



**Politecnico
di Torino**

Politecnico di Torino

Corso di Laurea

A.a. 2024/2025

Sessione di laurea Luglio 2025

FacultyApp: creazione del frontend dell'applicazione mobile per i docenti del Politecnico di Torino

Relatori:

Luigi De Russis

Giuseppe Piazza

Candidati:

Gabriele Medica

Ringraziamenti

Innanzitutto vorrei ringraziare i miei relatori e tutti coloro che mi hanno affiancato durante questo percorso di realizzazione della tesi.

Ma il grazie più grande va alla mia famiglia, e in particolare a voi: papà, mamma e Aurorina. Mi avete sempre supportato, nel bene e nel male. Vi voglio un mondo di bene, anche se a volte non lo dimostro: senza di voi non ce l'avrei mai fatta.

Anche a Paolettina, la mia fidansatina, grazie. Hai sempre creduto in me incondizionatamente, anche quando io ero il primo a dubitare. Ad ogni mio successo durante questo percorso, eri felice persino più di me. Da quando ci sei tu, tutto è diventato magicamente più facile. Ti amo.

Un ringraziamento speciale va anche a tutti i miei amici.

Rubio, mio fratello, grazie per esserci stato sempre, dal primo anno delle superiori fino a oggi – e si spera ancora per molto, almeno fino al viaggio in Turchia! Scusa se non ti ho fatto stampare la tesi in 3D.

Vinz, grazie, anche se mi hai abbandonato per andare in Francia. Ma tanto lo so che è una cosa passeggera e che tornerai presto da me.

Max, grazie per avermi accettato come tuo amico nonostante ti abbia detto di no mille volte per ogni festa techno a cui mi hai invitato. Serve che porto giù la spazzatura, per caso?

Gianlu, grazie perché mi hai salvato la vita parecchie volte e ci hai sempre intrattenuti con le tue disgrazie amorose. Non ti ringrazio del tutto solo perché sei di Sampieri e non vieni quasi mai a Modica quando siamo giù.

Nick, grazie per tutti i taglieri che mi hai offerto (e che mi offrirai), per i ristoranti swag che ci hai consigliato, e soprattutto grazie per il regalo di Natale. Davvero, non dovevi.

Un grazie va anche ai miei amici che mi hanno dovuto sopportare anche come coinquilino. Gianni, sei durato quattro anni e poi ti ho fatto scappare, ma almeno siamo ancora amici. Grazie, e salutami Marmotta! Infine, ma non per importanza: grazie Lucu. Ti ho conosciuto solo quest'anno, ma siamo già molto amici. Mi mancava qualcuno con cui ballare il reggaeton... gli altri hanno ancora bisogno di qualche lezione.

Ci sono cose per cui vorrei ringraziarvi che voi sapete, io so, ma che non posso scrivere qui. Quindi, a tutti i miei amici, anche a chi non ho nominato: grazie davvero, vi voglio veramente bene.

YOOOOOOOOO.

Indice

Elenco delle figure	VIII
1 Introduzione	1
1.1 Contesto e motivazioni	1
1.2 Obiettivi della tesi	2
2 Background	3
2.1 User-centered design	3
2.2 Metodologia	4
3 Analisi preliminare e raccolta dei requisiti	5
3.1 Metodologia di raccolta dati	5
3.1.1 Scelta dei partecipanti	6
3.1.2 Struttura e contenuti del questionario	6
3.2 Approfondimento delle esigenze emerse	7
3.2.1 Funzionalità esistenti da migliorare	7
3.2.2 Funzionalità amministrative richieste	8
3.2.3 Necessità legate alla didattica	9
3.2.4 Necessità aggiuntive suggerite	10
3.2.5 Comunicazione e gestione del materiale didattico	10
3.3 Analisi dei requisiti	11
3.3.1 Analisi quantitativa	11
3.3.2 Analisi qualitativa	13
3.4 Sintesi e definizione dei requisiti	14
3.4.1 Requisiti funzionali	14
3.4.2 Requisiti non funzionali	15
4 Progettazione del prototipo Mid-Fidelity	16
4.1 Scelte di design visuale e stile grafico	17
4.1.1 Coerenza con il sistema visivo PoliTO	17
4.1.2 Focus sulla funzionalità	20

4.1.3	Scelte di prototipazione	20
4.2	Architettura dell'informazione	21
4.3	Prototipazione con Figma	22
4.3.1	Descrizione generale dei servizi progettati	23
4.3.2	Rappresentazione dei servizi nel prototipo	23
5	Sviluppo dell'applicazione	26
5.1	Scelte tecnologiche	26
5.2	Architettura del frontend	27
5.2.1	Gestione della navigazione	27
5.2.2	Gestione dei dati fittizi	27
5.2.3	Lingua e tema	28
5.3	Scelte stilistiche e UI	28
5.3.1	Adattamenti e personalizzazioni	29
5.3.2	Coerenza e leggibilità	29
5.3.3	Uso di NativeWind per lo stile	29
6	Implementazione dei servizi e componenti	30
6.1	Profilo	30
6.1.1	Funzionalità implementate	31
6.1.2	Architettura del componente	34
6.1.3	Componenti adottati	34
6.2	Didattica	36
6.2.1	Funzionalità implementate	36
6.2.2	Architettura del componente	38
6.2.3	Componenti adottati	45
6.2.4	Problemi riscontrati e soluzioni	48
6.3	Agenda	49
6.3.1	Funzionalità implementate	49
6.3.2	Architettura del componente	50
6.3.3	Componenti adottati	55
6.4	Servizi	57
6.4.1	Funzionalità implementate	57
6.4.2	Architettura del componente	58
6.4.3	Componenti adottati	65
6.4.4	Luoghi	67
7	Usability Testing	69
7.1	Obiettivi e metodologia del test	69
7.2	Descrizione dei partecipanti	70
7.3	Scenari di utilizzo testati	70

7.4	Risultati del test	73
8	Conclusioni	77
8.1	Sviluppi futuri	77
	Bibliografia	79

Elenco delle figure

3.1	Esempi di funzionalità amministrative fortemente richieste.	8
3.2	Funzionalità didattiche più richieste dagli utenti.	9
3.3	Distribuzione dei partecipanti in base all'utilizzo dell'app esistente.	11
3.4	Lista di funzionalità esistenti e percentuali di utilizzo.	12
4.1	Confronto tra la visualizzazione dei servizi nelle due applicazioni.	18
4.2	Confronto tra la sezione avvisi nelle due applicazioni	19
4.3	Architettura dell'informazione del prototipo PoliTO Faculty.	22
4.4	Flusso interattivo di modifica di un avviso all'interno di un corso.	24
6.1	Schermata principale del profilo con smartcard e dati personali.	31
6.2	Form di modifica delle informazioni personali.	32
6.3	Schermata delle impostazioni del profilo.	33
6.4	Componente SectionHeader con titolo e azione.	34
6.5	Esempio di ListItem	34
6.6	IconButton in alto a destra.	35
6.7	Navigazione iniziale della sezione Didattica.	37
6.8	Panoramica sulla sezione di Info Corso	39
6.9	Panoramica sulla sezione Avvisi	40
6.10	Panoramica sulla sezione Materiale	41
6.11	Panoramica sulla sezione Lezioni	42
6.12	Panoramica sulla sezione Studenti	43
6.13	Panoramica sulla sezione Elaborati	44
6.14	Esempi di SectionHeader con opzioni.	45
6.15	Navigazione a tab all'interno di un corso.	45
6.16	Menu contestuali visualizzati con Popover	46
6.17	CtaButtons nel registro lezioni o nel calendario.	46
6.18	Esempio di utilizzo del componente Switch negli avvisi.	46
6.19	Gestione permessi tramite BottomModal	47
6.20	Esempio di Select nei vari form.	47

