-13"))
        {
            tutte2.add(tutte.get(i));
        }
        if(tutte.get(i).getOrario().equals("15-17"))
        {
            tutte3.add(tutte.get(i));
        }
        if(tutte.get(i).getOrario().equals("17-19"))
        {
            tutte4.add(tutte.get(i));
        }
    }
    // Per ogni List eseguo l'algoritmo ricorsivo 'cerca', se non è vuota
    // Salvo la soluzione migliore in 'totale'

    if(!tutte1.isEmpty())
    {
        List <Integer> id1 = this.getTutorDisponibiliOra(tutte1, fisse);
        cerca (parziale1,0,tutte1, fisse, id1);
        if(!LezMigliori.isEmpty())
        {
            List <Lezione> tot1 = LezMigliori;
            for(int i=0; i<tot1.size(); i++)
            {
                totale.add(tot1.get(i));
            }
        }else
        {
            this.LezMigliori.clear();
            return this.LezMigliori;
        }
    }
    this.LezMigliori = new ArrayList <Lezione> ();
    this.LezMigliori = new ArrayList <Lezione> (totale);
    return LezMigliori;
}
```

Al fine di ottimizzare la ricerca automatica, tramite algoritmo ricorsivo, il metodo 'getSoluzioneMigliore' ha il compito di creare quattro liste differenti, suddivise in base all'orario di prenotazione, per garantire una minore complessità di calcolo. Lo stesso meccanismo è utilizzato dal metodo 'getTutorDisponibiliOra' che restituisce solo i tutor che hanno garantito la loro disponibilità e che non sono già stati assegnati ad un'altra lezione che si è deciso di fissare, e quindi, di non ricalcolare tramite algoritmo. Queste due soluzioni sono state pensate per garantire un minor tempo di risposta del programma per il calcolo della soluzione migliore, tramite la suddivisione di un problema (trovare i migliori abbinamenti della giornata) in più sotto-problemi (trovare i migliori abbinamenti per i tutor disponibili per un orario indicato).

- Controllo se il tutor è stato già aggiunto
-

```
public boolean isTutorAggiunto (List <Lezione> lez, int id)
{
    for(int i=0; i<lez.size(); i++)
    {
        if((lez.get(i).getIdTutor() == id) &&
            (lez.get(i).getOrario().equals(orario)))
            return true;
        }
    }
    return false;
}
```

- Controllo se la lezione è stata già aggiunta
-

```
public boolean isLezioneAggiunta (List <Lezione> lez, int id)
{
    for(int i=0; i<lez.size(); i++)
    {
        if(lez.get(i).getIdLezione() == id)
            return true;
        }
    }
    return false;
}
```

- Controllo se la lista parziale è migliore della soluzione migliore
-

```
public boolean isMeglio (List <Lezione> parz, List <Lezione> tot)
{
    if(tot.isEmpty())
    {
        return true;
    }
    if(this.calcolaPunteggioTot(parz)>this.calcolaPunteggioTot(tot))
    {
        return true;
    }

    if(this.calcolaPunteggioTot(parz)==this.calcolaPunteggioTot(tot))
    {
        if(this.calcolaInterno(parz)>this.calcolaInterno(tot))
        {
            return true;
        }
        if(this.calcolaInterno(parz)==this.calcolaInterno(tot))
        {
            if (menoTutor(parz, tot))
            {
                return true;
            }
        }
    }
    return false;
}
```

Con il metodo 'menoTutor' si calcola quale delle due soluzioni richiede il minor numero di tutor, in modo da non dover contattare più liberi professionisti nel caso di soluzioni equivalenti.

- Controllo se è rispettato il vincolo delle lezioni non prenotabili
-

```
public boolean isNonZero (List<Lezione> l)
{
    for(int i=0; i<l.size(); i++)
    {
        if(this.calcolaPunteggio(l.get(i)) == 0)
        {
            return false;
        }
    }
    return true;
}
```

- Controllo se la soluzione parziale rispetta il vincolo di non avere doppioni
-

```
public boolean isCorretto (List <Lezione> parz)
{
    List <Integer> id = new ArrayList <Integer> ();
    for(int i = 0; i<parz.size(); i++)
    {
        id.add(parz.get(i).getIdLezione());
    }
    for(int i=0; i<id.size(); i++)
    {
        int cont = 0;
        for(int j=0; j<id.size(); j++)
        {
            if(id.get(i)==id.get(j))
            {
                cont++;
            }
        }
        if(cont>1)
            return false;
    }
    return true;
}
```

- Calcolo il numero di tutor interni
-

```
public int calcolaInterno (List <Lezione> l)
{
    int result = 0;
    for(int i=0; i<l.size(); i++)
    {
        Tutor t = this.trovaTutor(getTutor(), l.get(i).getIdTutor());
        if(t.isInterno())
        {
            result++;
        }
    }
    return result;
}
```

Questo metodo viene utilizzato nei casi in cui due soluzioni parziali ottengano lo stesso punteggio, allora si preferisce la soluzione che impiega più tutor interni al fine di massimizzare i ricavi dell'associazione.

