

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

SCUOLA DI SCIENZE

Corso di Laurea in Informatica per il Management

**Pianificazione di eventi
nel rispetto di vincoli hard e soft:
il caso di studio degli esami universitari**

Relatore:
Chiar.mo Prof.
Angelo Di Iorio

Presentata da:
Delia Cavalca

Sessione II
Anno Accademico 2022/2023

*Alla mia mamma e al mio papà,
per ogni loro gesto*

Introduzione

Da sempre le università sono la sede della cultura, della conoscenza, del sapere e della ricerca, svolgendo un ruolo centrale nell'evoluzione dell'innovazione e del pensiero collettivo. Caratterizzate da innumerevoli individui, ruoli, risorse e processi, richiedono una solida struttura organizzativa indispensabile per guidare in modo equilibrato l'intero ecosistema che le contraddistingue.

Fra le tante attività da gestire in un contesto accademico, la pianificazione degli esami universitari è sicuramente una di quelle di primaria importanza. Si tratta di un processo che deve tenere conto di una varietà di fattori, esigenze ed equilibri, cui è fondamentale porre una particolare attenzione per garantire un'esperienza accademica organizzata ed efficiente.

La presente tesi si propone di analizzare le esigenze intrinseche al contesto della programmazione degli esami universitari e introduce una soluzione tecnologica moderna e flessibile, progettata con l'obiettivo di semplificare e migliorare significativamente il processo di pianificazione.

ExamPlan è un'applicazione web pensata per guidare gli utenti nell'organizzazione e nella gestione di eventi, considerando i vincoli delineati dalla realtà di interesse come parte integrante del processo di planning.

Nello specifico, questa nuova piattaforma è concepita per un possibile aggiornamento dell'attuale sistema di programmazione degli esami utilizzato dal Dipartimento di Informatica - Scienza e Ingegneria (DISI) dell'Università di Bologna, non più in grado di soddisfare le necessità in continua evoluzione di studenti e docenti.

La rilevanza di questa tesi risiede nell'opportunità di migliorare la qualità e l'efficienza delle pianificazioni degli esami universitari, fornendo un'applicazione che sia allineata con le moderne esigenze degli utenti. Il presente studio si propone inoltre di valutare un'astrazione del sistema per un suo futuro adattamento a contesti differenti, per sfruttare l'abilità di integrazione di vincoli per un'evoluzione della pianificazione di eventi in diversi scenari.

Nel prosieguo di questa tesi, verranno esaminati i dettagli dell'implementazione dell'app ExamPlan, i suoi vantaggi potenziali e il modo in cui potrebbe contribuire a rinnovare il processo di pianificazione degli esami nell'Università di Bologna.

Indice

Introduzione	i
1 Analisi preliminare	1
1.1 Obiettivi	1
1.2 L'attuale sistema	3
1.3 Software esistenti	5
1.4 Motivazioni	10
2 Dal problema alla soluzione	13
2.1 Introduzione a ExamPlan	13
2.2 Requisiti e terminologia	14
2.2.1 Utenti	14
2.2.2 Esami	15
2.2.3 Vincoli	16
2.2.4 Anno Accademico	16
2.3 Funzionalità	17
2.3.1 Come si presenta ExamPlan	17
2.3.2 Programmazione di un nuovo esame	18
2.3.3 Piattaforma di amministrazione	20
2.3.4 Sistema di notifica	22
3 Progettazione	25
3.1 Base di dati	25
3.1.1 Analisi della realtà di interesse	25
3.1.2 Progettazione database	27

3.1.3	Implementazione database	27
3.2	Architettura dell'applicazione	29
3.2.1	Design Pattern MVVM	29
3.2.2	Struttura Client - Server	30
4	Implementazione	33
4.1	Tecnologie	33
4.2	Client Side	34
4.2.1	Il framework Vue.js	34
4.2.2	Componenti principali e meccanismo di autenticazione	35
4.2.3	Meccanismo di navigazione: Vue-Router	36
4.2.4	Interazione col server: Axios e Vue-Axios	37
4.2.5	Gestione dati di sessione tramite LocalStorage: Vuex	37
4.2.6	Il calendario di ExamPlan: Tui-Calendar	38
4.2.7	Implementazione dei vincoli	40
4.3	Server Side	41
4.3.1	Node.js e il framework Express	41
4.3.2	API RESTful per la gestione delle richieste Client	42
4.3.3	Connessione al database MySQL	43
	Valutazione e conclusioni	45
	Bibliografia	51

Elenco delle figure

1.1	Pannello di EasyTest per esprimere preferenze	7
1.2	Pannello di EasyTest per determinare i vincoli	7
2.1	Workflow di pianificazione di un esame	14
2.2	Come si presenta ExamPlan	17
2.3	Esame da confermare	18
2.4	Esame confermato	18
2.5	Fasi della pianificazione di un esame	18
2.6	Programmazione di un nuovo esame con vincoli in ExamPlan	19
2.7	Pannello di amministrazione di ExamPlan	20
2.8	Conferma esame e prenota aule	21
2.9	Pannello delle notifiche di ExamPlan	23
3.1	Modello dei dati dell'applicazione	27
3.2	Design Pattern MVVM	29
3.3	Interazione Client-Server	30
4.1	Architettura e tecnologie di ExamPlan	33
4.2	Schema delle View e delle Componenti di ExamPlan	35
4.3	Schema delle API di ExamPlan	43

Capitolo 1

Analisi preliminare

Per comprendere chiaramente gli obiettivi specifici del progetto è necessaria una fase di analisi delle esigenze nell'ambito della pianificazione e calendarizzazione degli esami universitari, nonché delle problematiche legate all'uso del sistema attuale del DISI dell'Università di Bologna nello specifico e dei software esistenti potenzialmente adottabili dalla stessa.

1.1 Obiettivi

Ogni ruolo coinvolto nella vita accademica, dai docenti agli uffici amministrativi, dagli studenti ai coordinatori, ha esigenze specifiche che devono essere prese in considerazione.

In un anno accademico sono attivi determinati corsi di studi cui gli studenti possono iscriversi, ogni corso prevede un insieme di insegnamenti (ognuno in uno specifico anno del corso) e di ciascuno dovranno essere programmati i relativi appelli d'esame.

La redazione di un piano d'esami universitario è un processo intricato in quanto deve tenere conto di una varietà di fattori, come la disponibilità dei docenti, le esigenze degli studenti e l'ottimizzazione delle risorse. Emerge per i docenti la necessità di creare un piano che armonizzi la sequenza degli esami, che sia coerente rispetto a quelli già stabiliti ed evitando possibili conflitti.

Uno dei principali vincoli che è necessario tenere in considerazione, riguarda l'impossibilità di programmare esami di insegnamenti di stesso corso di studi e stesso anno nella medesima giornata. Questo per non creare sovrapposizioni che sicuramente genererebbero disagio e difficoltà nell'organizzazione personale degli studenti.

Altro caso cui si vuole porre attenzione riguarda la situazione in cui il conflitto nasce da esami di anni differenti di uno stesso corso di studi. Può capitare infatti che gli studenti abbiano esami arretrati e conseguentemente abbiano la necessità di conseguire esami di anni diversi nel medesimo periodo. Questo secondo caso, a differenza del primo, non deve essere vincolante nella scelta della data in cui pianificare il nuovo appello ma solamente un indicatore di buona pratica per l'utente che sta programmando l'esame.

Per una pianificazione lineare e ottimale, è opportuno che gli appelli d'esame di uno stesso insegnamento siano correttamente distanziati fra loro.

Inoltre, analizzando accuratamente il contesto universitario, esistono insegnamenti particolari detti "mutuati". Si tratta di insegnamenti tenuti congiuntamente in più corsi di studi da un unico docente (con un unico orario di lezioni cui partecipano insieme gli studenti dei diversi corsi). Per un tale insegnamento mutuato, la calendarizzazione diviene ulteriormente complessa in quanto deve tenere conto di esigenze e sovrapposizioni con altri esami rispetto a tutti i vari corsi in cui è presente tale insegnamento.

Infine, tenendo conto delle necessità e degli impegni dei docenti, è bene che nel calendario accademico non vi siano sovrapposizioni di eventi di uno stesso professore.

Le varie esigenze intrinseche alla realtà accademica presentate riflettono un grado di rilevanza differente, secondo il quale una potrebbe prevalere sull'altra o la cui considerazione potrebbe essere necessaria o non durante un processo di pianificazione. Per questo, è fondamentale prevedere a livello di sistema una gestione dei vincoli in base a una loro classificazione di importanza.

Contestualmente alla pianificazione di nuovi esami da parte dei docenti, la segreteria didattica ha il compito di gestire, monitorare ed approvare gli stessi. Di fondamentale importanza è l'assegnazione delle aule ai singoli appelli, tenendo conto della disponibilità

e della capienza delle aule e delle esigenze del relativo esame.

L'obiettivo centrale è dunque la realizzazione di un prototipo che considera determinati vincoli come guida per la pianificazione degli esami universitari, eventualmente estendibile ed adottabile in ulteriori scenari.

Attualmente si prospetta una definizione dei vincoli a livello di sistema, senza l'aggiunta di particolari preferenze ed esigenze espresse dai singoli utenti da utilizzare per una eventuale programmazione automatizzata. I vincoli preimpostati hanno lo scopo di suggerire a docenti e segreteria indicazioni per una calendarizzazione ottimale, ma la pianificazione è completamente gestita e manipolabile dagli stessi.

Si tratta di un'attività di refactoring e porting dell'attuale sistema utilizzato dal Dipartimento di Informatica dell'UniBO, con un'evoluzione del processo di programmazione di eventi per un utilizzo maggiormente intuitivo e immediato.

1.2 L'attuale sistema

PrenEsami è la piattaforma attualmente utilizzata dai docenti del Dipartimento di Informatica dell'UniBO per la calendarizzazione degli esami. Tale software mostra alcuni punti deboli e può essere ulteriormente migliorato.

Caratterizzato da un'interfaccia molto semplice e basilare, consente di pianificare un nuovo evento tramite un processo caratterizzato da due fasi, durante le quali l'utente sceglie le opzioni dell'esame tramite una serie di menù a tendina e campi di testo.

Prima fase:

1. scelta della tipologia di esame (scritto, orale, prova di laboratorio, ecc)
2. scelta del corso di laurea
3. scelta dell'insegnamento (fra quelli di cui il docente è titolare)

Seconda fase:

1. scelta della data
2. scelta dell'ora di inizio e della durata dell'appello
3. selezione del numero di aule richieste e relativa capienza
4. inserimento di eventuali note per operatore e studenti

Durante la seconda fase, viene mostrato al docente l'elenco delle date già occupate da altri esami di stesso corso di studi e stesso anno, con suddivisione fra date di esami dell'insegnamento selezionato e di insegnamenti differenti. Nonostante ciò, l'utente può comunque selezionare una di tali date per il nuovo appello, solo in seguito viene mostrato un messaggio per indicare l'impossibilità di pianificare in tale giorno e annullata la prenotazione. Inoltre, la visualizzazione stessa di tale elenco di date non ha alcun impatto visivo sulla distribuzione degli esami, limitando la capacità del docente di individuare una data che sia favorevole anche agli studenti.

PrenEsami non implementa nessun meccanismo per il quale vengono considerate esigenze specifiche dei soggetti coinvolti, riducendo la possibilità di ottenere una calendarizzazione ottimale.

Dopo la creazione di una nuova richiesta di esame, un operatore della segreteria didattica deve intervenire per la verifica dei dati inseriti e l'assegnazione delle aule.

Il docente viene notificato tramite mail istituzionale quando la sua richiesta viene approvata e in seguito può effettivamente pubblicare il nuovo appello d'esame agli studenti. Tuttavia, non sono previste mail automatiche nel caso della modifica di un appello d'esame, nonostante sia una situazione in cui è importante informare il docente.

Inoltre, non è presente nella piattaforma un pannello con lo storico completo delle notifiche dell'utente, utile per averne una visione immediata e priva di messaggi di contesti differenti come può essere nella casella di posta.

Attualmente, PrenEsami non è dotato di un pannello di amministrazione per una gestione centralizzata di tutti i dati. Se necessarie, le modifiche vanno apportate manualmente dagli sviluppatori direttamente nel database.