6.21	In questo caso, il pulsante "+" porta al form per la creazione di una Lezione per il registro.	48
6.22	Schermata giornaliera: AgendaScreen	51
6.23	Schermata settimanale: AgendaWeekScreen	52
6.24	Form per nuovo appuntamento: NoteForm	53
6.25	Dettaglio di un evento: SingleElementScreen	54
6.26	Popover per la gestione del layout.	55
6.27	Esempio di LectureCard con descrizione e aula virtuale.	56
6.28	Esempio di LectureCard con descrizione e aula virtuale.	56
6.29	Schermata servizi principali	59
6.30	Schermi principali del servizio Persone.	60
6.31	Schermi principali del servizio Firma Digitale.	61
6.32	Schermi principali del servizio Segnala Guasto.	62
6.33	Schermi principali del servizio Prenotazione Spazi.	63
6.34	Schermi principali del servizio Emergenze.	64
6.35	Schermo della sezione Supporto audio-video	65
6.36	Una ServiceCard contrassegnata come preferita.	66
6.37	Esempi di utilizzo del componente Badge	67
6.38	Schermi principali del servizio Luoghi.	68

Capitolo 1

Introduzione

Negli ultimi anni, l'adozione di strumenti digitali all'interno degli atenei ha assunto un ruolo sempre più centrale nella gestione delle attività accademiche. Il Politecnico di Torino ha investito significativamente nello sviluppo di soluzioni mobile, con l'obiettivo di migliorare la fruibilità dei servizi da parte di studenti e docenti. Tuttavia, mentre l'applicazione per gli studenti ha raggiunto un buon livello di maturità, quella destinata al corpo docente ha mostrato diverse criticità in termini di funzionalità, usabilità e coerenza con gli standard grafici istituzionali.

L'esigenza di offrire un'interfaccia più moderna, intuitiva e in linea con le reali necessità del personale docente ha motivato l'avvio di un processo di riprogettazione, culminato nello sviluppo di una nuova applicazione: PoliTO Faculty. Il progetto ha beneficiato del supporto e del coinvolgimento diretto di ISIAD che ha collaborato attivamente alla definizione dei requisiti e alla validazione delle soluzioni proposte. Questo progetto di tesi si inserisce all'interno di tale percorso, concentrandosi in particolare sulla realizzazione del frontend dell'applicazione mobile.

1.1 Contesto e motivazioni

L'idea alla base di questa tesi nasce dalla consapevolezza delle limitazioni dell'app esistente per i docenti e dalla volontà di migliorarne l'efficacia attraverso un processo strutturato di ascolto degli utenti. A differenza della controparte rivolta agli studenti, l'applicazione attuale per i docenti risultava poco aggiornata e scarsamente utilizzata, a causa di una combinazione di problemi tecnici, scelte progettuali superate e una scarsa aderenza ai flussi di lavoro reali del personale accademico.

Il contesto tecnologico del Politecnico di Torino è caratterizzato dalla presenza di un dipartimento IT interno (ISIAD), responsabile della progettazione e manutenzione dei principali strumenti digitali utilizzati da studenti, docenti e personale

amministrativo. Questa struttura interna consente un'elevata capacità di personalizzazione delle soluzioni informatiche, ma comporta anche la necessità di ottimizzare le risorse disponibili e concentrare gli sforzi su ciò che è più strategico.

In tale quadro, le esigenze del personale docente non sempre sono state affrontate con la stessa priorità attribuita alle soluzioni rivolte alla popolazione studentesca. Ne è derivata una situazione in cui, da un lato, esiste una solida base di servizi accessibili da web, ma dall'altro lato manca un'interfaccia mobile moderna e adeguata alle esigenze lavorative quotidiane dei docenti. I riscontri raccolti negli anni mostrano come molti preferiscano non utilizzare affatto l'applicazione esistente, affidandosi invece a soluzioni web, spesso più complete ma meno immediate e accessibili in mobilità.

A livello tecnico, l'app per docenti esistente era stata sviluppata con tecnologie ormai obsolete, difficili da mantenere e aggiornare nel tempo, e con un'interfaccia poco allineata alle linee guida visive adottate dall'ateneo nelle applicazioni più recenti. Ciò ha reso evidente la necessità di una reimplementazione completa, sfruttando stack tecnologici aggiornati e interfacce coerenti con quelle già utilizzate con successo nell'app PoliTO per studenti.

Da qui l'idea di ripensare completamente l'applicazione per docenti, progettando un sistema che non solo fosse più funzionale e aggiornato, ma che potesse anche integrarsi in modo coerente nell'ecosistema digitale esistente dell'ateneo, condividendo componenti, stili e logiche già validate. Questo approccio ha favorito un riuso intelligente delle soluzioni già sviluppate, valorizzando l'esperienza maturata dal team interno nel progetto student-facing, ma adattandola alle peculiarità operative e comunicative del personale docente.

1.2 Obiettivi della tesi

L'obiettivo principale di questa tesi è la progettazione, implementazione e valutazione del *frontend della nuova applicazione mobile per i docenti*, con un focus particolare sul frontend e l'interazione. In aggiunta, gli obiettivi possono essere così delineati:

- Analizzare i bisogni del personale docente attraverso un'indagine strutturata;
- Definire i requisiti funzionali e non funzionali dell'applicazione;
- Progettare un prototipo mid-fidelity che rappresenti i flussi e i servizi richiesti;
- Sviluppare il frontend dell'app utilizzando **React Native** e componenti standard istituzionali;
- Validare l'interfaccia realizzata tramite test di usabilità con utenti reali.

Capitolo 2

Background

2.1 User-centered design

Il progetto descritto in questa tesi si basa sull'approccio *user-centered design* (UCD), una metodologia di progettazione centrata sull'utente che pone le sue esigenze, aspettative e contesto d'uso al centro di ogni fase del processo di sviluppo. Tale approccio prevede il coinvolgimento attivo degli utenti finali – in questo caso i docenti del Politecnico di Torino – dalla raccolta iniziale dei requisiti funzionali fino alla validazione finale dell'interfaccia.

Durante le fasi preliminari, è stato realizzato un *mid-fidelity prototype*, ovvero un prototipo navigabile a fedeltà intermedia. Si tratta di una versione semplificata dell'interfaccia utente, priva dello stile grafico definitivo, ma utile per testare i flussi principali e la disposizione dei contenuti. Il prototipo è stato costruito utilizzando lo strumento Figma, adottando esclusivamente i *componenti UI istituzionali*, cioè elementi grafici e interattivi forniti dal Politecnico di Torino per garantire coerenza visiva, accessibilità e riconoscibilità tra le applicazioni dell'ecosistema digitale di Ateneo.

La successiva fase di sviluppo ha visto l'adozione di *React Native*, un framework open-source per la creazione di applicazioni mobile multiplatforma (iOS e Android), che consente di mantenere un'unica base di codice scritta in JavaScript. Questa scelta ha permesso di accelerare il processo implementativo e garantire una maggiore uniformità tra i diversi dispositivi. Nelle prime fasi, in assenza di accesso diretto ai servizi back-end reali, si è fatto ricorso a *mock data* (dati fittizi): insiemi di informazioni simulate utilizzati per testare l'interfaccia in condizioni realistiche.

Uno degli strumenti fondamentali utilizzati per validare le scelte progettuali è stato il *usability testing*, ovvero l'osservazione diretta degli utenti mentre eseguono compiti reali e rappresentativi. Questo tipo di test consente di rilevare criticità sia quantitative (tempi di esecuzione, numero di errori, completamento dei task) che

qualitative (commenti spontanei, momenti di confusione, aspettative disattese), fornendo una base concreta per il miglioramento iterativo dell'interfaccia.

L'adozione di questa metodologia e degli strumenti sopra descritti ha consentito lo sviluppo di un progetto coerente con i principi dell'*user-centered design*, capace di rispondere con maggiore precisione alle esigenze del contesto accademico per cui è stato concepito.

2.2 Metodologia

Il lavoro è stato organizzato seguendo un approccio *user-centered*, articolato in diverse fasi:

1. **Raccolta dati:** è stato realizzato un questionario destinato ai docenti del Politecnico di Torino, volto a identificare le principali esigenze legate all'uso di strumenti digitali in ambito didattico e amministrativo;
2. **Analisi dei risultati:** i dati ottenuti sono stati analizzati sia in chiave quantitativa che qualitativa, con l'obiettivo di sintetizzare una lista di requisiti chiari e orientati all'utente;
3. **Prototipazione:** è stato sviluppato un prototipo mid-fidelity dell'interfaccia mediante lo strumento Figma, utilizzando i componenti ufficiali previsti dall'Ateneo;
4. **Sviluppo del frontend:** sulla base del prototipo, è stato realizzato il frontend dell'applicazione in **React Native**, integrando funzionalità simulate mediante dati fittizi;
5. **Usability testing:** l'interfaccia è stata testata con un campione selezionato di docenti per valutare l'efficacia del design e raccogliere feedback per eventuali miglioramenti futuri.