- Fissa le lezioni della giornata
-

```
void doFissaLezione3 (ActionEvent event)
{
    this.Area3.clear();
    try
    {
        Lezione l = this.BoxLezioni3.getValue();
        if(l==null)
        {
            this.Area3.setText("SELEZIONARE UNA LEZIONE PRIMA DI PROSEGUIRE");
            return;
        }
        this.LezFissate.add(l);
        this.Lezioni3.remove(l);
        this.Area3.appendText("LEZIONE FISSATA: "+l.toString()+"\n");
        this.Area3.appendText("LEZIONI DISPONIBILI: \n");
        for(int i=0; i<this.Lezioni3.size(); i++)
        {
            this.Area3.appendText(this.Lezioni3.get(i)+"\n");
        }

        this.BoxLezioni3.getItems().clear();
        this.BoxLezioni3.getItems().addAll(this.Lezioni3);
    }catch(NumberFormatException e)
    {
        this.Area3.setText("ATTENZIONE!\n ERRORE NELLA STAMPA DELLE LEZIONI GIORNALIERE ");
    }catch(RuntimeException e)
    {
        this.Area3.setText("ERRORE DI CONNESSIONE AL DATABASE");
    }
}
```

Questo metodo presente in FXMLController permette di selezionare, tramite ComboBox, una o più lezioni da voler 'fissare' in modo da non venire calcolate tramite il metodo ricorsivo.

5. OUTPUT GRAFICO

L'interfaccia è stata suddivisa in quattro pagine: Gestisci Lezione, Gestisci Studente, Gestisci Tutor e Resoconto. Questa suddivisione è stata pensata per rendere l'interfaccia di facile accesso per tutti gli operatori, in modo che sia utilizzabile sia dai responsabili della segreteria sia da quelli preposti alla rendicontazione.

Le interfacce grafiche sono state realizzate con *JavaFX*, utilizzando l'applicazione *SceneBuilder*.

5.1 Elementi comuni

Nelle tab principali ci sono degli elementi comuni utilizzati spesso per praticità e facilità nel reperimento delle informazioni necessarie per la prenotazione delle lezioni. Visto che l'obiettivo del programma è quello di essere utilizzato da tutti i dipendenti dell'associazione, una delle prime considerazioni fatte per lo sviluppo dell'interfaccia è stata quella di ridurre al minimo la possibilità di inserire informazioni errate durante le operazioni di segreteria. Per questo motivo, i tutor potranno essere selezionati attraverso l'utilizzo di una *ComboBox*, che sarà riempita all'apertura dell'interfaccia con l'elenco di tutti i dipendenti presenti nel database. Lo stesso procedimento verrà effettuato per la scelta della materia svolta durante la lezione, il database contiene nella tabella 'Materie' un elenco di tutte le materie prenotabili, alcune di esse presentano un simbolo '+' al termine del nome, quest'ultimo indica che la materia è richiesta ad un livello avanzato rispetto a quello che viene richiesto generalmente in un programma scolastico, ad esempio, alcune scuole forniscono dei corsi con potenziamento per specifiche materie scientifiche (es. chimica, matematica, fisica, ...) oppure alcune università che richiedono una conoscenza più approfondita della materia.

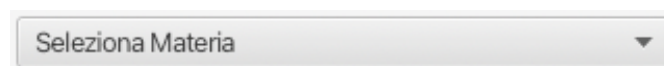


Figura 6: ComboBox Materie

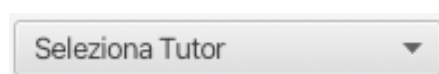


Figura 7: ComboBox Tutor

Per l'inserimento dei dati dello studente non è stato utilizzato lo stesso procedimento visto in precedenza in quanto il numero di iscritti ogni anno è molto più elevato e diventerebbe più complicato la gestione e il riempimento di una ComboBox, soprattutto perché gli studenti vengono generalmente aggiunti o rimossi più frequentemente rispetto ai tutor.

Saranno presenti due TextField nei quali inserire rispettivamente il nome ed il cognome dello studente, per ridurre la possibilità di errori nell'inserimento dati, le due stringhe ottenute verranno elaborate per rispettare alcuni criteri di uniformità utili per la ricerca nel database e per la stampa nei resoconti.

Per selezionare la data richiesta vengono utilizzati i DatePicker che permettono di visualizzare un calendario in cui selezionare la data richiesta.

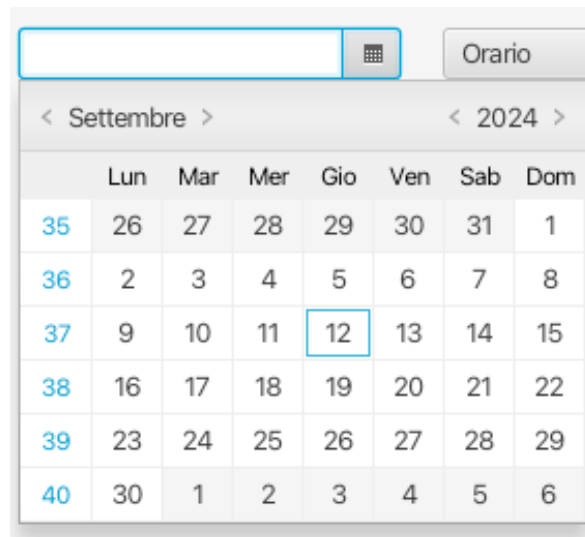


Figura 8: DatePicker

Per la selezione dell'orario viene utilizzata una ComboBox riempita con gli orari disponibili per le lezioni ovvero: 9-11, 11-13, 15-17 e 17-19.