1.3 Software esistenti

Nel prosieguo di questo capitolo, viene condotta un'analisi di mercato per individuare i software esistenti nel panorama della pianificazione di eventi, adottabili per l'organizzazione degli esami universitari.

Tale studio si prefigge l'obiettivo di valutare se già esistono soluzioni che possano adattarsi in maniera soddisfacente alle esigenze specifiche dell'Università di Bologna, fornendo una panoramica critica sulle opportunità e le sfide che tali soluzioni comportano.

Per comprendere al meglio se una nuova soluzione software sia necessaria invece che adottare ed integrare un sistema esistente, è importante evidenziare i vantaggi e la compatibilità in termini di funzionalità dei software individuati rispetto alle esigenze del nostro caso ed è bene analizzare i limiti che derivano dalla possibilità di personalizzarli solo in maniera circoscritta in caso di una loro adozione.

Si elencano di seguito alcuni prodotti oggi presenti sul mercato:

- **EasyAcademy:** [1] sviluppato da Zucchetti S.p.a. per la gestione e l'organizzazione dei calendari didattici, per la gestione delle aule negli Istituti Universitari e per una migliore efficienza organizzativa e ottimizzazione dell'utilizzo complessivo delle risorse didattiche. EasyAcademy è una suite che si compone di diversi moduli algoritmici come EasyCourse, EasyTest e EasyThesis che permettono, attraverso l'uso di motori automatici, di pianificare automaticamente i calendari delle lezioni, degli esami e gestire le commissioni dei docenti nelle sessioni di laurea. Inoltre, EasyAcademy permette la gestione del calendario degli eventi svolti nelle varie aule con le relative statistiche, attraverso i moduli EasyRoom e EasyAnalytics. I moduli della suite EasyAcademy sono integrati tra loro e sono integrabili con sistemi informatici già presenti negli Istituti Universitari al fine di ereditare utili informazioni di base.

L'analisi del sistema viene focalizzata nello specifico sulla gestione del calendario d'esami, tema centrale di questa tesi.

Tramite la piattaforma EasyAcademy, il calendario degli esami di un ateneo viene generato in automatico dal sistema, il quale tiene conto di due fattori: preferenze

espresse dai docenti in merito alle date in cui svolgere o non svolgere i propri esami e vincoli e regole definite dagli amministratori, alle quali è possibile associare anche un certo livello di importanza.

Un docente non può quindi pianificare un singolo appello d'esame scegliendone data e ora specifica, come avviene nel sistema PrenEsami, può esclusivamente esprimere preferenze o impossibilità relative a fasce orarie in cui pianificare i propri esami in una determinata sessione.

Le indisponibilità temporali definite dai professori possono essere generali o specifiche per esame (per uno specifico insegnamento). Inoltre, un determinato docente ha un numero massimo di fasce di indisponibilità da poter scegliere, stabilito in fase di setting dall'amministratore.

Gli amministratori gestiscono i dati della facoltà e la generazione del calendario di una certa sessione, con la possibilità di apportarvi eventuali modifiche manuali.

Anche l'assegnazione di aule e laboratori agli eventi è totalmente gestita in automatico dal sistema.

EasyAcademy offre una visualizzazione delle prenotazioni tramite una griglia, nella quale per ciascuna fascia temporale di una certa giornata e ciascuna aula inserita a sistema, viene associato uno specifico evento. La griglia può fornire una visualizzazione degli eventi differente rispetto ad esigenze specifiche, ad esempio colorando gli esami a seconda della tipologia, del docente titolare o in base al corso di studi.

Il software analizzato è caratterizzato da un'architettura solida ed è allo stesso tempo intuitivo, fornendo una esperienza d'uso gradevole. Permette di ottenere una gestione centralizzata dell'organizzazione accademica in un'unica soluzione informatica, utile per la gestione di tutte le problematiche legate ai calendari didattici. Si tratta quindi di un sistema valido e completo che ha sicuramente innumerevoli vantaggi, ma altrettanto elevato è anche il suo costo.

Vengono riportati di seguito due screenshot di EasyTest ottenuti da un video di presentazione del sistema stesso presente sul canale YouTube del Gruppo Zucchetti [2], per mostrare come si presenta la piattaforma (si precisa che il video è datato).

La prima figura mostra il pannello tramite il quale un docente può esprimere le fasce orarie in cui ha impossibilità di svolgere propri esami. La seconda mostra invece l'interfaccia che si presenta agli amministratori per impostare quali vincoli (con la relativa importanza) deve considerare il sistema durante la generazione automatica del calendario d'esami.

	lunedì	martedì	mercoledì	giovedì	venerdì	sabato	domenica
giugno	<input type="checkbox"/> 18.6.2012	<input type="checkbox"/> 19.6.2012	<input type="checkbox"/> 20.6.2012	<input type="checkbox"/> 21.6.2012	<input type="checkbox"/> 22.6.2012		
	<input checked="" type="checkbox"/> mattina	<input checked="" type="checkbox"/> mattina	<input type="checkbox"/> mattina	<input type="checkbox"/> mattina	<input type="checkbox"/> mattina		
	<input checked="" type="checkbox"/> pomeriggio	<input type="checkbox"/> pomeriggio	<input type="checkbox"/> pomeriggio	<input type="checkbox"/> pomeriggio	<input type="checkbox"/> pomeriggio		
	<input type="checkbox"/> 25.6.2012	<input type="checkbox"/> 26.6.2012	<input type="checkbox"/> 27.6.2012	<input type="checkbox"/> 28.6.2012	<input type="checkbox"/> 29.6.2012		
	<input type="checkbox"/> mattina	<input type="checkbox"/> mattina	<input type="checkbox"/> mattina	<input type="checkbox"/> mattina	<input type="checkbox"/> mattina		
	<input type="checkbox"/> pomeriggio	<input type="checkbox"/> pomeriggio	<input type="checkbox"/> pomeriggio	<input type="checkbox"/> pomeriggio	<input type="checkbox"/> pomeriggio		
luglio	<input type="checkbox"/> 2.7.2012	<input type="checkbox"/> 3.7.2012	<input type="checkbox"/> 4.7.2012	<input type="checkbox"/> 5.7.2012	<input type="checkbox"/> 6.7.2012		
	<input type="checkbox"/> mattina	<input type="checkbox"/> mattina	<input type="checkbox"/> mattina	<input type="checkbox"/> mattina	<input type="checkbox"/> mattina		
	<input type="checkbox"/> pomeriggio	<input type="checkbox"/> pomeriggio	<input type="checkbox"/> pomeriggio	<input type="checkbox"/> pomeriggio	<input type="checkbox"/> pomeriggio		
	<input type="checkbox"/> 9.7.2012	<input type="checkbox"/> 10.7.2012	<input type="checkbox"/> 11.7.2012	<input type="checkbox"/> 12.7.2012	<input type="checkbox"/> 13.7.2012		
	<input type="checkbox"/> mattina	<input type="checkbox"/> mattina	<input type="checkbox"/> mattina	<input type="checkbox"/> mattina	<input type="checkbox"/> mattina		
	<input type="checkbox"/> pomeriggio	<input type="checkbox"/> pomeriggio	<input type="checkbox"/> pomeriggio	<input type="checkbox"/> pomeriggio	<input type="checkbox"/> pomeriggio		
	<input type="checkbox"/> 16.7.2012	<input type="checkbox"/> 17.7.2012	<input type="checkbox"/> 18.7.2012	<input type="checkbox"/> 19.7.2012	<input type="checkbox"/> 20.7.2012		
	<input type="checkbox"/> mattina	<input type="checkbox"/> mattina	<input type="checkbox"/> mattina	<input type="checkbox"/> mattina	<input type="checkbox"/> mattina		
	<input type="checkbox"/> pomeriggio	<input type="checkbox"/> pomeriggio	<input type="checkbox"/> pomeriggio	<input type="checkbox"/> pomeriggio	<input type="checkbox"/> pomeriggio		
	<input type="checkbox"/> 23.7.2012	<input type="checkbox"/> 24.7.2012	<input type="checkbox"/> 25.7.2012	<input type="checkbox"/> 26.7.2012	<input type="checkbox"/> 27.7.2012		
	<input type="checkbox"/> mattina	<input type="checkbox"/> mattina	<input type="checkbox"/> mattina	<input type="checkbox"/> mattina	<input type="checkbox"/> mattina		
	<input type="checkbox"/> pomeriggio	<input type="checkbox"/> pomeriggio	<input type="checkbox"/> pomeriggio	<input type="checkbox"/> pomeriggio	<input type="checkbox"/> pomeriggio		

Figura 1.1: Pannello di EasyTest per esprimere preferenze

Peso voci di costo per aule

Due esami nella stessa aula nella stessa fascia

Esami senza aula

Aule in videoconferenza

Peso voci di costo per docenti

Sovrapposizioni impossibili tra esami dello stesso docente

Sovrapposizioni tra esami con stesso docente che è titolare e in commissione

Sovrapposizioni tra esami con docente in commissione

Fascia impossibile per docente

Fascia preferita per docente

Ordinamento fasce preferite per docente

Figura 1.2: Pannello di EasyTest per determinare i vincoli

- **UniTime:** [3] progetto supportato da Apereo, è un sistema completo per la programmazione di attività didattiche, che supporta la pianificazione temporale di corsi ed esami e la gestione delle modifiche a questi orari. Si tratta di un sistema distribuito che consente alle università di coordinare gli sforzi per costruire e modificare un programma che soddisfi le proprie diverse esigenze organizzative, consentendo al contempo di ridurre al minimo i conflitti fra corsi degli studenti.

La generazione del calendario accademico è automatica, basata su un processo di assegnazione di lezioni o esami a fasce temporali ed aule, considerando il problema legato alla compatibilità fra esigenze differenti di studenti, di disponibilità di spazio e altri requisiti. L'obiettivo è minimizzare i conflitti e ottenere una pianificazione di lezioni ed esami efficiente, limitando e ottimizzando l'uso delle risorse.

Anche in questo caso vengono applicati determinati vincoli per la definizione del piano d'esami, come la dimensione delle aule, la loro disponibilità, preferenze espresse relative a fasce temporali ed esigenze di studenti e docenti. I vincoli sono distinti in base al livello di necessità che siano rispettati.

Il risultato del processo è una tabella temporale degli esami, che ha una visualizzazione adattabile secondo differenti parametri (filtro per fascia temporale o per data, scelta della modalità di visualizzazione e altri).

La piattaforma presenta inoltre un'area amministrativa attraverso la quale è possibile aggiungere e modificare i dati del sistema.

Il sistema è generico, facilmente estendibile e personalizzabile. È open source, con disponibile supporto commerciale in caso di necessità. Può essere utilizzato da solo per creare e mantenere l'orario delle lezioni e/o degli esami di un istituto o integrato in un sistema informativo per gli studenti esistente.

Tuttavia, è un software obsoleto con un'interfaccia poco accattivante e talvolta di difficile comprensione. Le funzionalità che offre non sono del tutto compatibili con quelle richieste dall'Università di Bologna.

- **Series25:** [4] sistema della società statunitense CollegeNet, si tratta di un insieme di tool progettati per la programmazione automatizzata di eventi e corsi accademici e la gestione delle relative risorse.

Uno dei principali obiettivi del software è la gestione degli spazi accademici, tramite una pianificazione dei corsi nelle aule disponibili considerando un'ottimizzazione dell'uso delle stesse.

È possibile pianificare e gestire eventi, e ottenere una visualizzazione degli stessi su calendario. Ogni utente può creare eventi, gestirli, richiedere l'uso di aule e assegnare risorse. Non è prevista una chiara separazione delle funzionalità in ruoli specifici.

Il sistema offre alle università il pieno coordinamento e controllo su tutti i programmi accademici ed eventi, si integra perfettamente con tutti i principali sistemi informativi per studenti.

Non si tratta però di un software specifico per la programmazione di esami universitari ed è completamente assente l'integrazione di determinati vincoli come guida durante la pianificazione di eventi.

Si ritiene quindi ulteriormente complessa una sua personalizzazione al fine di ottenere soddisfatti tutti i requisiti del nostro caso.