Capitolo 3

Analisi preliminare e raccolta dei requisiti

La definizione accurata dei requisiti rappresenta una fase fondamentale per lo sviluppo di qualsiasi applicazione software orientata all'utente. In questo capitolo viene descritto il processo seguito per raccogliere e analizzare i dati necessari a comprendere le esigenze del personale del Politecnico di Torino, destinatario della nuova applicazione PoliTO Faculty.

3.1 Metodologia di raccolta dati

Per fondare il progetto su basi concrete, è stato realizzato un questionario indirizzato al corpo docente dell'Ateneo. L'obiettivo era duplice: da un lato, valutare l'efficacia percepita dell'applicazione esistente; dall'altro, raccogliere suggerimenti e richieste in vista della progettazione di una nuova soluzione più funzionale e accessibile.

La progettazione e la somministrazione del questionario sono state condotte in collaborazione con l'area IT del Politecnico di Torino (Dipartimento ISIAD), che ha fornito supporto nella definizione dei quesiti e nella distribuzione del modulo attraverso i canali istituzionali. Il coinvolgimento diretto dell'area tecnica ha garantito maggiore affidabilità nella raccolta dei dati e nella selezione del campione.

Il questionario ha consentito di raccogliere dati eterogenei: informazioni demografiche, modalità di utilizzo degli strumenti digitali, criticità riscontrate, esigenze specifiche e proposte migliorative. L'analisi integrata (compiuta anche seguendo le direttive di Fink [1]) di risposte chiuse e aperte ha permesso di ottenere sia una visione quantitativa che qualitativa del contesto d'uso.

3.1.1 Scelta dei partecipanti

Il questionario è stato distribuito tramite mail istituzionale e compilato su base volontaria da professori, ricercatori, assegnisti e collaboratori didattici, con l'intento di raccogliere feedback da un campione il più possibile eterogeneo.

La fascia anagrafica prevalente è risultata quella nata tra gli anni '70 e '80, che rappresenta circa il 50% del totale dei rispondenti. In termini di ruolo:

- **Professori** (43%)
- **Ricercatori** (36%)
- **Assegnisti** (25%)
- **Docenti esterni e/o collaboratori didattici** (19%)

La partecipazione è risultata ben distribuita anche tra i vari dipartimenti, con il tasso di risposta rispetto ai componenti totali mai inferiore al 25% e con i tassi più elevati registrati presso il **DIGEP** (41%) e il **DISMA** (42%).

Questa composizione ha reso il campione rappresentativo di una molteplicità di esperienze e punti di vista, rendendo più affidabile l'individuazione delle esigenze comuni.

3.1.2 Struttura e contenuti del questionario

Il questionario, realizzato con Google Forms, si articolava in circa 30 domande suddivise in più sezioni tematiche. L'obiettivo era quello di esplorare a fondo il rapporto dei docenti con l'attuale ecosistema digitale dell'Ateneo, in particolare l'app PoliTO. Le sezioni principali includevano:

- Dati anagrafici e professionali (anno di nascita, dipartimento, ruolo);
- Dispositivi utilizzati quotidianamente e frequenza d'uso;
- Esperienza con l'app esistente: funzionalità più usate, problemi riscontrati, motivi del mancato utilizzo;
- Esigenze funzionali: preferenze per l'accesso a servizi didattici e amministrativi da mobile;
- Suggerimenti e proposte: risposte aperte su miglioramenti attesi, funzionalità desiderate e criticità da risolvere.

Le domande a risposta chiusa includevano scale di valutazione (es. Likert, una scala numerica utilizzata per misurare il grado di accordo con un'affermazione), selezioni

multiple e opzioni binarie, mentre quelle aperte hanno permesso ai partecipanti di esprimere liberamente opinioni e richieste specifiche.

Questa struttura ha garantito l'ottenimento di dati misurabili, ma anche spunti qualitativi fondamentali per l'identificazione di pattern ricorrenti e esigenze trasversali.

3.2 Approfondimento delle esigenze emerse

L'esame più dettagliato delle risposte fornite dai partecipanti è stato condotto sfruttando strumenti di intelligenza artificiale (AI), in particolare modelli linguistici in grado di analizzare e sintetizzare grandi quantità di testo libero in modo efficiente.

Sono stati utilizzati più modelli AI per ottenere una visione comparativa delle ricorrenze tematiche e delle priorità espresse. A seguito di questa prima elaborazione automatica, è stato effettuato un incrocio manuale e critico dei risultati, con l'obiettivo di convalidare le interpretazioni ed estrarre le categorie di bisogno più rilevanti.

Questo processo ha consentito di individuare cinque principali aree di intervento, che riflettono in modo coerente le difficoltà e le aspettative emerse dal contesto reale d'uso.

3.2.1 Funzionalità esistenti da migliorare

Diversi partecipanti hanno indicato la necessità di migliorare alcune funzionalità già presenti nell'applicazione attuale, ma considerate limitate o poco efficaci. In particolare:

- Appelli d'esame, calendario e orario delle lezioni sono tra le sezioni più utilizzate, ma spesso giudicate poco chiare o lente nel caricamento.
- La sezione incarichi didattici richiede una gestione più flessibile e una visualizzazione dei contenuti più efficace.
- Le mappe e la funzione aule libere, pur essendo disponibili, risultano di difficile consultazione a causa della bassa precisione o della scarsa usabilità.

3.2.2 Funzionalità amministrative richieste

È emersa una forte richiesta di strumenti che semplifichino le attività burocratiche e gestionali. Tra le necessità maggiormente evidenziate, spiccano la prenotazione di spazi per lezioni, riunioni o eventi interni, e la firma digitale di documenti direttamente da mobile, come illustrato nei grafici in Figura 3.1.

Altre funzionalità richieste includono un sistema per la segnalazione di guasti o emergenze, con possibilità di inviare foto e geolocalizzazione, e strumenti per la gestione delle missioni, incluse funzionalità per l'inserimento di giustificativi o richieste di anticipi.