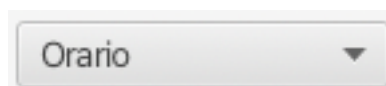


Figura 9: ComboBox Orario

In ogni tab è presente una TextArea che permette di stampare gli output richiesti e un bottone 'RESET' che consente di cancellare gli output e gli elementi selezionati attualmente al fine di poter iniziare una nuova prenotazione.

5.2 Descrizione interfaccia grafica

5.2.1 Gestisci Lezioni

La sezione per la gestione delle lezioni è suddivisa in tre Tab: Lezioni Singole, Lezioni Multiple e Lezioni Giornaliere.

Gestisci Lezioni | Gestisci Studente | Gestisci Tutor | Resoconto | ?

Lezioni Singole | Lezioni Multiple | Lezioni Giornaliere

Prenota Lezioni Singole:

Nome Studente | Cognome Studente

Selezionare Data | Orario

Seleziona Tutor | CALENDARIO

Seleziona Materia

VERIFICA DISPONIBILITA'

PRENOTA | CANCELLA

RESET

Figura 10: Lezioni Singole

Le funzionalità presenti in questa Tab sono:

- **CALENDARIO:** Permette di stampare le lezioni già prenotate per il tutor selezionato e le disponibilità che il tutor garantisce dalla data attuale, in modo da poter visualizzare il calendario e poter fissare le date delle lezioni future per studenti con cui ha un percorso fisso.
- **VERIFICA DISPONIBILITA':** Restituisce come output un elenco dei tutor disponibili per il giorno e orario selezionato con il rispettivo punteggio in caso di assegnazione della lezione.
- **PRENOTA & CANCELLA:** Permette di aggiungere o rimuovere la lezione indicata.

Figura 11: Lezioni Multiple

- **AGGIUNGI & CANCELLA:** Permette di aggiungere in maniera provvisoria una lezione senza selezionare il tutor a cui affidarla, da questo elenco sarà anche possibile rimuovere alcune lezioni tramite il bottone cancella.
- **CALCOLA LEZIONI MULTIPLE:** Restituisce in output l'elenco delle lezioni che sono state aggiunte in maniera provvisoria, con i propri tutor abbinati in modalità automatica, ottenendo il miglior punteggio possibile.
- **PRENOTA:** Se l'elenco delle lezioni soddisfa l'utente sarà possibile prenotarle in maniera definitiva grazie a questo bottone.

Figura 12: Lezioni Giornaliere

- **VISUALIZZA:** Avendo selezionato la data, stampa come output tutte le lezioni prenotate e riempie la ComboBox.
- **FISSA LEZIONE:** Permette di fissare la lezione selezionata tramite il menù a tendina. Le lezioni fissate non verranno calcolate tramite l'algoritmo ricorsivo ma rimarranno uguali.
- **CALCOLA LEZIONI GIORNALIERE:** Calcola in maniera automatica i migliori abbinamenti per le lezioni selezionate e le stampa come output nella TextArea laterale.
- **PRENOTA:** Permette di prenotare in maniera definitiva tutte le lezioni aggiunte grazie al calcolo automatico delle lezioni.

5.2.2 Gestisci Studenti

In questa Tab sarà possibile aggiungere, modificare e gestire le cartelle degli studenti in cui sono indicate le generalità e la classe frequentata.

Gestisci Lezioni | **Gestisci Studente** | Gestisci Tutor | Resoconto | ?

Aggiungi/Aggiorna Studenti:

Nome Studente Cognome Studente

Classe

Studente DSA: ☐ Si ☐ No

Stampa Info:

Figura 13: Gestisci Studente

- **AGGIORNA:** Permette di aggiornare le informazioni dello studente indicato, tramite nome e cognome, come la classe frequentata e se è in possesso di un certificato DSA o BES.
- **AGGIUNGI STUDENTE & RIMUOVI:** Viene aggiunto o rimosso dal database lo studente indicato.
- **STAMPA INFO:** Stamperà come output i dettagli delle lezioni prenotate nel mese attuale e le lezioni prenotate in seguito alla data attuale.