- **AscTimetables:** [5] si tratta di un software utile per l'organizzazione di attività didattiche, in grado di generare automaticamente una pianificazione oraria di eventi.

La schedulazione di eventi avviene tramite l'importazione di dati e requisiti nella fase iniziale e l'inserimento di una serie di parametri quali anno accademico, fasce orarie giornaliere, insegnamenti da programmare, aule disponibili da considerare.

Una volta ottenuta la calendarizzazione proposta dalla piattaforma, è possibile apportare modifiche manuali al piano generato. Il sistema fornisce inoltre la possibilità di esportare la pianificazione di un singolo corso, di una singola aula, di un singolo docente o studente.

Il sistema non consente ai docenti di effettuare una propria pianificazione ed esprimere preferenze in merito a quando svolgere gli appelli d'esame. Si tratta di una pianificazione automatizzata che però non tiene in considerazione l'applicazione di vincoli imposti da amministratori e docenti.

Inoltre nel sistema non è previsto un pannello di amministrazione per una gestione centralizzata dei dati della piattaforma.

Per questo, non può essere considerato una soluzione efficiente per la gestione degli esami del DISI dell'Università di Bologna, in quanto processo complesso che necessita dell'inclusione di una serie più estesa di fattori rispetto a quelli considerati da AscTimetables. Inoltre, è un software con interfaccia molto basilare e, in conclusione, non specifico per le nostre esigenze.

1.4 Motivazioni

Nonostante la vasta gamma di software esistenti per la pianificazione di eventi e attività, pochi sistemi presentano caratteristiche e funzionalità in linea con le esigenze dettate dal processo di pianificazione degli esami universitari e in particolare con i requisiti specifici dell'Università di Bologna.

Il software obiettivo deve prevedere la possibilità per i singoli docenti di programmare i propri appelli d'esame in specifiche date, non si sta cercando un sistema che effettua una calendarizzazione automatica sulla base di preferenze, come invece risulta essere nella maggior parte delle piattaforme individuate.

L'elemento chiave che distingue questa situazione è la necessità di gestire eventi con vincoli specifici impostati a livello di sistema, vincoli che tengano conto delle restrizioni relative a docenti, aule, orari e a una molteplicità di fattori variabili.

In questa prospettiva, emerge un aspetto cruciale: la scarsità di soluzioni software esistenti che offrano una pianificazione basata su vincoli altamente personalizzabili e adattabili al contesto universitario in questione, garantendo pieno controllo sulla pianificazione stessa a docenti e segreteria didattica.

Per questo nasce ExamPlan, una soluzione moderna e specifica che integra e migliora questi aspetti. La sfida di progettare e implementare un sistema di pianificazione degli esami universitari che soddisfi appieno le necessità dell'Università di Bologna richiede un approccio su misura.

L'adozione di un sistema preesistente, sebbene possa apparire come un'opzione immediata, potrebbe comportare limitazioni significative. Non si avrebbe infatti un controllo completo sul sistema, potrebbero esservi limitazioni nella personalizzazione e sussistere vincoli tecnici o funzionali non adattabili alle specifiche esigenze.

Scegliere di sviluppare un software proprietario, come ExamPlan, offre ulteriori vantaggi: l'aspetto della scalabilità e dell'estendibilità si rivelano fondamentali. Ciò significa che, con l'evoluzione delle esigenze e dei processi universitari, ExamPlan può essere continuamente aggiornato e personalizzato senza le restrizioni che potrebbero derivare dall'adozione di un prodotto esterno.

Capitolo 2

Dal problema alla soluzione

ExamPlan è il risultato dello studio delle esigenze di docenti, studenti e segreteria didattica in continua evoluzione. L'ottimizzazione dei tempi e l'uso immediato degli strumenti è oggi un obiettivo fondamentale.

2.1 Introduzione a ExamPlan

ExamPlan è concepita per ottenere una pianificazione degli esami ottimale per ogni ruolo presente nella realtà accademica. Il sistema si propone di ridurre significativamente il tempo impiegato dai docenti nell'organizzazione dei propri esami, offrendo loro uno strumento intuitivo che consenta di gestire facilmente le pianificazioni.

ExamPlan adotta un approccio di classificazione dei vincoli delineati dalla realtà accademica in base al loro livello di importanza, per distinguere se si tratta di vincoli obbligatori o non, esprimendo con chiarezza in quali giornate sia opportuno programmare nuovi esami di determinati insegnamenti. Un calendario visivo permette a docenti e segreteria didattica di avere una visione immediata e comprensibile dell'intera programmazione degli appelli d'esame, garantendo un'esperienza d'uso piacevole e consentendo una gestione più accurata dei tempi.

Gli operatori della segreteria hanno inoltre la possibilità di gestire in modo centralizzato i dati della piattaforma, contribuendo così a una manipolazione globale delle risorse accademiche più efficace.

Così come nell'attuale sistema PrenEsami, il processo di pianificazione di un nuovo appello d'esame prevede due fasi principali: una prima fase di inserimento dei dati e il successivo intervento della segreteria didattica per l'assegnazione degli spazi e conferma dell'evento.

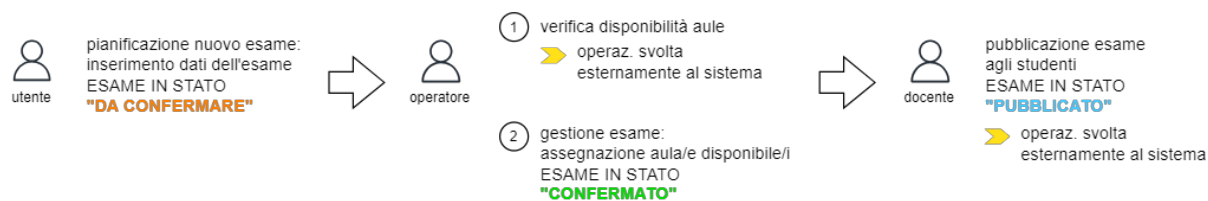


Figura 2.1: Workflow di pianificazione di un esame

2.2 Requisiti e terminologia

Sulla base degli obiettivi sono stati definiti i requisiti dell'applicazione. Rispetto alla precedente implementazione, è stato integrato un pannello di amministrazione per una gestione centralizzata di tutti i dati ed è stato implementato un sistema di notifica completo, aggiungendo la visibilità dello storico di notifiche sulla piattaforma stessa.

2.2.1 Utenti

La piattaforma prevede due tipologie di utenti: “docente” e “operatore”. Per “operatore” si intende un membro della segreteria didattica incaricato di gestire la pianificazione degli esami.

Il **docente** può effettuare le seguenti azioni:

- visualizzare gli esami programmati tramite calendario
- creare un nuovo appello d'esame
- modificare o eliminare un proprio appello d'esame

- modificare i propri dati personali e le impostazioni relative alle notifiche

L'**operatore** può effettuare le seguenti azioni:

- visualizzare e operare su esami come per i docenti
- confermare un esame: assegnazione aule all'appello d'esame
- amministrazione dati: aggiungere/modificare/cancellare i dati coinvolti
- impostazione della piattaforma docenti: compilazione messaggio di info
- impostazione dei vincoli
- modificare i propri dati personali e le impostazioni relative alle notifiche

2.2.2 Esami

Gli esami sono l'oggetto delle pianificazioni effettuate dagli utenti. Come mostrato precedentemente nel workflow di una pianificazione, nel momento in cui viene creato un appello d'esame tale evento è in stato "**da confermare**". Infatti, il nuovo appello è caratterizzato da una serie di dati che devono essere verificati e necessita dell'intervento di un operatore per la sua validazione ed assegnazione di una o più aule in base alle esigenze espresse dal docente che lo ha creato. Solo nel momento in cui l'operatore gestisce e conferma l'esame, quest'ultimo passa in stato "**confermato**". Un esame confermato non è ancora visibile agli studenti, la pubblicazione ufficiale dell'appello avviene tramite un'operazione effettuata esternamente alla piattaforma.

Un esame può essere modificato o eliminato in un qualsiasi momento. Se si modificano i dati di un appello che era già stato confermato, l'aula assegnata verrà rilasciata e sarà necessario un nuovo intervento da parte di un operatore per la verifica e la prenotazione di una nuova aula (l'esame torna quindi in stato "da confermare"). La modifica di un esame può riguardare anche la sua data, in questo caso l'utente deve seguire un procedimento particolare che verrà spiegato dopo aver chiarito e dettagliato come avviene la programmazione di un nuovo esame in ExamPlan. Un docente ha la possibilità di modificare o eliminare esclusivamente i propri appelli d'esame, un operatore ha invece pieno controllo su tutti i dati della piattaforma.

2.2.3 Vincoli

La creazione di un nuovo appello d'esame avviene nel rispetto dei vincoli della realtà accademica analizzati precedentemente, con distinzione fra vincoli "Hard" e vincoli "Soft" sulla base della necessità di applicazione.

Si definiscono **Vincoli Hard** quelle regole che devono necessariamente essere rispettate e **Vincoli Soft** quelle condizioni cui è bene porre attenzione ma che non necessariamente comportano un conflitto vincolante per la programmazione.

Nel nostro caso di studio, il principale Vincoli Hard riguarda gli insegnamenti di uno stesso corso di studi e stesso anno, esteso anche al caso di un insegnamento mutuato in cui vanno considerati congiuntamente diversi corsi di studi. Si considerano Vincoli Soft le relazioni fra insegnamenti di stesso corso ma anno differente, quelle fra esami di uno stesso docente e la distanza minima temporale fra esami dello stesso tipo.

Un operatore ha la possibilità di scegliere quali Vincoli Soft deve considerare il sistema durante il processo di pianificazione, attivando o disattivando la loro applicazione nelle impostazioni della piattaforma.

2.2.4 Anno Accademico

Le associazioni fra i dati della piattaforma esistono nel contesto di un determinato anno accademico. L'interfaccia fa sempre riferimento ai dati dell'anno accademico selezionato dall'utente.

L'operatore ha la possibilità di programmare un nuovo anno accademico in due modi: inizializzando il nuovo anno con gli stessi dati dell'anno precedente (stessi corsi di studi e insegnamenti) o inizializzandolo vuoto e inserendo manualmente i dati in un momento successivo.

Ogni anno accademico è caratterizzato da due semestri, ognuno dei quali ha un periodo di lezioni e un periodo di sospensione delle lezioni (sessione d'esami) impostabili nella piattaforma di amministrazione.

2.3 Funzionalità

Di seguito vengono presentate e descritte nel dettaglio le funzionalità caratteristiche di ExamPlan.

2.3.1 Come si presenta ExamPlan

Gli esami programmati vengono mostrati nell'interfaccia principale di ExamPlan tramite un calendario, che consente di avere una chiara e immediata visione sulla distribuzione degli stessi e sulla disponibilità delle date per nuove prenotazioni.