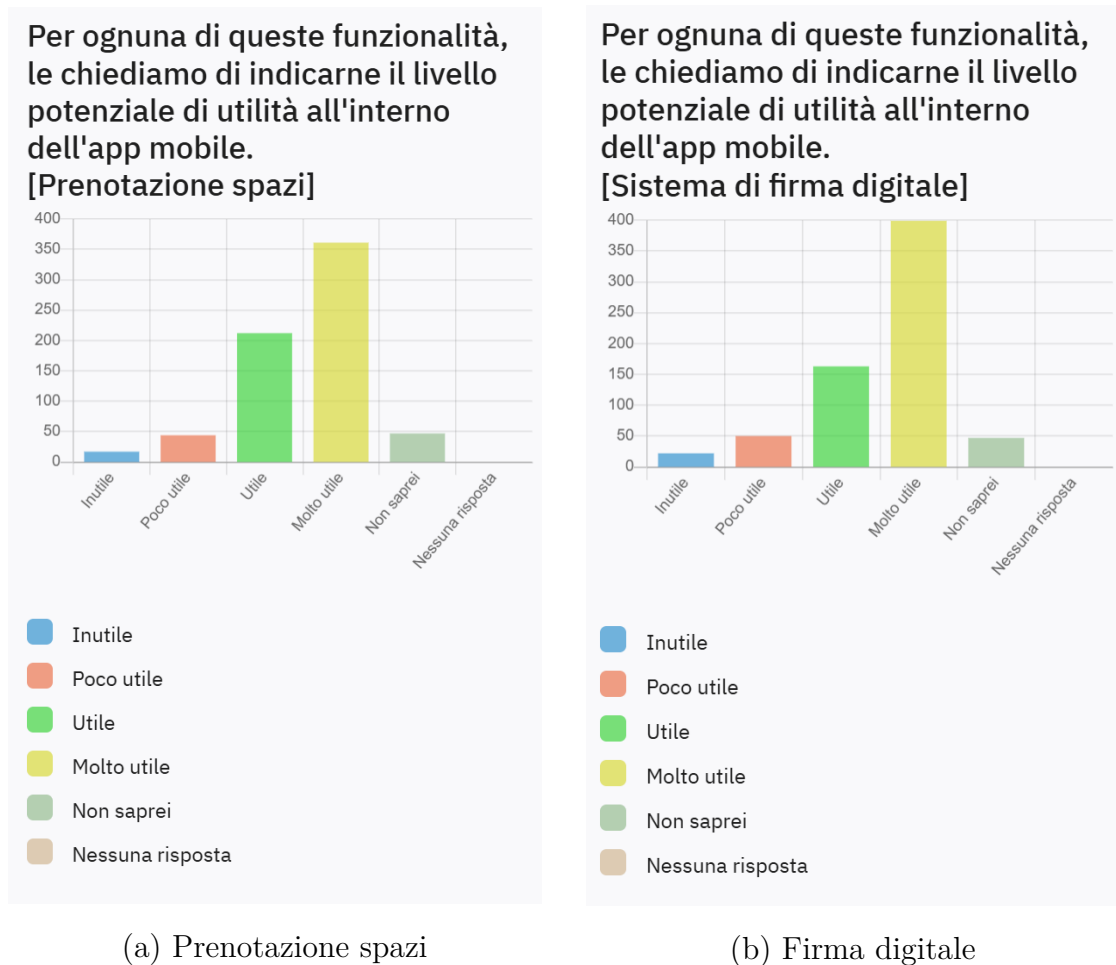


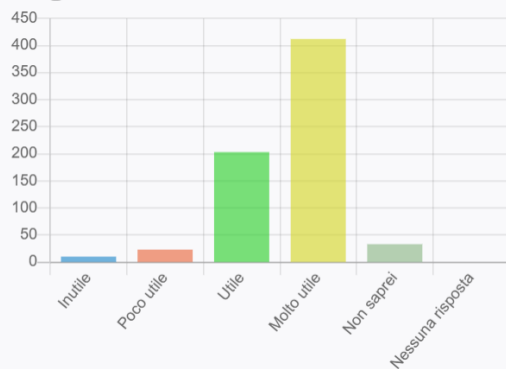
Figura 3.1: Esempi di funzionalità amministrative fortemente richieste.

3.2.3 Necessità legate alla didattica

Molti docenti hanno evidenziato la necessità di una maggiore integrazione tra l'app e le attività didattiche quotidiane. In particolare, è stata sottolineata l'importanza di poter visualizzare e gestire facilmente gli studenti iscritti ai corsi o agli appelli d'esame, nonché di poter caricare e condividere in modo rapido il materiale didattico. Queste due funzionalità sono risultate tra le più richieste, come mostrato nei grafici in Figura 3.2.

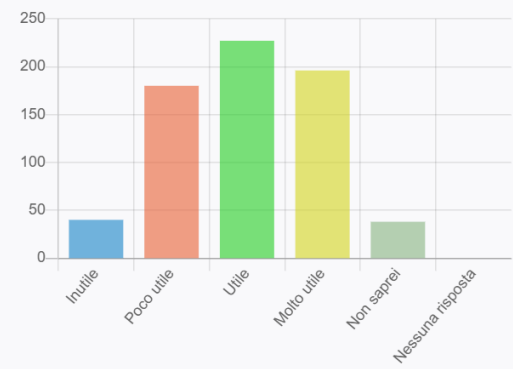
Oltre a queste, altri suggerimenti hanno riguardato l'invio diretto di avvisi agli studenti e l'integrazione con strumenti esterni già adottati, come Moodle o il Portale della Didattica, al fine di semplificare e centralizzare le operazioni quotidiane.

Per ognuna di queste funzionalità o sezioni, le chiediamo di indicarne il livello potenziale di utilità all'interno dell'app mobile.
[Esame (elenco iscritti, calendario, luogo,...)]



(a) Gestione degli esami

Per ognuna di queste funzionalità o sezioni, le chiediamo di indicarne il livello potenziale di utilità all'interno dell'app mobile.
[Materiale didattico]



(b) Materiale didattico

Figura 3.2: Funzionalità didattiche più richieste dagli utenti.

3.2.4 Necessità aggiuntive suggerite

Oltre alle esigenze più strettamente legate alla didattica e all'amministrazione, i partecipanti hanno proposto ulteriori funzionalità che puntano ad ampliare il raggio d'azione dell'app. In molti hanno richiesto la possibilità di cercare rapidamente aule libere, persone all'interno dell'Ateneo, eventi in programma e documenti istituzionali. Un'altra proposta ricorrente riguarda l'introduzione di un profilo personale docente, che raccolga dati contrattuali, incarichi assegnati e storico delle attività svolte.

Alcuni hanno suggerito anche l'integrazione di un calendario sincronizzabile con strumenti esterni, utile per ricevere notifiche su scadenze importanti o eventi imminenti. Infine, una parte del campione ha espresso interesse per l'accesso da mobile a servizi informativi complementari, come il menu della mensa, gli orari dei trasporti pubblici o iniziative promozionali dell'Ateneo come PoliSave. Queste proposte, sebbene non direttamente legate alla didattica, evidenziano il desiderio di un'app capace di centralizzare l'esperienza accademica del docente anche nelle sue componenti di contesto.

3.2.5 Comunicazione e gestione del materiale didattico

Un tema particolarmente ricorrente nelle risposte aperte riguarda la comunicazione con gli studenti e la gestione del materiale didattico. Molti docenti hanno espresso frustrazione per l'eccessiva frammentazione degli strumenti attualmente in uso: avvisi pubblicati sul portale, email, canali Telegram, Microsoft Teams e altre soluzioni vengono utilizzati in parallelo, spesso senza un'integrazione tra loro. Questa molteplicità di canali rende difficile garantire la ricezione puntuale delle comunicazioni da parte degli studenti, e molti docenti hanno manifestato il bisogno di un sistema centralizzato che fornisca maggiore tracciabilità e conferma di lettura dei messaggi inviati.

Anche la condivisione dei materiali risulta poco pratica, specialmente in situazioni in cui è necessario inviare file o comunicazioni urgenti da dispositivi mobili. Nonostante l'adozione di strumenti ufficiali come il Portale della didattica, una parte significativa dei partecipanti ha dichiarato di trovarne l'interfaccia migliorabile e poco adatta all'uso quotidiano. L'integrazione di funzionalità di comunicazione efficaci e semplici da utilizzare rappresenta quindi un'esigenza prioritaria, soprattutto in un contesto che richiede tempestività e continuità nella relazione tra docente e studenti.

3.3 Analisi dei requisiti

Dall’elaborazione dei dati raccolti tramite il questionario somministrato al corpo docente, è stato possibile trarre indicazioni fondamentali per definire i requisiti della nuova applicazione. L’analisi è stata condotta seguendo due approcci distinti ma complementari: da un lato quello quantitativo, volto a interpretare in modo statistico le risposte chiuse; dall’altro, quello qualitativo, orientato a cogliere esigenze e criticità espresse liberamente dai partecipanti.

Questa doppia prospettiva ha permesso di costruire una visione articolata dei bisogni reali, utile per delineare priorità progettuali concrete.

3.3.1 Analisi quantitativa

L’elaborazione statistica delle risposte ha evidenziato alcuni trend significativi. Il 66% dei docenti dichiara di aver utilizzato l’applicazione esistente, mentre il 22% non l’ha mai installata e il 12% l’ha rimossa dopo un primo utilizzo, come mostrato in Figura 3.3.

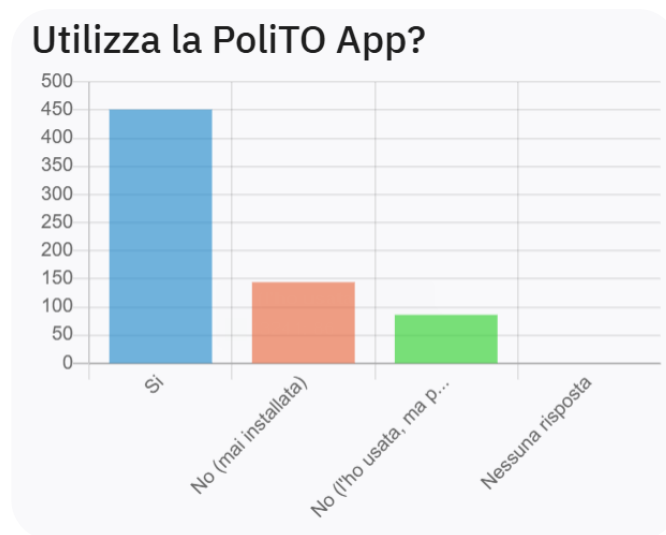


Figura 3.3: Distribuzione dei partecipanti in base all’utilizzo dell’app esistente.