5.2.3 Gestisci Tutor

The screenshot shows the 'Gestisci Tutor' interface. It features a top navigation bar with tabs: 'Gestisci Lezioni', 'Gestisci Studente', 'Gestisci Tutor' (active), 'Resoconto', and '?'. Below this is a sub-navigation bar with 'Aggiorna Tutor' and 'Aggiungi Tutor'. The main content area includes a 'Seleziona Tutor' dropdown. Under the heading 'Aggiungi/Rimuovi Materie:', there is a 'Seleziona Materia' dropdown, 'AGGIORNA', and 'RIMUOVI' buttons. Under 'Aggiungi/Rimuovi Disponibilità:', there is a 'Selezionare Data' field with a calendar icon, an 'Orario' dropdown, 'AGGIUNGI', and 'RIMUOVI' buttons. At the bottom, there is a 'Stampa:' label, a 'STAMPA INFO' button, a 'RESET' button, and a 'RIMUOVI TUTOR' button. A large empty rectangular box is on the right side of the interface.

Figura 14: Aggiorna Tutor

- **AGGIORNA & RIMUOVI:** Selezionando un tutor e una materia dai rispettivi menù a tendina sarà possibile aggiungerla o rimuoverla dall'elenco delle materie che garantisce di poter seguire.
- **AGGIUNGI & RIMUOVI:** Permette di aggiungere o rimuovere una disponibilità per il giorno e l'orario selezionato.
- **STAMPA INFO:** Stampa le informazioni relative al tutor indicato nella ComboBox.

Nella sezione 'Aggiungi Tutor' sono presenti i campi relativi all'inserimento delle informazioni necessarie alla registrazione e i seguenti bottoni:

- **AGGIUNGI MATERIA & RIMUOVI MATERIA:** Per registrare un nuovo tutor sarà fondamentale indicare un elenco di materie che garantisce di poter seguire.
- **AGGIUNGI NUOVO TUTOR:** Dopo aver controllato se il tutor non fosse già inserito nel database, questo bottone permette la corretta registrazione.

5.2.4 Resoconto

The screenshot shows a web application interface with a navigation bar at the top containing the following tabs: 'Gestisci Lezioni', 'Gestisci Studente', 'Gestisci Tutor', 'Resoconto' (which is the active tab), and a question mark icon. The main content area is divided into several sections:

- Resoconto Tutor :** This section includes a 'Seleziona Mese' dropdown menu at the top right. Below it is a 'Seleziona Tutor' dropdown menu. At the bottom of this section are two buttons: 'RESOCONTO TUTOR' and 'LEZIONI PRENOTATE'.
- Resoconto Studente :** This section contains two input fields for 'Nome Studente' and 'Cognome Studente'. Below these fields is a button labeled 'RESOCONTO STUDENTE'.
- Resoconto Mensile :** This section features a button labeled 'TUTTI I TUTOR'.
- Stampa Lezioni Prenotate :** This section includes a button labeled 'LEZIONI PRENOTATE'.

At the bottom center of the interface is a 'RESET' button. A large, empty rectangular box occupies the right side of the main content area, likely intended for displaying the results of the selected actions.

Figura 15: Resoconto

- **RESOCONTO TUTOR:** Stampa le lezioni svolte e prenotate per il tutor nel mese selezionato grazie alla ComboBox.
- **LEZIONI PRENOTATE:** Stampa le lezioni che sono prenotate dalla data odierna per il tutor selezionato.
- **RESOCONTO STUDENTE:** Restituisce in output un elenco delle lezioni svolte e prenotate dallo studente nel mese selezionato.
- **TUTTI I TUTOR:** Permette di ottenere la lista delle lezioni svolte nel mese divise per tutor dove viene anche indicato il totale delle ore svolte da ciascun dipendente.
- **LEZIONI PRENOTATE:** Stampa un elenco delle lezioni prenotate dalla data odierna, questo metodo è pensato per la comunicazione dei calendari ai lavoratori occasionali.

5.3 Output grafico

- Lezioni Singole - CALENDARIO

Gestisci Lezioni Gestisci Studente Gestisci Tutor Resoconto ?

Lezioni Singole Lezioni Multiple Lezioni Giornaliere

Prenota Lezioni Singole:

Nome Studente Cognome Studente

Selezionare Data

Roberto Petrasso

CALENDARIO

VERIFICA DISPONIBILITA'

PRENOTA CANCELLA

RESET

RESOCONTO LEZIONI PRENOTATE TUTOR

Data	Studente	Materia	Orario
2024-10-02	Riccardo Dentis	Matematica+	17-19
2024-10-03	Riccardo Dentis	Matematica+	17-19

RESOCONTO DISPONIBILITA' TUTOR

Data	Orario
2024-10-01	17-19
2024-10-02	17-19
2024-10-03	17-19
2024-10-04	17-19

Figura 16: Calendario

- Lezioni Singole - VERIFICA DISPONIBILITA'

Gestisci Lezioni Gestisci Studente Gestisci Tutor Resoconto ?