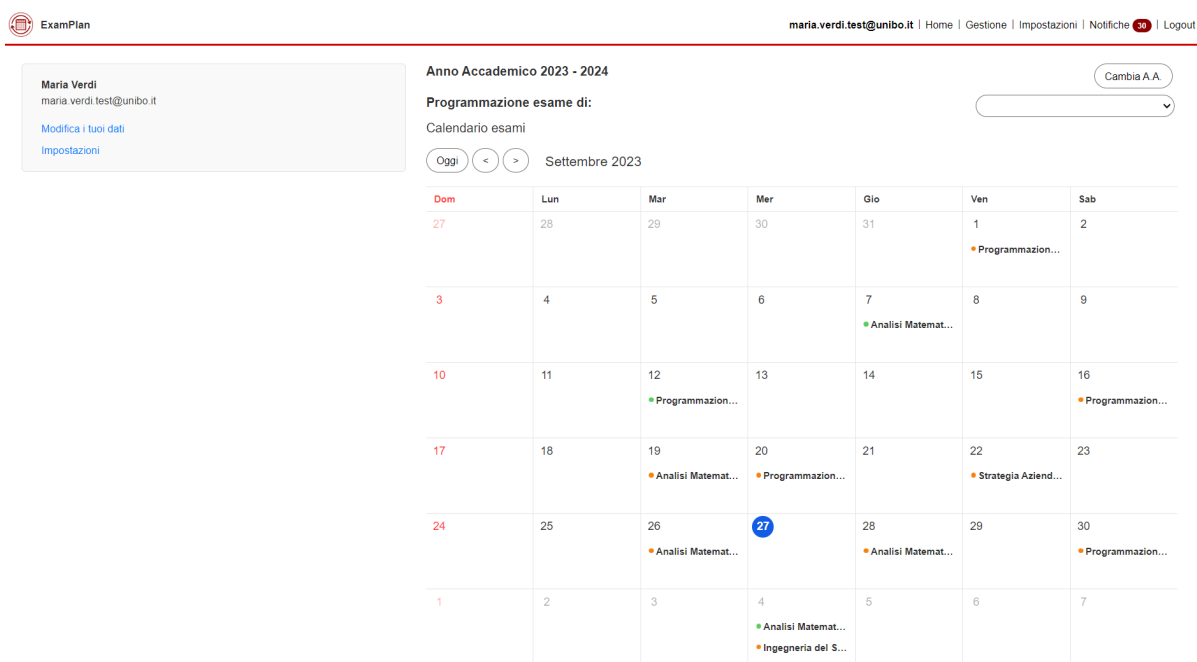


Figura 2.2: Come si presenta ExamPlan

Il calendario di un particolare docente è caratterizzato dalla visualizzazione di tutti i suoi esami già schedulati, e in più dagli esami di altri insegnamenti che hanno influenza sulla sua programmazione, ovvero esami di stessi corsi di studi dei suoi.

È possibile distinguere gli esami da gestire da quelli già confermati grazie alla colorazione dell'icona associata all'evento, arancione nel primo caso e verde nel secondo. Un docente ha informazione in merito allo stato dei soli propri appelli, gli esami di altri docenti hanno l'icona nera e non ne viene visualizzato il luogo.

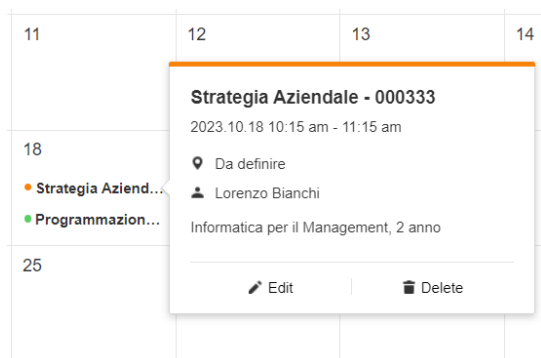


Figura 2.3: Esame da confermare

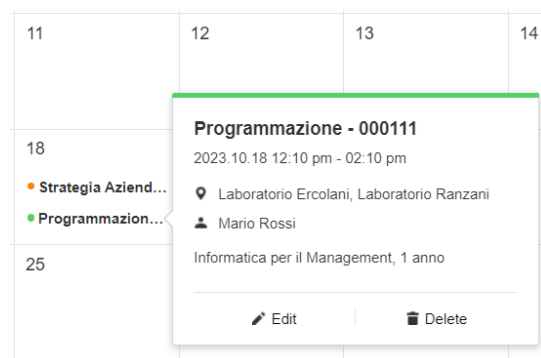


Figura 2.4: Esame confermato

2.3.2 Programmazione di un nuovo esame

Prima di passare a una spiegazione descrittiva dettagliata, viene presentato uno schema per individuare e comprendere le fasi che caratterizzano il processo di pianificazione di un nuovo appello d'esame, distinguendo le azioni che un utente deve compiere dai comportamenti in risposta dell'app.



Figura 2.5: Fasi della pianificazione di un esame

Come primo passo per la programmazione di un nuovo appello d'esame è quindi necessario scegliere l'insegnamento di cui si vuole effettuare la prenotazione. Selezionato l'insegnamento, il calendario viene in automatico colorato in base alla disponibilità dei giorni e

viene abilitato il click sui giorni di calendario “liberi”. La colorazione del calendario serve ancora una volta per guidare l’utente nella scelta della data migliore per la schedulazione di nuovi esami e avviene nel seguente modo: i giorni non disponibili in base al Vincolo Hard vengono colorati di rosso, i giorni in cui sono presenti esami in correlazione con l’insegnamento selezionato secondo il Vincolo Soft vengono colorati di giallo. Inoltre, vengono colorati di giallo un po’ più chiaro le date corrispondenti al distanziamento minimo consigliato rispetto ad appelli già schedulati di stesso insegnamento. I giorni rossi sono giorni in cui non è possibile schedulare un nuovo esame di tale insegnamento e i giorni gialli sono invece abilitati ma l’utente viene avvertito della possibilità che vengano riscontrate problematiche se aggiunto un esame in tali date.

La programmazione dell’esame avviene dunque cliccando su un determinato giorno disponibile del calendario e conseguente inserimento dei dati necessari.

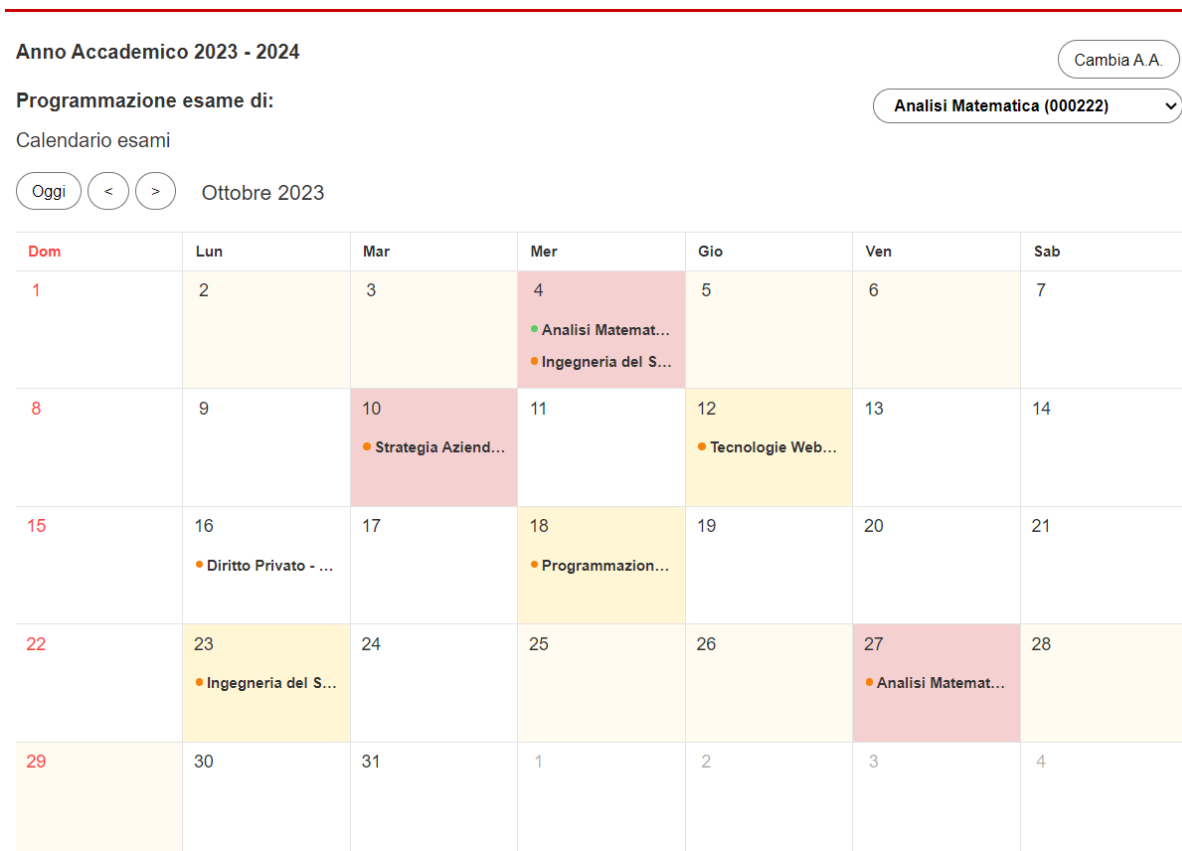


Figura 2.6: Programmazione di un nuovo esame con vincoli in ExamPlan

Dopo aver confermato la creazione di un nuovo esame, appare sul calendario l'icona del nuovo appello nel corrispondente giorno.

Se si intende modificare la data di un esame, l'utente ha la possibilità di selezionarne una nuova sul calendario, guidato dal sistema per rispettare esigenze e vincoli come avviene per la creazione di un nuovo evento. Il calendario verrà infatti colorato in base a Vincoli Hard e Soft relativi all'insegnamento di tale appello d'esame e l'utente potrà scegliere come nuova data una fra quelle disponibili. Nel caso in cui l'appello fosse già stato confermato, le aule assegnate verranno rilasciate, così come avviene per la modifica degli altri dati dell'esame.

2.3.3 Piattaforma di amministrazione

È presente in ExamPlan una piattaforma di amministrazione tramite la quale gli operatori della segreteria didattica hanno la possibilità di gestire e confermare gli esami programmati e manipolare i dati dell'intero sistema.

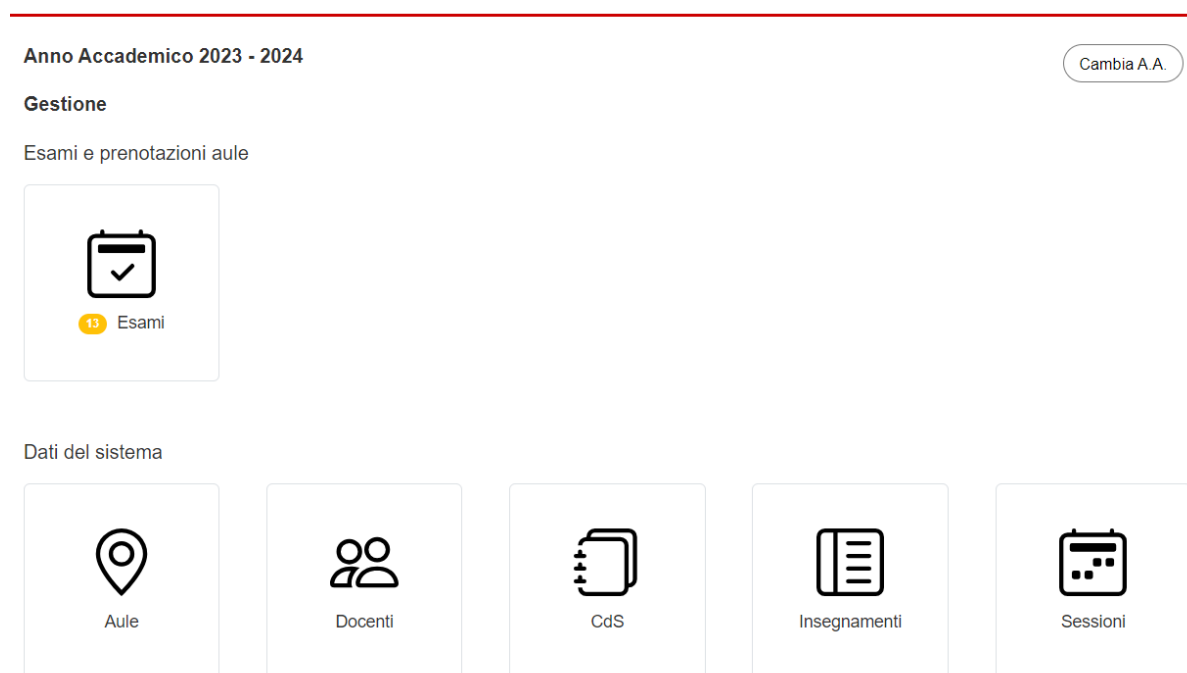


Figura 2.7: Pannello di amministrazione di ExamPlan

Nel momento in cui viene schedato un nuovo esame, un operatore deve intervenire per assegnare una o più aule all'evento in base a quanto specificato nella richiesta stessa. Il sistema elenca come aule prenotabili per un certo esame tutte le aule presenti sul sistema con capienza pari o maggiore rispetto a quella richiesta.