Questo dato suggerisce una diffusione limitata e non omogenea dello strumento attuale. Per quanto riguarda le funzionalità più utilizzate, spiccano (come mostrato in Figura 3.4:

- Appelli d'esame (49,93%)
- Calendario delle lezioni (44,79%)
- Visualizzazione degli orari (37,00%)
- Sezione dedicata agli incarichi didattici (31,72%).

In parallelo, è emersa con forza l'esigenza di introdurre nuove funzionalità non ancora disponibili nell'applicazione attuale. Tra le più richieste vi sono strumenti per la gestione dettagliata degli esami, la possibilità di prenotare spazi per lezioni o riunioni, l'accesso rapido ai dati degli studenti e la firma digitale dei documenti.

L'interesse espresso verso questi aspetti è risultato particolarmente significativo, confermando la necessità di sviluppare un'applicazione più completa e centralizzata, capace di semplificare le attività quotidiane legate alla didattica e all'amministrazione.

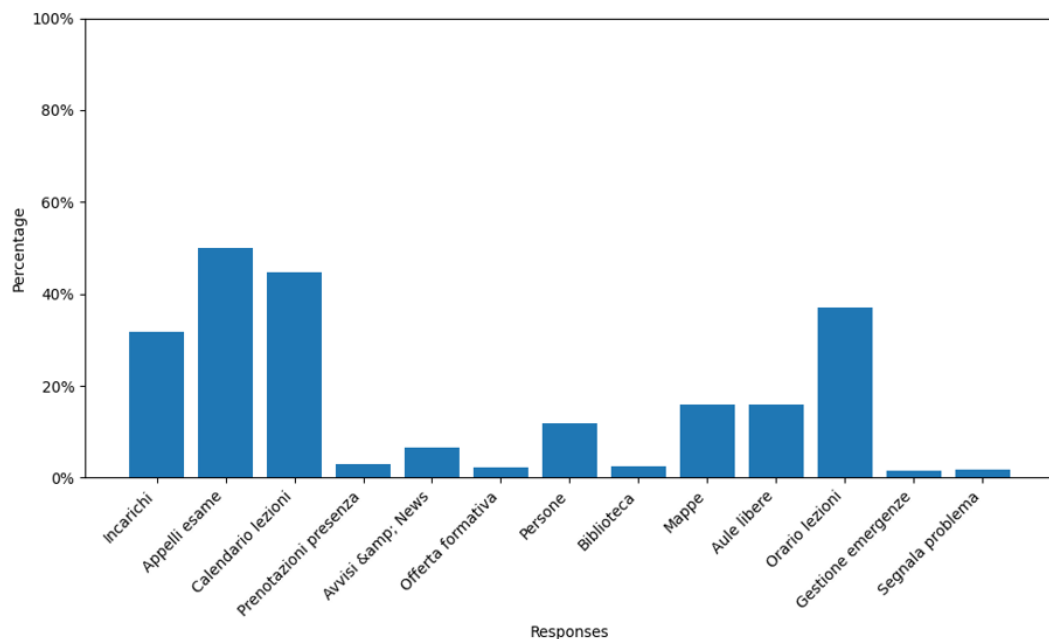


Figura 3.4: Lista di funzionalità esistenti e percentuali di utilizzo.

3.3.2 Analisi qualitativa

Le risposte aperte raccolte tramite il questionario hanno offerto una prospettiva più profonda sull'esperienza degli utenti, evidenziando aspetti che difficilmente emergerebbero da una semplice analisi statistica. In molti casi, i docenti hanno descritto una serie di difficoltà legate all'utilizzo dell'app attuale, soffermandosi su problematiche di natura tecnica e strutturale.

Tra le criticità più frequentemente menzionate figurano:

- **Interfaccia poco intuitiva e visivamente datata:** diversi utenti hanno sottolineato la scarsa leggibilità e l'inefficacia dell'interfaccia attuale. Come riportato: *“L'interfaccia non era comoda”, oppure “Interfaccia degli item poco leggibile o limitata rispetto al portale.”*
- **Frequenti problemi di stabilità come rallentamenti o crash:** alcuni partecipanti hanno segnalato problemi tecnici persistenti che hanno compromesso l'utilizzo dell'app: *“Spesso in crash o completamente priva di informazioni nelle diverse sezioni”, “Ho risolto con uno screenshot dal portale... e la disinstallazione dell'app.”*
- **Scarsa integrazione con altri strumenti digitali in uso all'interno dell'Ateneo:** è stata evidenziata la mancanza di interconnessione con strumenti esterni utili alla didattica: *“Comodità di condivisione file, integrazione con One Drive, gestione corsi di progetto di gruppo”, “Integrazione con il portale.”*
- **Sistema di notifiche ritenuto inefficace o mal configurato:** più docenti hanno espresso il bisogno di canali di comunicazione più affidabili e tracciabili: *“Gli studenti dovrebbero ricevere una notifica immediata sulla loro app”, “È un canale ufficiale, e mi aspetto che gli studenti ricevano una notifica.”*

Accanto a queste problematiche, i partecipanti hanno espresso aspettative piuttosto definite per la futura versione dell'applicazione. Tra le richieste principali spiccano una maggiore semplicità d'uso, una migliore affidabilità complessiva e una coerenza grafica con le altre piattaforme ufficiali, in particolare con l'app attualmente utilizzata dagli studenti. Un altro aspetto ricorrente riguarda la necessità di semplificare l'accesso alle funzionalità oggi distribuite su più portali, con l'obiettivo di migliorare l'efficienza nella gestione quotidiana di attività legate a didattica, esami e processi amministrativi.

Nel complesso, queste osservazioni spontanee hanno contribuito in modo significativo alla definizione dei requisiti, arricchendo l'analisi con elementi qualitativi che riflettono le esigenze reali degli utenti.

3.4 Sintesi e definizione dei requisiti

La fase di analisi ha consentito di tradurre le esigenze emerse dall'indagine in un insieme strutturato di requisiti, suddivisi in due macro-categorie: *funzionali* e *non funzionali*.

I requisiti funzionali riguardano ciò che l'app deve essere in grado di fare, mentre quelli non funzionali definiscono vincoli qualitativi e tecnici legati all'esperienza d'uso, alle prestazioni e alla coerenza con l'ecosistema digitale dell'Ateneo.

3.4.1 Requisiti funzionali

I principali requisiti funzionali individuati includono:

ID	Titolo	Descrizione	Priorità
RF1	Gestione appelli d'esame	Visualizzazione e modifica degli appelli, con accesso agli iscritti	Alta
RF2	Calendario e orari	Consultazione delle lezioni e gestione dell'orario personale	Alta
RF3	Incarichi didattici	Controllo e aggiornamento degli incarichi e responsabilità	Media
RF4	Dati studenti	Accesso alle informazioni anagrafiche e di carriera	Alta
RF5	Prenotazione spazi	Richiesta e gestione di aule e sale per eventi	Media
RF6	Firma digitale	Firma di documenti direttamente da mobile	Alta
RF7	Materiale didattico	Caricamento, condivisione e gestione di file	Alta
RF8	Comunicazioni agli studenti	Invio di avvisi e notifiche in-app	Alta
RF9	Gestione missioni	Inserimento e monitoraggio di trasferte e rimborsi	Bassa
RF10	Integrazione strumenti esterni	Collegamento con Moodle e altri sistemi istituzionali	Bassa

3.4.2 Requisiti non funzionali

Dal punto di vista qualitativo e tecnico, l'applicazione dovrà rispettare i seguenti requisiti non funzionali:

ID	Titolo	Descrizione	Priorità
RNF1	Usabilità	Interfaccia semplice, intuitiva, coerente con l'app studenti PoliTO	Alta
RNF2	Affidabilità	Stabilità dell'app, assenza di crash, ri-sposta rapida	Alta
RNF3	Accessibilità	Compatibilità con tecnologie assistive e standard WCAG	Media
RNF4	Performance	Tempi di caricamento ridotti, fluidità nella navigazione	Alta
RNF5	Sicurezza	Protezione dei dati personali, autenticazione con credenziali	Alta
RNF6	Manutenibilità	Codice modulare e riutilizzabile	Media
RNF7	Compatibilità	Supporto per i principali OS mobile (iOS, Android)	Alta
RNF8	Coerenza grafica	Uso dei componenti UI ufficiali dell'Android	Alta

Questa classificazione fornisce una base solida per la successiva fase di progettazione e sviluppo dell'interfaccia, assicurando che le scelte implementative siano guidate da bisogni reali e orientate alla qualità dell'esperienza utente.