Lezioni Singole Lezioni Multiple Lezioni Giornaliere

Prenota Lezioni Singole:

Simone Poggio

04/10/2024

17-19

Seleziona Tutor

CALENDARIO

Fisica

VERIFICA DISPONIBILITA'

PRENOTA CANCELLA

RESET

ELENCO TUTOR DISPONIBILI:

Tutor: Manuel Gajoni PUNTI: 4

Tutor: Mario Rossi PUNTI: 3

Tutor: Roberto Petrasso PUNTI: 2

Tutor: Ester Bottazzi PUNTI: 0

ELENCO LEZIONI GIA' PRENOTATE:

TUTOR: Ester Bottazzi

LEZIONE: IdLezione:32, IdStudente:10, IdTutor:13, Data:2024-10-04, Orario:17-19, Materia:Compiti

Figura 17: Verifica Disponibilità

- Lezioni Multiple - CALCOLA LEZIONI MULTIPLE

Gestisci Lezioni | Gestisci Studente | Gestisci Tutor | Resoconto | ?

Lezioni Singole | **Lezioni Multiple** | Lezioni Giornaliere

Prenota Lezioni Multiple:

Samuele Tono

03/10/2024 17-19

Inglese

AGGIUNGI CANCELLA

CALCOLA LEZIONI MULTIPLE PRENOTA

RESET

IdLezione:-1, IdStudente:7, IdTutor:1, Data:2024-10-03, Orario:15-17, Materia:Geografia
 IdLezione:-1, IdStudente:7, IdTutor:1, Data:2024-10-04, Orario:15-17, Materia:Fisica+
 IdLezione:-1, IdStudente:9, IdTutor:1, Data:2024-10-03, Orario:17-19, Materia:Inglese
 LEZIONI CON IL PUNTEGGIO DI: 7

Figura 18: Calcola lezioni multiple

- Resoconto - TUTTI I TUTOR

Ottobre

Tutor: Ester Bottazzi			
Data	Studente	Materia	Orario
2024-10-04	Alessandro Antonucci	Compiti	17-19
ORE TOTALI:		2	
Tutor: Roberto Petrasso			
Data	Studente	Materia	Orario
2024-10-02	Riccardo Dentis	Matematica+	17-19
2024-10-03	Riccardo Dentis	Matematica+	17-19
ORE TOTALI:		4	

Figura 19: Resoconto tutor

- **CALCOLA LEZIONI GIORNALIERE**

Se sarà stata trovata una soluzione migliore di quella attuale l'output sarà:

The screenshot shows the 'Calcola Lezioni Giornaliere' interface. The date is set to 01/10/2024. The 'VISUALIZZA' button is highlighted. The output panel on the right displays the following information:

TROVATA UNA SOLUZIONE MIGLIORE!
 IdLezione:25, IdStudente:8, IdTutor:1, Data:2024-10-01, Orario:15-17, Materia: Matematica
 IdLezione:34, IdStudente:11, IdTutor:12, Data:2024-10-01, Orario:15-17, Materia: Chimica
 IdLezione:27, IdStudente:10, IdTutor:13, Data:2024-10-01, Orario:17-19, Materia: Matematica
 IdLezione:33, IdStudente:7, IdTutor:1, Data:2024-10-01, Orario:17-19, Materia: Fisica
 PUNTEGGIO LEZIONI PRECEDENTI: 3
 PUNTEGGIO LEZIONI CALCOLATE: 9

Figura 20: Output Calcola lezioni giornaliere: Abbinamenti migliori

The screenshot shows the 'Calcola Lezioni Giornaliere' interface. The date is set to 02/10/2024. The 'VISUALIZZA' button is highlighted. The output panel on the right displays the following information:

LE SOLUZIONI CERCATE SONO EQUIVALENTI A QUELLA GIA' UTILIZZATA!
LEZIONI PRENOTATE:
 IdLezione:28, IdStudente:7, IdTutor:1, Data:2024-10-02, Orario:15-17, Materia:Fisica
 IdLezione:26, IdStudente:8, IdTutor:14, Data:2024-10-02, Orario:17-19, Materia:Matematica+
LEZIONI CALCOLATE:
 IdLezione:28, IdStudente:7, IdTutor:1, Data:2024-10-02, Orario:15-17, Materia:Fisica
 IdLezione:26, IdStudente:8, IdTutor:14, Data:2024-10-02, Orario:17-19, Materia:Matematica+
 PUNTEGGIO LEZIONI PRECEDENTI: 6
 PUNTEGGIO LEZIONI CALCOLATE: 6
 PREMERE IL PULSANTE 'PRENOTA' PER PRENOTARE I NUOVI ABBINAMENTI

Figura 21: Output Calcola lezioni giornaliere: Nessun miglioramento

6. RISULTATI OTTENUTI E CONSIDERAZIONI FINALI

6.1 Raccolta dati

Per la creazione del Data-Set è stato reso disponibile il foglio di calcolo Excel contenente tutte le lezioni prenotate dal mese di settembre 2023 alla prima settimana di agosto 2024, dal quale, è stato possibile ricavare il numero di lezioni e i vari abbinamenti adottati dall'associazione. Tramite le cartelline degli studenti è stato possibile ricavare le informazioni utili al completamento dei rispettivi profili (scuola frequentata, presenza di certificato DSA o BES,...). In accordo con l'associazione è stato deciso di sostituire i nominativi e di non inserire dati sensibili quali data di nascita e scuola frequentata, per ragioni di privacy.