La prenotazione di una certa aula deve avvenire dopo un'operazione di verifica della disponibilità dell'aula stessa nella giornata e nella fascia oraria dell'esame in questione. È compito dell'operatore fare questo accertamento prima dell'assegnazione, operazione effettuata esternamente al sistema. Solo dopo questo intervento, l'evento diviene confermato.


Anno Accademico 2023 - 2024



Cambia A.A.


Gestione appelli d'esame



000223 - Analisi Matematica - Matematica

 Mattia Rossi

 26/09/2023

 11:50  60 minuti

 Esame orale

 1 aula  da 50 persone

Selezione aula
Sono elencate le aule con capienza pari o maggiore di quella richiesta.
Ricorda di verificare la disponibilità di una certa aula prima di assegnarla.

Aula 1

[Prenota aula](#)

Figura 2.8: Conferma esame e prenota aule

Per quanto riguarda la gestione dei dati del sistema, la piattaforma predispone un pannello per ogni tipologia di dato tramite il quale è possibile visualizzare, modificare, aggiungere o eliminare un qualsiasi elemento in modo semplice e immediato.

2.3.4 Sistema di notifica

ExamPlan è dotata di un sistema di notifica che consente di informare gli utenti in merito ad operazioni effettuate sui propri appelli d'esame e consentire un uso efficiente della piattaforma.

Il sistema di notifica è stato implementato nel seguente modo:

- se un docente crea o modifica un appello d'esame, viene inviata una notifica di riepilogo al docente stesso e una notifica agli operatori perché sappiano che è presente un nuovo evento da gestire
- se un operatore crea un appello d'esame, viene inviata una notifica di riepilogo ad operatore e a docente coinvolto e una notifica informativa agli operatori per la gestione del nuovo evento
- quando un operatore gestisce un esame, ovvero verifica i dati ed assegna le aule richieste, viene inviata una notifica informativa dell'avvenuta conferma sia a docente coinvolto che a operatore stesso

Ogni notifica consiste nell'invio di una mail e nella generazione di una nuova voce nel pannello delle notifiche dell'utente, integrato nella piattaforma.

Il contenuto di una notifica è composto dall'informazione relativa all'operazione effettuata e dai dati completi dell'appello d'esame coinvolto.

L'utente può scegliere se disattivare l'invio delle notifiche via mail dalle impostazioni del proprio profilo, per evitare la presenza di un eccesso di messaggi nella casella di posta elettronica che spesso comporta confusione e perdita di comunicazioni importanti. In ogni caso le notifiche continuano sempre ad essere generate e visibili nell'apposito pannello di ExamPlan.

Avere un'area dedicata per le sole informazioni relative alla gestione dei propri appelli d'esame integrata direttamente nell'applicazione, risulta essere una funzionalità ulteriore vantaggiosa.

Ecco come si presenta il pannello delle notifiche dell'applicazione.

Anno Accademico 2023 - 2024

Cambia A.A.

Le tue notifiche

Gestire nuovo appello d'esame	4 set
Riepilgo nuovo appello d'esame	4 set 09:39
Gentile utente, ecco i dettagli del nuovo appello d'esame creato: Docente: Mario Rossi Insegnamento: 000001 - Programmazione - Informatica per il Management Data e ora appello: 14/09/2023 alle ore 10:40 Durata: 2 ore Tipologia di esame: Esame orale Aule richieste: 1 di capienza Circa 50 posti Note per l'operatore: Note per gli studenti: L'appello d'esame verrà presto gestito da un operatore. Verrai notificato della conferma, con le informazioni relative all'aula/alle aule assegnate al tuo appello.	
Confermato appello d'esame	2 set
Confermato appello d'esame	26 ago

Figura 2.9: Pannello delle notifiche di ExamPlan

Le notifiche dell'utente loggato sono riportate per ordine di ricezione decrescente. È possibile distinguere i messaggi già letti da quelli non ancora visualizzati in base alla colorazione della notifica stessa (in grassetto se non visualizzata).

Capitolo 3

Progettazione

Dopo una chiara contestualizzazione del progetto di tesi è possibile analizzare di seguito la fase di progettazione e definire l'architettura e la struttura predisposta per la realizzazione dell'applicazione.

3.1 Base di dati

Per una qualsiasi applicazione, è fondamentale una progettazione accurata della base di dati, modellando in modo preciso e coerente la realtà di interesse sotto analisi. In questa sezione, vengono descritti i concetti fondamentali che costituiscono l'ossatura della base di dati di ExamPlan e il processo di implementazione del database vero e proprio.

3.1.1 Analisi della realtà di interesse

Grazie alla collaborazione con gli amministratori dell'attuale sistema PrenEsami è stata analizzata attentamente la realtà di interesse da modellare. Inizialmente, sono state individuate le componenti centrali presenti nel contesto della pianificazione degli esami universitari. A seguire, sono state aggiunte componenti secondarie necessarie per il funzionamento del sistema e la definizione di particolari vincoli e relazioni fra le entità della realtà stessa.

Componenti principali della realtà di interesse:

- **Utente:** utente del sistema, può essere un docente o un operatore della segreteria didattica. È colui che effettua una prenotazione.
- **Esame:** oggetto di una prenotazione. Ha varie caratteristiche, come tipologia, data, ora, ecc... e fa riferimento a uno specifico insegnamento.
- **Insegnamento:** corso facente parte di un piano di studi, tenuto da un docente.
- **Corso di Studi:** piano di studi dell'ateneo, costituito da un insieme di insegnamenti.
- **Stanza:** aula o laboratorio dell'ateneo. Può essere associata a un esame in fase di conferma dello stesso.
- **Anno Accademico:** annata in cui viene contestualizzata la calendarizzazione di specifici esami. Ogni insegnamento e corso di studi fa riferimento a uno specifico anno accademico.

Componenti secondarie:

- **Sessione:** fascia temporale di un anno accademico in cui pianificare esami.
- **Esame - Stanza:** esprime la relazione fra una o più stanze ad appelli d'esame.
- **Insegnamento correlato:** relazione fra istanze di insegnamento, esprime il concetto di insegnamento mutuato.
- **Notifica:** notifica di uno specifico utente da mostrare nel relativo pannello della piattaforma.
- **Impostazioni:** impostazioni dell'app - vincoli gestiti dal sistema durante una nuova pianificazione, messaggio di testo da mostrare agli utenti nell'interfaccia in fase di pianificazione.

3.1.2 Progettazione database

Di seguito viene riportato il modello della base di dati, rappresentazione astratta e strutturata dei dati e delle relazioni utile a una migliore comprensione e modellazione della realtà accademica sotto analisi.

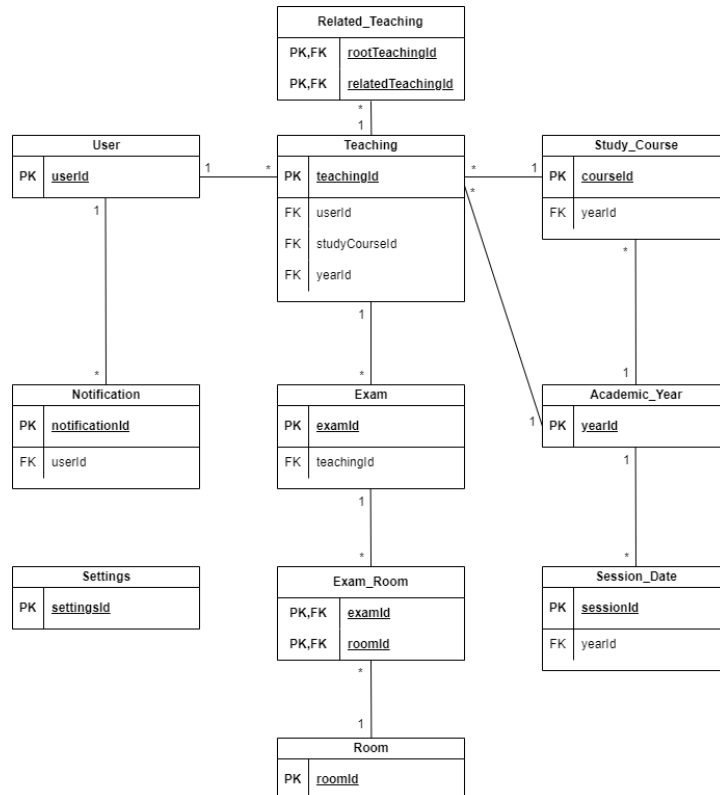


Figura 3.1: Modello dei dati dell'applicazione

3.1.3 Implementazione database

L'implementazione della base di dati di ExamPlan è stata guidata dalla scelta di adottare un **database relazionale** [6], con **MySQL** [7] come sistema di gestione dei database. Questa decisione è stata motivata dai numerosi vantaggi intrinseci offerti da questa struttura, che si adattano in modo ottimale alle esigenze di gestione di dati complessi e interconnessi.

Sono descritti di seguito i principali vantaggi offerti da un database relazionale.

Struttura organizzata : i dati sono organizzati in tabelle con righe e colonne, consentendo una rappresentazione chiara e strutturata delle informazioni. Ciò favorisce la comprensione e la gestione dei dati, facilitando l'individuazione di relazioni tra le diverse entità.

Relazioni definite : un database relazionale consente di stabilire relazioni significative tra diverse tabelle attraverso chiavi primarie e chiavi esterne.

Integrità dei dati : i database relazionali supportano regole di integrità dei dati, come vincoli di chiave primaria e vincoli di chiave esterna. Questi meccanismi assicurano che i dati siano coerenti, corretti e affidabili.

Flessibilità nelle query : SQL (Structured Query Language) è un linguaggio potente per l'interrogazione dei dati nei database relazionali. Offre flessibilità nell'esecuzione di query complesse e nel recupero delle informazioni desiderate.

Affidabilità e scalabilità : i database relazionali come MySQL sono noti per la loro affidabilità e scalabilità. Possono gestire grandi volumi di dati e possono essere ottimizzati per migliorare le prestazioni in situazioni ad alto carico.

Transazionalità : i database relazionali supportano transazioni, che garantiscono l'integrità dei dati anche in caso di errori o interruzioni. Questo è particolarmente importante in contesti accademici in cui l'accuratezza dei dati è essenziale.

Sicurezza dei dati : i database relazionali offrono opzioni avanzate per la gestione della sicurezza dei dati, inclusi i controlli di accesso ai dati e la crittografia.

La scelta di MySQL come sistema di gestione dei database per ExamPlan è stata dettata dalla sua ampiamente riconosciuta affidabilità e dalla sua efficacia nella gestione dei dati strutturati. La struttura relazionale si presta perfettamente alla rappresentazione dei vincoli e delle relazioni all'interno del contesto della pianificazione accademica, garantendo un solido fondamento per il sistema di gestione dei dati sottostante.

3.2 Architettura dell'applicazione

Un'architettura robusta è ciò che permette ad un software di essere sviluppato, implementato e mantenuto facilmente. È fondamentale adottare un approccio architetturale che rifletta le migliori pratiche per l'efficienza, la modularità e la scalabilità.

3.2.1 Design Pattern MVVM

Per raggiungere tali obiettivi, è stato scelto il **design pattern MVVM** (Model-View-ViewModel) [8] come guida principale nell'architettura di ExamPlan.

Tale design pattern è costituito da 3 componenti:

- **Model:** si riferisce al modello di dominio, rappresenta lo stato e i dati dell'app.
- **View:** definisce la struttura, il layout e l'aspetto dell'interfaccia utente. Visualizza una rappresentazione del modello e riceve gli input dall'utente.
- **ViewModel:** è il collegamento tra la vista e il modello, espone proprietà e metodi che la vista utilizza tramite data-binding. Il ViewModel gestisce la relativa View tramite un meccanismo di propagazione di eventi. È la componente che contiene la logica dell'applicazione.