Capitolo 4

Progettazione del prototipo Mid-Fidelity

Dopo aver raccolto e analizzato i requisiti funzionali e non funzionali, il passo successivo ha riguardato la progettazione di un prototipo mid-fidelity[2], con l'obiettivo di tradurre i bisogni emersi in soluzioni concrete e navigabili. Questa fase ha rappresentato un momento cruciale per verificare la coerenza tra ciò che gli utenti si aspettano e ciò che l'interfaccia potrà offrire.

Il prototipo è stato realizzato utilizzando lo strumento *Figma*, selezionato per la sua flessibilità e per il supporto a componenti condivisi, già impiegati nell'applicazione PoliTO rivolta agli studenti. Il lavoro si è concentrato sulla definizione dell'architettura dell'informazione, sull'organizzazione dei contenuti e sulla rappresentazione interattiva delle funzionalità prioritarie.

Nel corso del capitolo verranno illustrate le scelte progettuali adottate, la struttura informativa dell'applicazione e i principali servizi integrati nel prototipo, che fungerà da base per la fase successiva di sviluppo frontend.

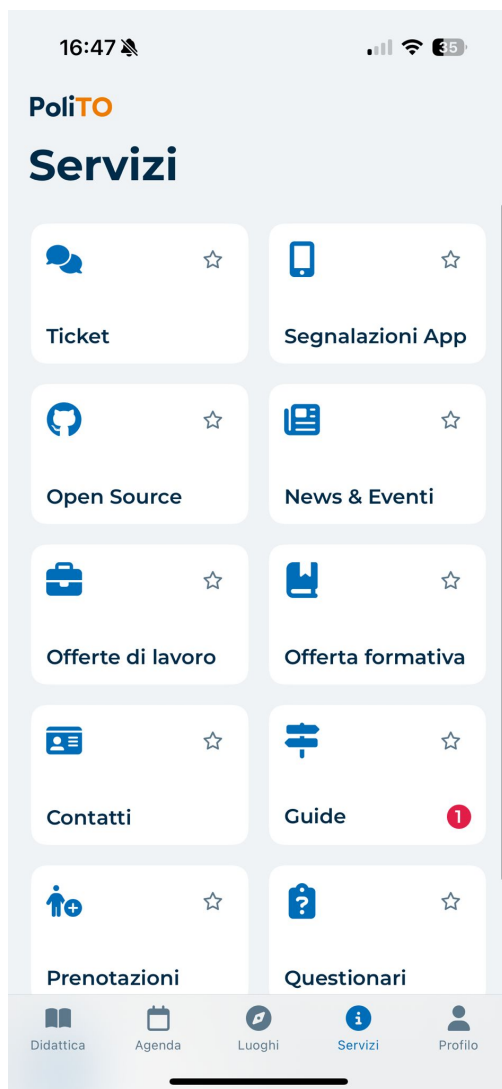
4.1 Scelte di design visuale e stile grafico

Il prototipo *mid-fidelity* dell'applicazione **PoliTO Faculty** è stato sviluppato con un focus primario sulla progettazione funzionale dell'interfaccia, in quanto lo stile visivo era già vincolato dalle linee guida esistenti dell'ecosistema digitale del Politecnico di Torino.

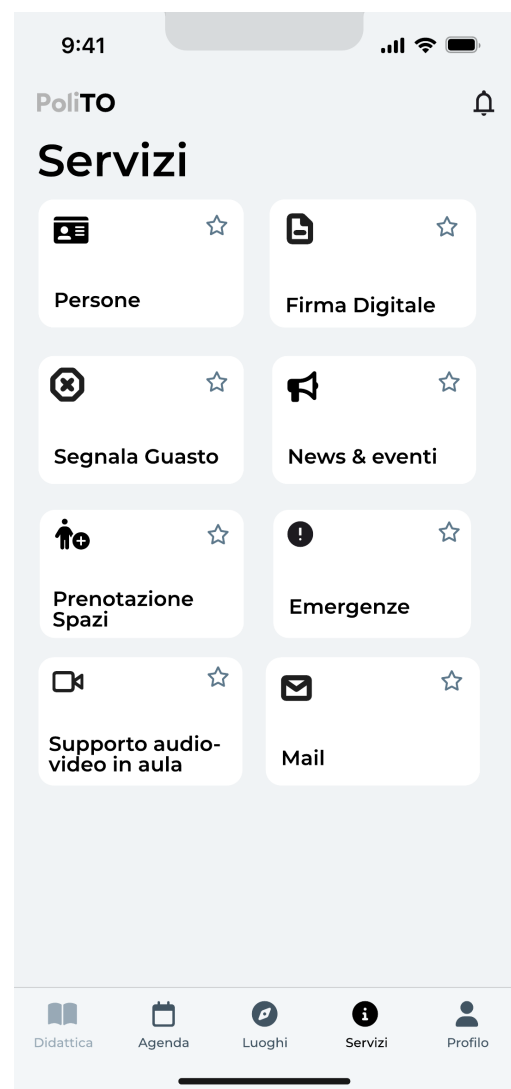
In particolare, il progetto ha adottato fin da subito i principi grafici e tipografici dell'app ufficiale per studenti, già consolidati e riconosciuti a livello istituzionale. Questo ha reso non necessario un processo di definizione stilistica ex novo, ma ha richiesto piuttosto un'attenta integrazione e adattamento del layout funzionale ai componenti esistenti.

4.1.1 Coerenza con il sistema visivo PoliTO

L'interfaccia è stata costruita seguendo fedelmente lo stile visivo dell'applicazione studenti, mantenendo così coerenza grafica e funzionale all'interno dell'ecosistema digitale dell'Ateneo. In particolare, sono stati utilizzati lo stesso font istituzionale e la relativa gerarchia testuale, nonché strutture di layout già collaudate in contesti simili. Inoltre, l'intero progetto si è basato sull'impiego dei componenti standard previsti dalla libreria UI ufficiale del Politecnico, come card, elenchi, tab e pulsanti, assicurando così una maggiore uniformità tra le diverse applicazioni mobile dell'Ateneo.

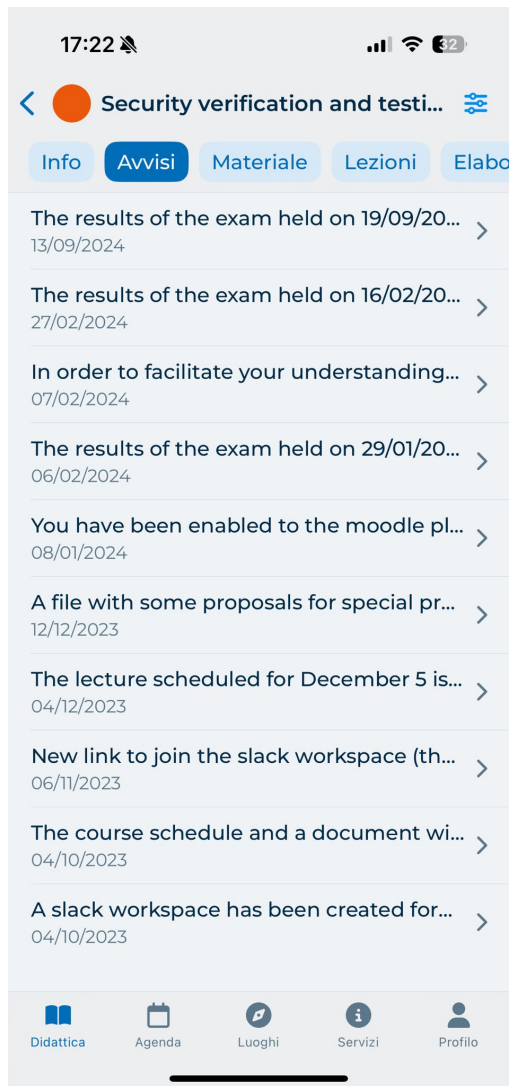


(a) Applicazione PoliTO Students – schermata servizi



(b) Prototipo PoliTO Faculty – schermata servizi

Figura 4.1: Confronto tra la visualizzazione dei servizi nelle due applicazioni.