Per la compilazione delle sezioni relative ai tutor, è stato consegnato ad ognuno di loro un questionario nel quale indicare quali tra le materie indicate sarebbero stato in grado di garantire lo svolgimento di una lezione per studenti delle scuole superiori o università.

Le disponibilità dei tutor sono state dedotte dal calendario delle lezioni, così come le materie portate. Quest'ultime non sono sempre state segnalate correttamente sul foglio Excel in quanto è frequente che gli studenti portino più di una materia o che cambino le richieste poco prima della lezione stessa, per questo motivo è stata introdotta la categoria 'Compiti' per indicare le materie generiche che non rappresentano un ostacolo per i tutor. In caso di materie specifiche o che potessero rappresentare una difficoltà nell'assegnazione, sono sempre state segnalate correttamente.

Il principale obiettivo del programma era quello di ridurre il tempo per il calcolo dei calendari ed eliminare gli errori negli abbinamenti dovuti alla distrazione e alle poche informazioni comunicate nel calendario. Per questo motivo, lo studio è stato concentrato maggiormente su tre aspetti: la riduzione o eliminazione delle lezioni prenotate con punteggio zero, l'incremento della media dei punteggi delle lezioni per migliorare la qualità del servizio offerto e l'aumento delle lezioni assegnate agli interni al fine di incrementare i ricavi.

6.2 Dati ottenuti

Per ricavare i dati necessari per la simulazione sono stati calcolati in un primo momento i punteggi e le informazioni relative al caso in esame, successivamente, tramite l'utilizzo del programma, sono stati calcolati in modalità automatica gli abbinamenti per le stesse date ed infine sono stati raccolti i dati necessari alla comparazione dei due metodi.

6.2.1 Lezioni non prenotabili

Analizzando i dati reali relativi all'anno accademico 2023/24, si può notare come siano state prenotate 45 lezioni che hanno totalizzato un punteggio pari a zero. Questo numero risulta allarmante, dato il fatto che solo studenti delle scuole superiori o universitari possano portare all'assegnazione di questo punteggio. Il primo obiettivo era dunque quello di diminuire drasticamente questo numero, tramite la riorganizzazione dei calendari nei giorni presi in esame. Il risultato ottenuto è il seguente:

	SETTEMBRE	OTTOBRE	NOVEMBRE	DICEMBRE
PRIMA	1	5	6	3
DOPO	0	1	1	1

Tabella 16: Numero lezioni con punteggio 0 (Anno 2023)

	GENNAIO	FEBBRAIO	MARZO	APRILE	MAGGIO	GIUGNO	LUGLIO	AGOSTO
PRIMA	3	3	8	1	10	1	4	0
DOPO	1	1	4	0	1	0	4	0

Tabella 17: Numero lezioni con punteggio 0 (Anno 2024)

Sono stati raggiunti dei buoni risultati tramite la simulazione portando il numero di lezioni con punteggio zero da 45 a 14, migliorando così la qualità del servizio grazie alla riorganizzazione delle lezioni assegnate.

Di queste lezioni rimaste si può notare che:

- Cinque di esse sono impossibili da prenotare in quanto è stata richiesta una materia che nessun tutor sarebbe stato in grado di seguire (Matematica Finanziaria), per sopperire a questo problema è stato assegnato il tutor più affine alla materia.
- Cinque lezioni hanno ottenuto un punteggio nullo perché i tutor che avevano le capacità per seguirle erano già stati assegnati ad altre lezioni oppure non avevano garantito la disponibilità. In questo caso la soluzione migliore sarebbe stata quella di spostare le lezioni in una data consona per i tutor in questione.
- Le restanti quattro hanno come oggetto una materia teorica che può essere seguita per eseguire schemi o per lo studio individuale ma non per la parte più dettagliata, ad esempio Storia dell'Arte.