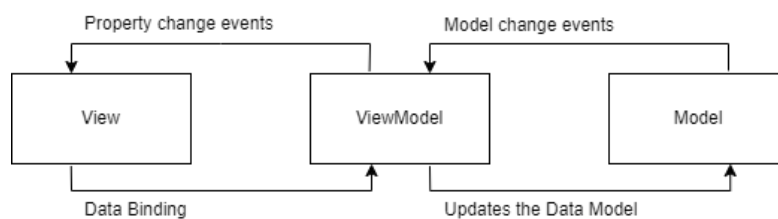


Figura 3.2: Design Pattern MVVM

Questo design pattern separa chiaramente la logica di presentazione (ViewModel) dalla rappresentazione dell'interfaccia utente (View) e dai dati sottostanti (Model). Ciò favorisce la separazione delle responsabilità e semplifica il processo di sviluppo, consentendo un miglior isolamento dei componenti e facilitando la manutenzione nel tempo.

3.2.2 Struttura Client - Server

La struttura del progetto si divide in due componenti: **Client** e **Server**. Questa struttura è essenziale per garantire un ambiente di sviluppo efficiente e scalabile, consentendo un'elevata flessibilità e manutenibilità dell'applicazione nel suo complesso.

L'architettura Client-Server è basata su una divisione chiara delle responsabilità e dei compiti tra le due parti distinte: la componente Client definisce l'interfaccia e si occupa della gestione delle interazioni con l'utente, la componente Server si occupa della gestione dei dati dell'applicazione.

Client e Server comunicano fra loro tramite il **protocollo HTTP** [9].

Il Client gestisce l'interfaccia dell'applicazione. Per ottenere ed effettuare operazioni sui dati presenti nel database, necessita dell'interazione col Server. La comunicazione tra Client e Server avviene tramite l'invio di richieste e risposte HTTP.

Il Server riceve le richieste dal Client e, in base al percorso specificato nell'URL, avvia il meccanismo di routing per determinare quale logica di gestione deve essere eseguita.

Il Server elabora la richiesta e risponde opportunamente al Client. Il Client riceve la risposta e renderizza l'interfaccia utente conseguentemente.

L'interazione Client-Server è una comunicazione bidirezionale che consente all'utente di visualizzare e interagire con l'interfaccia (Client) e al Server di gestire i dati e rispondere alle richieste (Server). Questo flusso costante di richieste e risposte consente all'applicazione di funzionare in modo dinamico e reattivo.

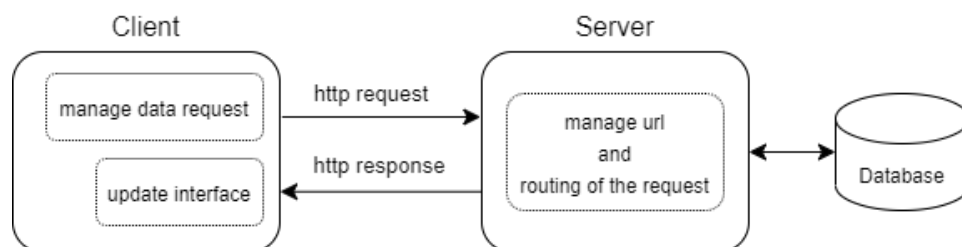


Figura 3.3: Interazione Client-Server

In sintesi, l'architettura di ExamPlan è strutturata in modo da favorire la modularità, la reattività e la scalabilità, basandosi sulla sinergia tra il design pattern MVVM, l'interazione Client-Server e l'utilizzo di framework JavaScript moderni che verranno presentati in seguito. Questa combinazione di scelte architetturali fornisce una solida base per la creazione di un'applicazione web efficiente, flessibile e in grado di gestire in modo ottimale la pianificazione degli esami universitari.

Capitolo 4

Implementazione

Avendo definito gli obiettivi e schematizzato la struttura del progetto, si procede con l'analisi delle tecnologie utilizzate per l'implementazione e la definizione delle principali componenti dello stesso.

4.1 Tecnologie

ExamPlan è un'applicazione web JavaScript che sfrutta appieno la potenza dei framework moderni. Lato Client, è stato utilizzato il framework **Vue.js** [10], noto per la sua reattività e la struttura modulare che si allinea perfettamente con il pattern MVVM. Lato Server, ExamPlan si avvale di un ambiente **Node.js** [11] e del framework **Express** [12], che offre una solida base per la creazione di API RESTful e la gestione delle richieste e delle risposte HTTP.

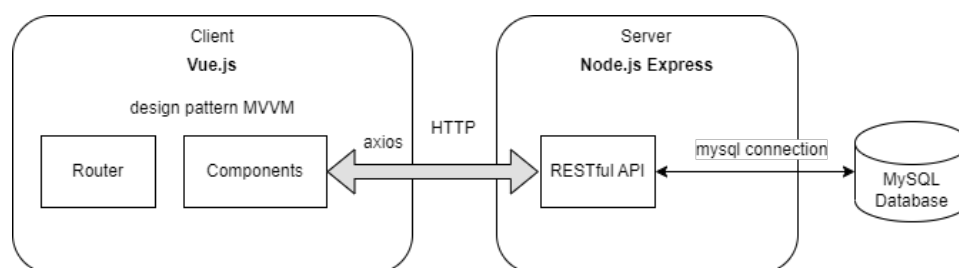


Figura 4.1: Architettura e tecnologie di ExamPlan

4.2 Client Side

4.2.1 Il framework Vue.js

Vue.js è un framework open-source JavaScript utilizzato per la creazione di interfacce utente reattive e dinamiche. È progettato per semplificare lo sviluppo delle parti front-end delle applicazioni web, consentendo agli sviluppatori di creare interfacce utente interattive con maggiore facilità ed efficienza.

Vengono elencate le caratteristiche chiave di Vue.js.

Reattività : quando i dati cambiano, l'interfaccia utente si aggiorna automaticamente per riflettere tali cambiamenti senza la necessità di aggiornamenti manuali.

Struttura Modulare : struttura costituita da più componenti. Un componente è un blocco autonomo di funzionalità e interfaccia che può essere riutilizzato in diverse parti dell'applicazione, risparmiando la ripetizione di codice identico.

Template : meccanismo che permette agli sviluppatori di dichiarare in modo dichiarativo come l'interfaccia utente dovrebbe apparire in risposta ai dati caricati.

Come il design pattern MVVM è applicato alle componenti Vue

Oggetto “data” : è definito in ogni componente e contiene lo stato e i dati utilizzati per la definizione dell'interfaccia della componente stessa. I dati presenti nell'oggetto “data” costituiscono il **modello**.

Template : contiene il markup HTML che determinata come verranno visualizzati i dati all'interno dell'interfaccia utente. Definisce la **vista** di una certa componente.

ViewModel : il collegamento tra il modello e la vista è automaticamente gestito dal sistema di reattività di Vue, tramite una sincronizzazione fra dati e vista.

Configurazione dell'applicazione Vue

L'applicazione Client è configurata nel file “main.js”, nel quale viene inizializzata la vera e propria app Vue, vengono importate ed applicate ad essa tutte le librerie necessarie

al suo funzionamento e viene “montata” l’istanza dell’applicazione Vue su un elemento specifico del DOM (l’elemento con id “app”). Questo è un passaggio fondamentale per far sì che la componente principale dell’app Vue venga renderizzata e visualizzata sulla pagina web. La pagina web in cui viene caricata l’app Vue è “index.html” che si trova nella cartella “public”. Quest’ultima, è l’unica pagina HTML del progetto, nella quale vengono caricate dinamicamente viste differenti dell’app durante il corso di navigazione fra pagine del sistema.

4.2.2 Componenti principali e meccanismo di autenticazione

ExamPlan è caratterizzata dal seguente schema di componenti:

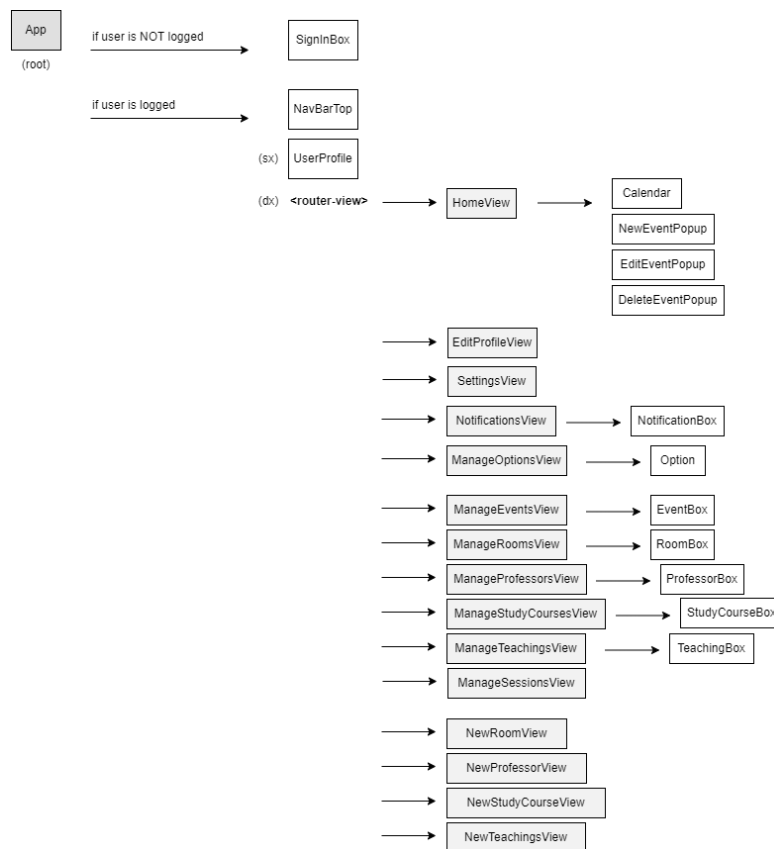


Figura 4.2: Schema delle View e delle Componenti di ExamPlan

Esclusa l'interfaccia di login che ha un suo specifico layout, il componente radice **App.vue** definisce il template per tutte le pagine dell'applicativo, caratterizzato dalla presenza di una barra di navigazione in alto (**NavBarTop**), da una colonna di sinistra con le informazioni dell'utente (**UserProfile**) e una colonna di destra in cui viene caricato il contenuto in base alla route visualizzata dall'utente (inserito al posto del tag “**router-view**” presente nel template).

All'interno della componente App.vue stessa si verifica se l'utente è autenticato o meno, e in base al risultato viene considerata la corretta porzione di codice del template, che comporta il caricamento della view corrispondente alla route corrente.

Meccanismo di autenticazione

Il meccanismo di accesso alla piattaforma è stato progettato per supportare le due diverse tipologie di utenti, docenti e operatori, ciascuno con privilegi e funzionalità specifiche. L'autenticazione avviene attraverso un processo di verifica delle credenziali, composte da email e password. Dopo aver verificato l'identità dell'utente, il sistema effettua una verifica del ruolo associato all'account. Questa fase di verifica del ruolo è fondamentale in quanto determina quali funzionalità e sezioni dell'applicazione sono accessibili e visibili all'utente che ha fatto l'accesso.

4.2.3 Meccanismo di navigazione: Vue-Router

Per la gestione della navigazione fra le viste dell'applicazione è stata utilizzata la libreria Vue-Router [13]. Tale libreria consente di creare un'applicazione a pagina singola (**Single Page Application, SPA**) con una navigazione fluida e reattiva. All'interno di una SPA, il concetto di navigazione avviene senza che il browser debba caricare una nuova pagina HTML per ciascuna transazione. Invece, il **router** manipola dinamicamente l'URL e cambia la vista mostrata all'utente senza ricaricare l'intera pagina.

Vue-Router permette di definire **regole di routing** e la corrispondenza fra URL e specifiche componenti Vue.

In ExamPlan è presente un file dedicato alla gestione delle varie route dell'applicazione. Nella cartella “router” è presente infatti il file “index.js” che implementa la libreria Vue-Router. Ogni route è caratterizzata da un path univoco, un nome identificativo ed è associata a una certa view dell'applicazione (componente che viene caricata quando tale route è la corrente).

4.2.4 Interazione col server: Axios e Vue-Axios

Per ottenere e gestire i dati presenti nel database, l'applicazione Client deve necessariamente interagire col Server. A questo scopo, sono state utilizzate le librerie Axios [14] e Vue-Axios [15].