(a) Applicazione PoliTO Students – schermata avvisi



(b) Prototipo PoliTO Faculty – schermata avvisi

Figura 4.2: Confronto tra la sezione avvisi nelle due applicazioni

4.1.2 Focus sulla funzionalità

Poiché la componente estetica era già definita dalle linee guida grafiche dell'Ateneo e molti componenti erano già disponibili nella libreria ufficiale, la progettazione del prototipo si è concentrata esclusivamente sugli aspetti funzionali. In particolare, l'attenzione si è rivolta all'organizzazione logica delle schermate, alla distribuzione dei contenuti e alla definizione dei flussi di interazione tra le varie sezioni.

Questi tre aspetti sono stati scelti perché rappresentavano il vero obiettivo della fase di prototipazione: realizzare un *mid-fidelity prototype* in grado di testare l'efficacia e l'intuitività delle funzionalità previste, senza soffermarsi su dettagli grafici o stilistici già consolidati. L'intento era verificare, attraverso la navigazione, se le funzioni richieste dai docenti — come la gestione degli appelli (RF1), la consultazione del calendario e dell'agenda (RF2), l'accesso ai dati degli studenti (RF4), la visualizzazione degli incarichi (RF3) o l'invio di avvisi (RF8) — potessero essere disposte e strutturate in modo coerente, accessibile e facilmente navigabile (RNF1, RNF4, RNF5).

Questa attenzione alla struttura funzionale e all'esperienza utente ha permesso di mantenere una forte coerenza con i requisiti qualitativi previsti, assicurando al tempo stesso affidabilità nei flussi (RNF2), compatibilità con i vincoli progettuali dell'Ateneo (RNF8) e un'organizzazione scalabile utile anche in fase di sviluppo (RNF6).

4.1.3 Scelte di prototipazione

Essendo un prototipo a fedeltà intermedia, non sono stati inseriti elementi decorativi, colori o stili avanzati: l'obiettivo era infatti quello di testare il comportamento funzionale dell'interfaccia in scenari realistici, ma ancora neutri dal punto di vista estetico. La progettazione si è quindi concentrata su aspetti come la dimensione dei componenti, la logica di disposizione dei contenuti e la definizione delle interazioni previste (come tap, scroll e navigazione tra schermate), aspetti fondamentali per validare il comportamento atteso di funzionalità centrali come l'agenda (RF2), la firma digitale (RF6), l'accesso ai corsi (RF3), o l'utilizzo dei servizi istituzionali (RF10).

Questa scelta progettuale ha garantito non solo una maggiore efficacia nei test con utenti reali, ma ha anche rispettato vincoli di usabilità (RNF1), affidabilità (RNF2), manutenibilità (RNF6) e coerenza con le linee guida grafiche ufficiali (RNF8), rendendo il prototipo un valido punto di partenza per lo sviluppo successivo.

4.2 Architettura dell'informazione

La progettazione dell'architettura dell'informazione si è concentrata sulla definizione di una struttura chiara, gerarchica e facilmente navigabile, in grado di riflettere le reali esigenze operative del personale docente. L'organizzazione dei contenuti ha seguito un criterio tematico-funzionale, finalizzato a garantire un accesso rapido alle funzionalità più utilizzate e a facilitare l'apprendimento dell'interfaccia da parte di utenti non esperti.

L'interfaccia si basa su una struttura a cinque tab principali, posizionati persistentemente nella parte inferiore dello schermo: **Didattica**, **Agenda**, **Luoghi**, **Servizi** e **Profilo**. Ciascun tab raccoglie funzionalità omogenee e presenta, al suo interno, un'ulteriore articolazione in sottosezioni.

- **Didattica**: include l'accesso ai corsi assegnati al docente attraverso due voci principali, *I miei corsi* e *Corsi in gestione*, e consente la consultazione degli *appelli*. Ogni corso apre una sezione dedicata, dalla quale è possibile gestire avvisi, materiale, lezioni, studenti ed elaborati.
- **Agenda**: consente la visualizzazione degli impegni didattici e istituzionali tramite due layout (giornaliero e settimanale). Ogni appuntamento può essere esplorato per ottenere informazioni dettagliate.
- **Luoghi**: offre una panoramica degli spazi del campus con accesso a mappe, localizzazione degli uffici e consultazione della posizione delle aule.
- **Servizi**: raggruppa le funzionalità trasversali più richieste dal personale docente, tra cui *firma digitale*, *mail istituzionale*, *prenotazione spazi*, *segnalazione guasti*, *emergenze*, *news & eventi*, *supporto audio-video* e *rubrica del personale*.
- **Profilo**: permette di accedere alle informazioni personali, modificare le impostazioni dell'applicazione e disconnettersi in sicurezza.

L'architettura è stata pensata per supportare anche percorsi alternativi all'interno dell'interfaccia (scorciatoie, notifiche contestuali, link secondari), al fine di migliorare l'usabilità e offrire maggiore flessibilità d'uso.

La Figura 4.3 mostra graficamente la struttura informativa del prototipo *PoliTO Faculty*, rappresentando in modo sintetico i principali flussi di accesso e la suddivisione tematica dei contenuti.

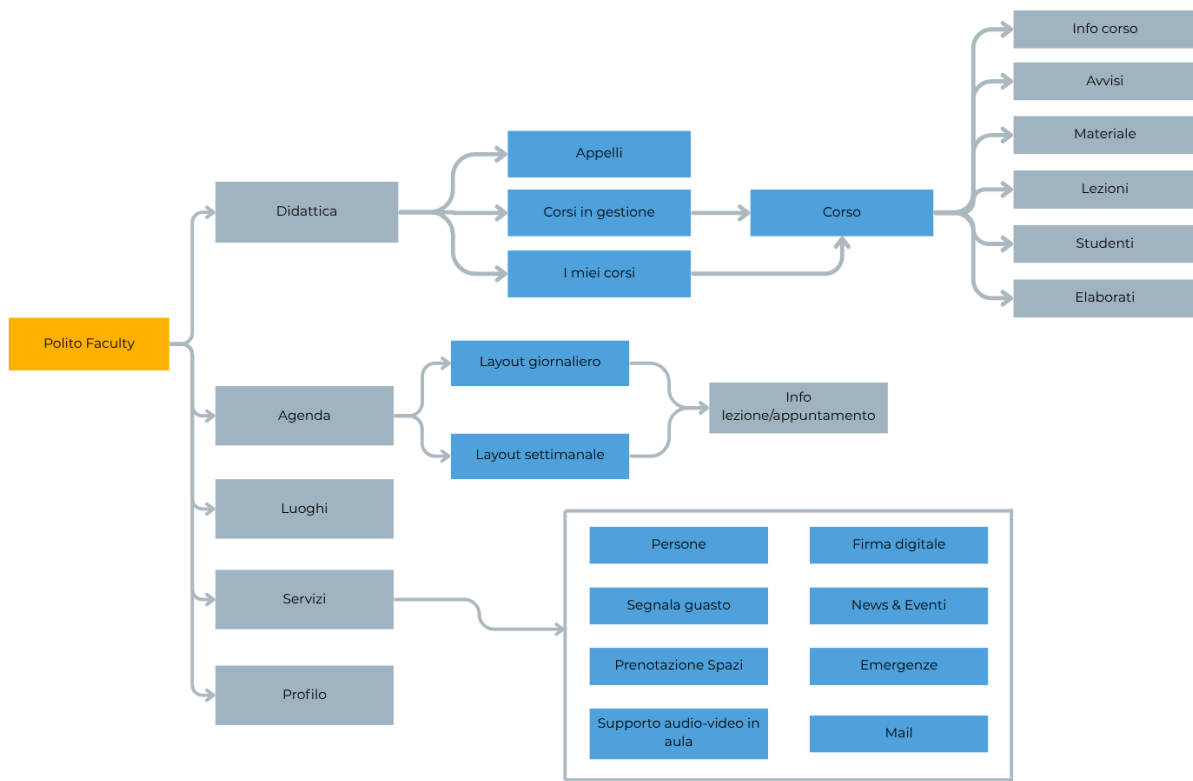


Figura 4.3: Architettura dell'informazione del prototipo PoliTO Faculty.