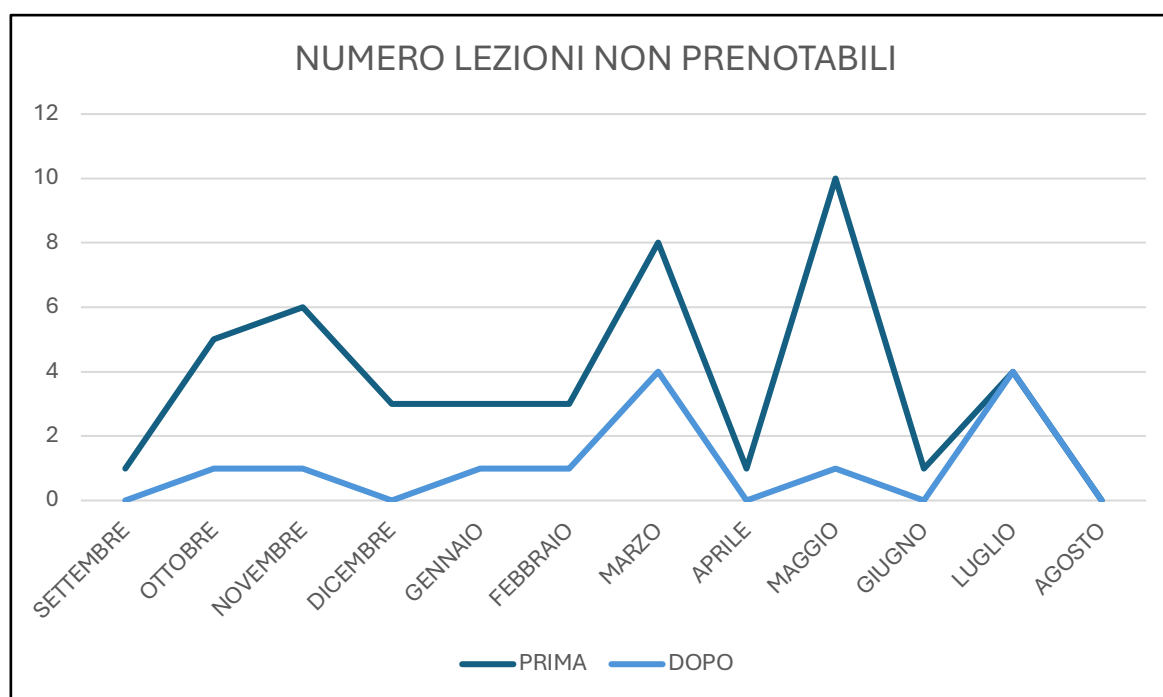


Tabella 18: Numero di lezioni non prenotabili (A.A. 2023/24)

Una possibile soluzione per ridurre ulteriormente il numero di lezioni con punteggio zero è quella di assicurarsi con studenti e tutor che la materia richiesta possa essere effettivamente seguita prima di prenotare un pacchetto di lezioni e di riprogrammare le lezioni di materie specifiche in cui c'è un ristretto numero di tutor quando è presente disponibilità.

6.2.2 Media punteggio mensile

Per migliorare il servizio offerto è necessario offrire i migliori abbinamenti possibili per gli studenti, al fine di garantire anche una continuità nel percorso didattico. Tramite l'assegnazione automatica delle lezioni sono stati possibili ricavare i seguenti dati:

	GENNAIO	FEBBRAIO	MARZO	APRILE	MAGGIO	GIUGNO	LUGLIO
PRIMA	743	879	951	1000	1309	351	491
DOPO	789	900	977	1013	1345	365	493

Tabella 19: Punteggio mensile (Anno 2024)

	GENNAIO	FEBBRAIO	MARZO	APRILE	MAGGIO	GIUGNO	LUGLIO
PRIMA	2,52	2,77	2,57	2,58	2,69	2,76	2,77
DOPO	2,68	2,84	2,64	2,61	2,76	2,87	2,79
%	+6,2 %	+2,4 %	+2,7 %	+1,3 %	+2,8 %	+4,0 %	+0,4 %

Tabella 20: Media mensile (Anno 2024)

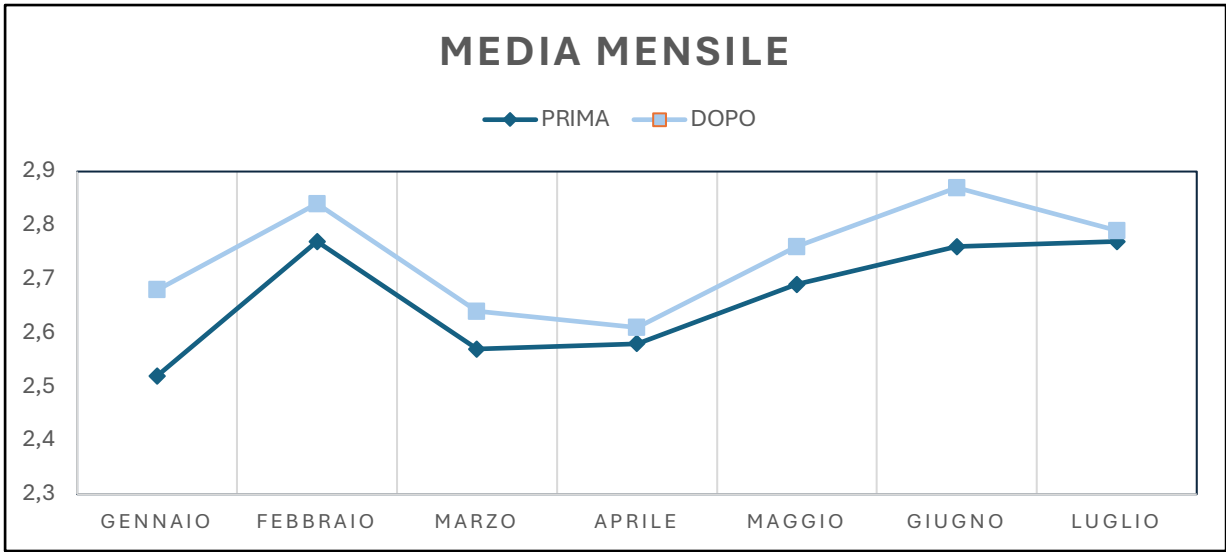


Tabella 21: Media mensile (Anno 2024)

6.2.3 Numero di educatori interni

Dopo aver analizzato due fattori utili per la qualità del servizio offerto, l'ultimo obiettivo era quello di poter portare un guadagno economico all'associazione. Grazie al programma si è verificato un leggero incremento delle lezioni sostenute da interni, secondo i seguenti dati:

PRIMA			DOPO		
TOTALE	INTERNI	ESTERNI	TOTALE	INTERNI	ESTERNI
3129	1641	1488	3129	1646	1483
	52 %	48 %		53 %	47 %

Tabella 22: Distribuzioni lezioni-tutor

I dati ottenuti sono molto simili a quelli ricavati dopo lo studio iniziale in quanto, attualmente, le lezioni vengono prima assegnate ai tutor interni e poi successivamente a quelli esterni.