Axios è una libreria JavaScript per effettuare **richieste HTTP** tramite un'interfaccia semplice e intuitiva. **Vue-Axios** è un wrapper necessario per **l'integrazione di Axios** in un progetto **Vue**. Permette di effettuare facilmente richieste HTTP da componenti Vue e gestirne i risultati, semplificando il processo di chiamate API e l'aggiornamento dello stato dell'app in risposta ai dati ricevuti dal Server. Axios gestisce le richieste e le risposte in modo asincrono, per ottenere e inviare dati al server in modo efficace e senza interruzioni dell'interfaccia utente.

L'implementazione di tale libreria è configurata nel file “axios.js” che si trova nella cartella “plugins”. Esso definisce l'URL su cui il server è caricato, che durante la fase di sviluppo è localhost:8089.

Tutte le componenti che necessitano di comunicare col Server fanno riferimento a tale file e utilizzano l'istanza axios definita per l'invio di una specifica richiesta HTTP.

4.2.5 Gestione dati di sessione tramite LocalStorage: Vuex

I dati di sessione includono le informazioni relative all'autenticazione degli utenti e altri dati necessari per fornire un'esperienza utente coerente e personalizzata. Tali dati devono persistere durante la sessione dell'utente e attraverso diverse pagine dell'app. Per affrontare questa esigenza, è stato utilizzato il meccanismo del **LocalStorage** per la persistenza dei dati e la libreria Vuex [16], che agisce come un sistema di gestione dello

stato globale. Grazie all'uso di Vuex, l'app usa una componente detta “**store**” che fa riferimento a un'entità centralizzata in cui vengono conservati e gestiti i dati condivisi tra le varie componenti. Lo “store” è costituito da uno stato (dati), mutazioni (funzioni che modificano lo stato) e getters (funzioni per ottenere dati dello stato).

Tale meccanismo consente di gestire il flusso dei dati in modo chiaro e separato dalla logica delle componenti e permette una gestione centralizzata dello stato dell'app, rendendola più facile da mantenere e gestire.

Il LocalStorage viene gestito nel file “index.js” presente nella cartella “store”. I dati che vengono memorizzati nello store dell'applicazione sono: dati e insegnamenti dell'utente autenticato, dati che individuano l'anno accademico e l'insegnamento selezionato per la prenotazione di un nuovo esame, ed eventuali messaggi informativi o di errore da segnalare all'utente. Si tratta di informazioni necessarie per l'impostazione dell'interfaccia e la gestione delle operazioni effettuate dall'utente.

Da una qualsiasi componente dell'app, è possibile accedere ai dati dello store tramite il comando “this.\$store.state” seguito dal nome della proprietà che si vuole ottenere e alle mutazioni tramite l'istruzione “this.\$store.commit('NomeMutazione', parametro)”. Per ottenere i getters è possibile invece utilizzare la funzione “mapGetters” fornita dalla libreria Vuex, che consente di utilizzare direttamente il valore restituito dai getter nel template o nei metodi della componente senza dover far riferimento manualmente allo store.

4.2.6 Il calendario di ExamPlan: Tui-Calendar

Come descritto in precedenza, si è deciso di integrare un calendario in ExamPlan per la visualizzazione e la pianificazione di esami. L'obiettivo primario era offrire un'esperienza utente fluida e intuitiva, adattabile alle diverse esigenze di pianificazione.

Le caratteristiche chiave per la libreria di calendario utile nel nostro caso erano: un'interfaccia utente user-friendly, la capacità di visualizzare eventi nei relativi giorni, la gestione agevole degli eventi e un grado di personalizzazione che consentisse di allineare il calendario all'estetica dell'app.

Tra le varie opzioni prese in considerazione, le cui principali sono le librerie FullCalendar [17] e Vue-Cal [18], è stata adottata la libreria **Tui-Calendar** [19]. Questa scelta è stata guidata dalle sue caratteristiche di personalizzazione e funzionalità offerte nella gestione degli eventi. Tui-Calendar permette di integrare una varietà di viste di calendario per una panoramica completa degli eventi e fornisce la possibilità di creare, di modificare e cancellare eventi direttamente dalla vista del calendario, semplificando la gestione delle attività. Inoltre, la sua interfaccia intuitiva permette agli utenti di interagire con facilità.

L'implementazione del calendario si trova nella componente "Calendar" di ExamPlan. L'istanza viene inizializzata applicando configurazioni e tema personalizzati.

La libreria predispone inoltre determinati eventi cui associare comportamenti specifici per la gestione degli schedule del calendario.

Gli eventi utili alla nostra implementazione che sono stati gestiti sono i seguenti:

- **clickSchedule**, associato al click di uno specifico schedule: nel momento in cui un esame programmato viene cliccato, viene visualizzata una finestra predefinita con i relativi dettagli e bottoni per la modifica/eliminazione. Nel nostro caso, viene gestito tale evento per determinare se visualizzare o meno tali bottoni, in quanto un certo evento può essere gestito esclusivamente da operatori e creatore dello stesso.
- **beforeUpdateSchedule**, associato al click del bottone di modifica: utilizzato per gestire la visualizzazione del popup personalizzato di modifica.
- **beforeDeleteSchedule**, associato al click del bottone di eliminazione: utilizzato per gestire la visualizzazione del popup personalizzato di eliminazione.
- **afterRenderSchedule**, evento scatenato da ogni singola pianificazione a seguito del rendering dell'intero calendario: la gestione di tale evento è necessaria per la colorazione del calendario secondo i vincoli predisposti.

Viene inoltre impostato sull'istanza di calendario un listener che associa un determinato metodo all'evento di **click** su calendario. Quest'ultimo è necessario per l'applicazione dei vincoli alla pianificazione di nuovi esami. Segue nel paragrafo successivo la descrizione dettagliata dell'applicazione dei vincoli.

4.2.7 Implementazione dei vincoli

Colorazione del calendario

La colorazione del calendario avviene tramite l'**applicazione dei vincoli** Hard e Soft della realtà accademica precedentemente descritti.

Selezionato l'insegnamento di cui si vuole effettuare una programmazione, vengono inizializzate le **liste di esami** necessarie al confronto con le schedulazioni di una specifica data.

Viene creata una lista contenente gli esami programmati di insegnamenti di stesso anno e corso rispetto all'insegnamento scelto (Vincolo Hard), una lista con gli esami di insegnamenti di stesso corso e anno differente (Vincolo Soft) e infine un'altra con gli esami del docente titolare dell'insegnamento selezionato (Vincolo Soft).

Per ciascuno schedule si verifica se presente in una delle liste preimpostate e conseguentemente viene colorata o non la relativa casella del calendario. Se un determinato Vincolo Soft è stato disabilitato dall'operatore, il confronto con la relativa lista non verrà effettuato. Se in una certa data sono presenti più eventi per i quali vale un vincolo, viene fatto prevalere il Vincolo Hard e la relativa colorazione in rosso della casella.

La colorazione dei giorni che indicano la distanza minima consigliata rispetto agli appelli di stesso insegnamento di quello selezionato avviene subito dopo l'aggiunta di tutti gli schedule sul calendario, tramite una lista delle date da colorare impostata durante il rendering di ciascun appello d'esame.

Pianificazione di un nuovo esame

La pianificazione di un nuovo esame avviene tramite la scelta di un particolare giorno del calendario disponibile. Il click su calendario comporta quindi la comparsa del popup per l'inserimento dei dati del nuovo evento.

La **disponibilità** di una certa data è determinata dai soli **Vincoli Hard**, ovvero è necessario disabilitare la possibilità di scegliere un giorno tale per cui sono soddisfatti

tali vincoli. In primo luogo, è quindi importante una verifica della data su cui è stato effettuato il click. Per controllare se in tale giorno è possibile pianificare, si effettua un confronto incrociato fra gli eventi schedulati in tale data e gli eventi di stesso anno e corso di studi dell'insegnamento selezionato. Se risultano esserci corrispondenze fra date, allora significa che non è possibile programmare un nuovo esame nel giorno scelto e il click in tale data è dunque disabilitato.

Inoltre, non è possibile schedulare eventi in giorni già trascorsi.

4.3 Server Side

4.3.1 Node.js e il framework Express

Node.js è un ambiente di runtime open-source basato su JavaScript che consente l'esecuzione di codice JavaScript lato Server. A differenza del tradizionale utilizzo di JavaScript, tale ambiente è altamente performante e scalabile per lo sviluppo di applicazioni web e servizi. Node.js è inoltre basato su un modello di programmazione asincrona e non bloccante, consentendo l'esecuzione di più operazioni in parallelo rendendo l'applicazione reattiva e prestazionale.

Express è un framework basato su Node.js che semplifica la creazione di applicazioni web ed API. Fornisce una struttura organizzata per definire e per gestire routing e richieste HTTP.

Si è scelto di utilizzare lato Server la combinazione di tali tecnologie in quanto estremamente performanti, flessibili e scalabili, costituiscono una solida base per uno sviluppo semplificato e intuitivo del back-end di un'applicazione.

L'applicazione Server Express è configurata nel file "app.js", dove vengono impostate dipendenze, middleware e configurazioni di base. Quando l'app viene avviata, viene lanciato il file "server_starter.js" che mette in ascolto il Server Express su una determinata porta. Il Server è quindi pronto a ricevere chiamate HTTP dal client.

4.3.2 API RESTful per la gestione delle richieste Client

Per consentire la comunicazione tra Client e Server attraverso richieste HTTP, è necessario definire lato Server particolari **servizi web** detti **API** (Application Programming Interface) [20]. Le API sono dei veri e propri intermediari software grazie ai quali applicazioni differenti possono essere messe in comunicazione e integrate.

Le API RESTful sono definite seguendo i principi dello **stile architetturale REST**, basato proprio sul protocollo HTTP. Il concetto fondamentale alla base di REST è l'esistenza di **risorse** (dati) a cui si può accedere tramite un identificatore globale, detto **URI**. Per utilizzare le risorse, le componenti di una rete (Client e Server) comunicano attraverso un'interfaccia standard (HTTP) per scambiare rappresentazioni di queste stesse risorse.

Il funzionamento prevede una struttura degli URL ben definita che identifica univocamente una risorsa o un insieme di risorse e l'utilizzo dei metodi HTTP specifici per il recupero di informazioni (GET), per l'aggiunta (POST), la modifica (PUT), l'eliminazione (DELETE) ed altri scopi.

Questo approccio fornisce una struttura chiara per creare servizi web che siano scalabili, interoperabili e che consentano l'accesso ai dati e alle funzionalità attraverso richieste HTTP ben definite.

Seguendo tali principi, sono state definite nel Server le API necessarie per la gestione di tutte le possibili richieste provenienti dal Client, relative all'accesso o alla manipolazione dei dati presenti nel database.

Processo di comunicazione

Il Client che è l'interfaccia utente dell'app, interagisce con il Server inviando richieste HTTP ai vari endpoint API. Ciascuna richiesta viene creata in base all'operazione che il Client desidera eseguire in un determinato istante.

Nel Server, il framework Express riceve le richieste HTTP in arrivo e le indirizza alle API corrispondenti, ognuna individuata in base al path e al metodo HTTP di una certa

richiesta. Questo coinvolge l'uso delle rotte definite in Express, che associano un endpoint a una funzione di gestione. Una volta gestita la richiesta ed elaborata la risposta, il Server risponde al Client tramite un nuovo messaggio HTTP.

Per mantenere il codice organizzato e gestibile, le API sono divise in diversi file router in base alla risorsa che gestiscono. In ciascun file sono definite le route e gli endpoint associati a una specifica risorsa. Per questo, nel file "app.js" si fa riferimento ad ogni router contenente API associandolo al relativo percorso.

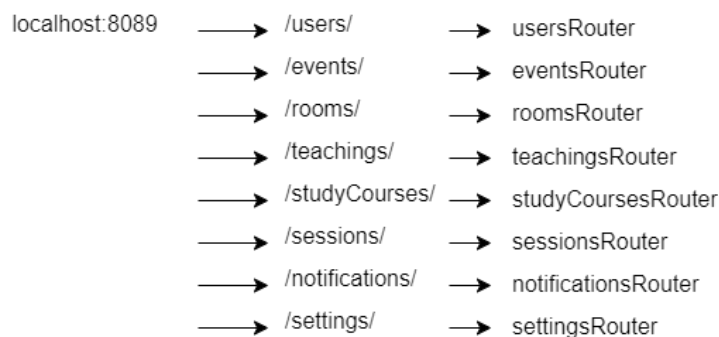


Figura 4.3: Schema delle API di ExamPlan

4.3.3 Connessione al database MySQL

Quando una richiesta necessita dell'accesso ai dati presenti nel **database**, il Server utilizza la libreria `mysql` per stabilire una connessione al database. Tale libreria offre funzionalità per creare, eseguire e gestire **query SQL**.

La connessione del Server al database è configurata nel file "db_conncet.js". Per istanziare la connessione è necessario specificare host, porta, utente, password e nome del database cui si vuole accedere. Tale istanza rappresenta la connessione al database vera e propria e viene utilizzata all'interno di ciascuna funzione di gestione associata a una chiamata API per ottenere e manipolare dati del database.

Valutazione e conclusioni

Il presente studio si è proposto di affrontare la sfida della pianificazione degli esami universitari considerando i vincoli specifici della realtà accademica come parte integrante del processo di planning.

L'obiettivo primario di questa tesi era quello di sviluppare un sistema focalizzato su un'efficace gestione di particolari vincoli, al fine di guidare gli utenti verso una programmazione semplificata e una migliore organizzazione degli esami universitari.

È nata così ExamPlan, applicazione progettata e implementata come soluzione per la schedulazione di eventi nel contesto di un ambiente con vincoli specifici.

L'introduzione di questo nuovo sistema mostra evidenti vantaggi rispetto al precedente approccio adottato dal Dipartimento di Informatica dell'UniBO, ipotetico destinatario della nuova applicazione.

Fornendo una chiara visione della calendarizzazione degli esami, consente agli utenti di creare un piano coerente rispetto ad eventi già schedulati ed esigenze analizzate. Grazie alla sua interfaccia intuitiva, ExamPlan permette a docenti e segreteria didattica di gestire e programmare esami con una migliore esperienza e velocità d'uso.

Di fondamentale importanza sono le funzionalità aggiuntive di ExamPlan rispetto a PrenEsami, quali la piattaforma di amministrazione e il pannello di notifiche integrato, che favoriscono la gestione dei dati di sistema e migliorano il processo di pianificazione nel suo complesso.

Inoltre, il refactoring dell'architettura e l'adozione di nuove tecnologie all'avanguardia hanno permesso di migliorare notevolmente scalabilità e manutenibilità del sistema.

Valutazione di ExamPlan

Al termine dello studio condotto da questa tesi e conclusa l'implementazione della nuova soluzione tecnologica ExamPlan, è stata svolta una valutazione preliminare dell'app per valutarne efficacia e potenzialità in una sua adozione futura.

Tale analisi e valutazione è stata effettuata internamente al team e coinvolgendo membri della segreteria didattica del DISI dell'UniBO, per ottenere un primo giudizio dai potenziali utenti destinatari.

Dalle prime analisi l'applicazione è stata apprezzata per la sua semplicità d'uso e per le funzionalità offerte che garantiscono una gestione centralizzata dei dati immediata. Un ulteriore potenziale individuato è il sistema di notifica automatico per ciascuna operazione svolta dagli utenti, con pannello delle notifiche integrato nella piattaforma stessa e possibilità di disattivare l'arrivo delle mail.

È prevista una futura valutazione più estesa, con il coinvolgimento di un numero di utenti destinatari maggiormente elevato, non solo operatori della segreteria ma anche docenti.

Sviluppi futuri e integrazione del sistema

ExamPlan vuole quindi essere una prima versione del nuovo applicativo per un aggiornamento delle metodologie ora utilizzate per la gestione della pianificazione degli esami. Si prevedono dunque una serie di sviluppi futuri perché ExamPlan possa essere adottata dall'Università di Bologna ed integrata coi relativi sistemi attuali.

Viene proposta di seguito una lista di punti su cui focalizzare i prossimi sviluppi:

- Implementare un servizio che permetta di importare eventi da altri calendari: ciò consentirebbe agli utenti di visualizzare direttamente sul calendario di ExamPlan ogni impegno personale, avendo una chiara visione sulla disponibilità dei giorni per una nuova pianificazione. Gli eventi importati potranno essere considerati come nuovi vincoli Hard o Soft.
- Aggiungere la possibilità per docenti e operatori di inserire nuovi vincoli personalizzati: ad esempio un utente potrebbe decidere di impostare un Vincolo Soft per il

quale un determinato giorno della settimana deve sempre essere considerato come “sconsigliato”, o potrebbe aggiungere Vincoli Hard per rendere “non disponibili” specifici giorni nonostante la mancata presenza di eventi nel calendario.

- Implementare nell’applicazione un meccanismo di verifica della disponibilità delle aule: attualmente, quando un operatore deve assegnare una o più aule a un esame da confermare può scegliere fra tutte le aule del sistema con capienza pari o uguale a quella richiesta dal docente per tale esame, senza però sapere quali di queste sono già occupate nella giornata prevista per tale evento. Si prevede quindi la possibilità di automatizzare la fase di verifica della disponibilità di una certa stanza in corrispondenza di un determinato giorno e fascia oraria.
- Integrazione dei meccanismi di autenticazione dell’Università di Bologna, per rendere ExamPlan parte dell’ecosistema dell’università e semplificare sia l’esperienza utente che la gestione complessiva del sistema. Questo comporterebbe sicuramente una serie di vantaggi come la semplificazione nella gestione delle credenziali da parte degli utenti, offrendo loro un accesso univoco ai diversi sistemi dell’università, comporterebbe un miglioramento nella gestione della sicurezza riducendo il rischio di accessi non autorizzati, oltre a consentire la sincronizzazione delle informazioni degli utenti per avere dati sempre aggiornati e coerenti.
- Ulteriori estensioni potrebbero coinvolgere l’integrazione di Microsoft Teams e AlmaEsami in ExamPlan. Il collegamento automatico a Teams consentirebbe agli utenti di comunicare e collaborare in modo semplice e immediato, condividere risorse e organizzare riunioni virtuali associate ad eventi della piattaforma. Il collegamento diretto ad AlmaEsami, piattaforma dell’Università di Bologna per la visualizzazione degli esami programmati e prenotazione di quest’ultimi da parte degli studenti, consentirebbe di automatizzare il processo di pubblicazione di un esame al momento della sua conferma, evitando l’intervento ulteriore da parte del docente.

Adattare l'app a un contesto differente

Sebbene inizialmente progettata specificamente per la pianificazione di esami universitari, ExamPlan apre le porte verso possibili adattamenti in svariati ambiti.

Dallo studio effettuato per valutare la fattibilità di adattare questa piattaforma ad altri contesti, emerge la possibilità di sfruttare il suo nucleo centrale: la pianificazione strategica rispettando vincoli precisi.

L'analisi si concentra dunque su una completa astrazione del sistema, dalla quale emergono le seguenti tre componenti principali: utente, evento, vincolo.

L'utente è colui che effettua e gestisce la programmazione di un evento, l'evento è l'oggetto della pianificazione e il vincolo è la condizione per la quale l'evento può essere schedato (è una relazione fra eventi e/o utenti).

Gli eventi che vengono visualizzati sul calendario della piattaforma potrebbero dunque essere eventi di un qualsiasi tipo, tramite un semplice intervento nell'adattare l'interfaccia rispetto alle proprietà specifiche che li caratterizzano.

L'applicazione dei vincoli a una nuova pianificazione verrebbe anch'essa gestita come nell'implementazione corrente. Ovviamente è necessaria la definizione dei vincoli della particolare realtà di interesse: nell'app in questione questo si traduce nel creare in maniera opportuna le liste di eventi che costituiscono l'oggetto che determina se una certa data è da considerare disponibile o non e permette l'inizializzazione del calendario.

Anche per quanto riguarda la piattaforma di amministrazione, sarebbe opportuno un intervento per adattare interfaccia e sistema ai nuovi dati: ciò non comporterebbe un elevato sforzo o impiego di tempo, si tratta di una semplice analisi delle differenze in termini di caratteristiche proprie dei nuovi elementi rispetto a quanto previsto attualmente, adattando la visualizzazione dei dati nell'interfaccia e i servizi di gestione e manipolazione del database.

Grazie alla sua struttura modulare, i concetti di ExamPlan potrebbero quindi essere adattati a svariati ambiti in maniera semplice.

Vengono proposti di seguito altri contesti di pianificazione di eventi con vincoli:

- Calendario di attività sportive: solitamente le palestre offrono vari corsi cui gli utenti possono partecipare. I gestori hanno dunque la necessità di pianificare gli eventi rispettando vincoli relativi alla disponibilità di istruttori, sale della struttura, ed evitando sovrapposizioni fra eventi in correlazione.
- Pianificazione di conferenze e riunioni aziendali: le aziende spesso devono pianificare riunioni tra compartimenti o clienti. Gli stessi principi di rispetto dei vincoli temporali e delle disponibilità possono essere applicati al contesto aziendale.
- Programmazione di eventi culturali e artistici: gli eventi come spettacoli teatrali, concerti e mostre d'arte richiedono una pianificazione dettagliata per assicurarsi che tutti gli aspetti siano in linea e coerenti rispetto al relativo contesto.
- Pianificazione di servizi medici: anche gli ospedali e le strutture sanitarie necessitano di un'accurata pianificazione di appuntamenti, considerando disponibilità relative a personale, risorse e spazi.

Ogni contesto ha le sue esigenze specifiche di pianificazione e vincoli da considerare.

Il vero punto innovativo di ExamPlan risiede nell'abilità di affrontare e integrare una vasta gamma di vincoli, con una loro possibile suddivisione in base al livello di rilevanza.

Le linee guida (riflesso dei vincoli impostati) che la piattaforma fornisce durante il processo di pianificazione permettono agli utenti di programmare eventi in modo semplificato e nel rispetto delle esigenze analizzate nella specifica realtà.

Bibliografia

- [1] Zucchetti Spa, EasyAcademy.
<https://www.zucchetti.it/website/cms/prodotto/2969-gestione-appelli-desame-e-orari-lezioni-universita.html>
- [2] Zucchetti Spa, Introduzione ad EasyTest.
<https://www.youtube.com/watch?v=nEtNQwTW7II>
- [3] Apereo, UniTime.
<https://www.unitime.org/index.php?tab=0>
- [4] CollegeNet, Series25: programmazione e gestione delle risorse.
<https://www.collegenet.com/products/scheduling/scheduling.html>
- [5] Applied Software Consultants, AscTimetables.
<https://www.asctimetables.com>
- [6] Atzeni P., Ceri S., Fraternali P., Paraboschi S., Torlone R., Basi di dati, McGraw-Hill, 2018
- [7] Oracle, MySQL.
<https://www.mysql.com/it/>
- [8] Microsoft, Model-View-ViewModel Pattern.
<https://learn.microsoft.com/en-us/dotnet/architecture/maui/mvvm>
- [9] Kurose J.F. e Ross K.W., Reti di calcolatori e Internet, Un approccio top-down, Pearson, 2017

- [10] Vue.js, The progressive javascript framework.
<https://vuejs.org/>
- [11] Node.js, Documentazione.
<https://nodejs.org/it/docs>
- [12] Express, Framework web veloce, non categorico e minimalista per Node.js.
<https://expressjs.com/it/>
- [13] Vue Router, The official Router for Vue.js.
<https://router.vuejs.org/>
- [14] Axios, What is Axios
<https://axios-http.com/docs/intro>
- [15] Vue Axios, A small wrapper for integrating Axios in vue.js.
<https://www.npmjs.com/package/vue-axios>
- [16] Vuex, What is Vuex.
<https://vuex.vuejs.org/>
- [17] FullCalendar, JavaScript Calendar.
<https://fullcalendar.io/>
- [18] Vue-Cal.
<https://antoniandre.github.io/vue-cal/>
- [19] NHN Cloud, Toast UI Calendar.
<https://nhn.github.io/tui.calendar/latest/>
- [20] IBM, Cos'è un'API REST.
<https://www.ibm.com/it-it/topics/rest-apis>