6.3 Considerazioni finali

Nel complesso i dati ottenuti mediante questa simulazione sono soddisfacenti, in quanto sono stati migliorati tutti gli aspetti che erano stati prefissati nonostante non ci sia stata la possibilità di intervenire attivamente sull'organizzazione dell'associazione. Infatti, sia il punteggio medio per le lezioni e il numero di tutor interni assegnati sono aumentati e il numero di lezioni in cui non si sarebbe garantita una preparazione adeguata sono drasticamente diminuite. Tuttavia, questo programma non può essere utilizzato per il calcolo in automatico di tutta la giornata lavorativa visto che l'algoritmo ricorsivo richiede un'elevata complessità nel calcolo e quindi, richiederebbe un tempo troppo elevato per la soluzione finale. Questo metodo può essere funzionale per il calcolo di 4/5 lezioni per ogni fascia oraria, perché viene restituita la soluzione ottimale in meno di un minuto. Nonostante i tentativi di suddividere la ricerca in fasce orarie e di calcolare solo gli abbinamenti con tutor plausibili, non si è riuscito a raggiungere un risultato ottimale per il calcolo di tutta la settimana lavorativa nella sua interezza, ma, questo programma può essere un sostegno per il calcolo degli abbinamenti delle lezioni con studenti che non seguono un percorso didattico fisso con un educatore specifico. Il suo utilizzo può, inoltre, essere utile per ridurre al minimo gli errori e per garantire uno standard elevato anche con più operatori impiegati nella segreteria.

In un futuro, l'obiettivo da raggiungere sarà quello di garantire un'automazione totale del processo di assegnazione delle lezioni grazie a questo programma e alla possibilità di gestire in autonomia la comunicazione con le famiglie degli studenti e con i tutor per la comunicazione degli orari.

Indice delle figure

Figura 1: Tabella Studente	14
Figura 2: Tabella Tutor	14
Figura 3: Tabella Disponibilità	15
Figura 4: Tabella Lezioni	15
Figura 5: Diagramma Entità-Relazione	16
Figura 6: ComboBox Materie	38
Figura 7: ComboBox Tutor	38
Figura 8: DatePicker	39
Figura 9: ComboBox Orario	39
Figura 10: Lezioni Singole	40
Figura 11: Lezioni Multiple	41
Figura 12: Lezioni Giornaliere	42
Figura 13: Gestisci Studente	43
Figura 14: Aggiorna Tutor	44
Figura 15: Resoconto	45
Figura 16: Calendario	46
Figura 17: Verifica Disponibilità	46
Figura 18: Calcola lezioni multiple	47
Figura 19: Resoconto tutor	47
Figura 20: Output Calcola lezioni giornaliere: Abbinamenti migliori	48
Figura 21: Output Calcola lezioni giornaliere: Nessun miglioramento	48

Questa relazione tecnica è rilasciata con licenza Creative Commons BY-NC-ND.

Tu sei libero di:

- **Condividere** — riprodurre, distribuire, comunicare al pubblico, esporre in pubblico, rappresentare, eseguire e recitare questo materiale con qualsiasi mezzo e formato

Il licenziante non può revocare questi diritti fintanto che tu rispetti i termini della licenza.

Alle seguenti condizioni:

- **Attribuzione** — Devi riconoscere una menzione di paternità adeguata , fornire un link alla licenza e indicare se sono state effettuate delle modifiche . Puoi fare ciò in qualsiasi maniera ragionevole possibile, ma non con modalità tali da suggerire che il licenziante avalli te o il tuo utilizzo del materiale.
- **Non Commerciale** — Non puoi utilizzare il materiale per scopi commerciali .
- **Non opere derivate** — Se remixi, trasformi il materiale o ti basi su di esso , non puoi distribuire il materiale così modificato.
- **Divieto di restrizioni aggiuntive** — Non puoi applicare termini legali o misure tecnologiche che impongano ad altri soggetti dei vincoli giuridici su quanto la licenza consente loro di fare.

Per visualizzare una copia di questa licenza, visitare :

<https://creativecommons.org/licenses/by-nc-nd/4.0/deed.it>