4.3 Prototipazione con Figma

Il prototipo *mid-fidelity* è stato realizzato utilizzando lo strumento **Figma**, selezionato per le sue caratteristiche collaborative, la gestione avanzata dei componenti e la possibilità di simulare flussi interattivi. L'interfaccia è stata costruita interamente con componenti *wireframe* e uno stile neutro, concentrandosi sull'organizzazione funzionale e logica delle schermate, senza inserire colori, icone dettagliate o elementi estetici superflui.

L'obiettivo della prototipazione era quello di validare i percorsi di navigazione e le funzionalità principali, raccogliendo feedback tempestivi prima dell'implementazione frontend.

4.3.1 Descrizione generale dei servizi progettati

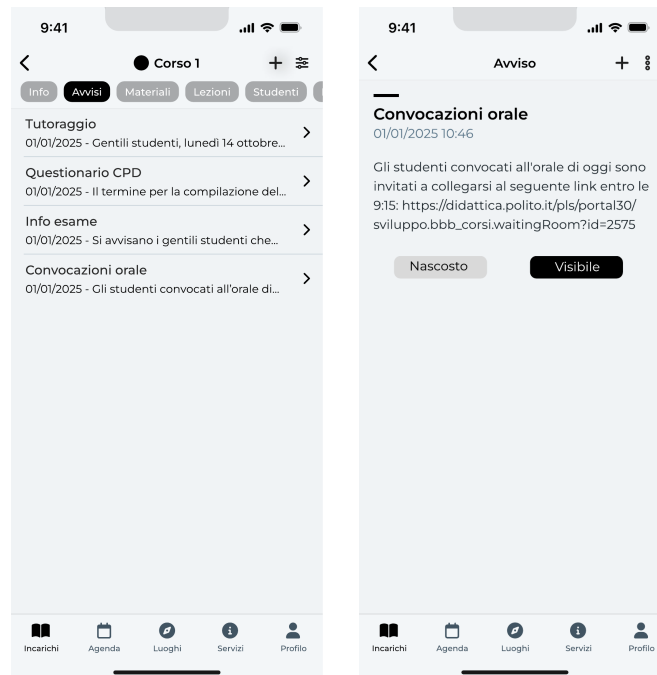
Il prototipo ha previsto la realizzazione delle schermate e dei flussi legati alle funzionalità prioritarie emerse dall'analisi dei requisiti. Sono state modellate sezioni come la didattica, con accesso ai corsi, gestione del materiale, degli studenti e degli avvisi (RF3, RF4, RF7, RF8), la gestione degli appelli (RF1), l'agenda visualizzabile in modalità giornaliera e settimanale (RF2), l'accesso ai luoghi e alla mappa del campus (RF5), l'utilizzo di servizi trasversali come firma digitale, posta istituzionale, prenotazione di spazi e segnalazione guasti (RF5, RF6, RF9), e infine la sezione del profilo personale (RF10).

Tutte queste scelte sono state effettuate con l'obiettivo di soddisfare in modo coerente le priorità funzionali espresse dagli utenti, ma anche di garantire un'esperienza stabile, fluida e accessibile (RNF1, RNF2, RNF4, RNF5), secondo una logica che favorisce la scalabilità del progetto e la successiva implementazione tecnica (RNF6, RNF7).

4.3.2 Rappresentazione dei servizi nel prototipo

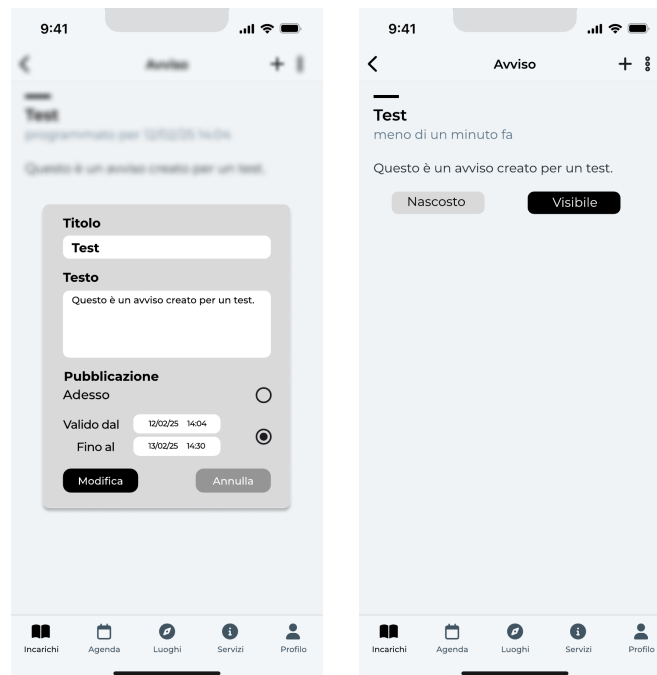
Oltre alla progettazione statica delle schermate, Figma ha consentito di creare un prototipo dinamico e interattivo, in grado di simulare realisticamente le azioni dell'utente. Ogni sezione dell'app è stata modellata con percorsi che riflettono l'uso quotidiano da parte di un docente, permettendo test di usabilità più efficaci.

Per illustrare questo approccio, di seguito viene riportato un esempio significativo: il flusso di **modifica di un avviso corso**.



(a) Accesso alla sezione *Avvisi*

(b) Apertura dell'avviso da modificare



(c) Editing del contenuto dell'avviso

(d) Salvataggio delle modifiche

Figura 4.4: Flusso interattivo di modifica di un avviso all'interno di un corso.

Questo flusso mostra l'interazione tipica che un docente potrebbe compiere per aggiornare una comunicazione rivolta agli studenti. Il prototipo simula il tap sui componenti interattivi, l'apertura di una card esistente, l'editing del testo e la conferma della modifica.

Analogamente, sono stati modellati altri flussi fondamentali: esplorazione dell'agenda, gestione degli elaborati, visualizzazione dei profili studente, navigazione tra sezioni della didattica.

L'obiettivo è stato fornire una base solida e realistica su cui iterare prima dello sviluppo frontend, riducendo gli errori progettuali e anticipando il comportamento dell'interfaccia.

Capitolo 5

Sviluppo dell'applicazione

Dopo aver validato i flussi principali e definito l'architettura dell'informazione, è stato possibile avviare la fase di sviluppo frontend dell'applicazione **PoliTO Faculty**. L'obiettivo era tradurre il prototipo *mid-fidelity* in un'interfaccia funzionante, mantenendo coerenza visiva con l'ecosistema PoliTO e assicurando una navigazione fluida tra le sezioni principali.

Il lavoro è stato svolto in modalità iterativa, con particolare attenzione all'organizzazione del codice, alla gestione modulare dei componenti e all'aderenza alle specifiche progettuali definite nelle fasi precedenti.

5.1 Scelte tecnologiche

Per la realizzazione dell'interfaccia è stato scelto il framework **React Native** [3], in quanto consente lo sviluppo di applicazioni mobile multiplatforma (iOS e Android) da un'unica base di codice. La scelta è motivata da:

- **Compatibilità con l'ecosistema esistente:** PoliTO utilizza già React Native per altre applicazioni ufficiali.
- **Rapidità di sviluppo:** la componente riusabile e il supporto a librerie consolidate riducono i tempi di implementazione.
- **Community attiva e ricco ecosistema:** semplifica l'integrazione di plugin per navigazione, gestione stato e simulazione dati.

Per lo styling è stato adottato **TailwindCSS** tramite la libreria **NativeWind**, che consente di definire le classi stile direttamente all'interno dei componenti, mantenendo leggibilità e coerenza con lo stile PoliTO.

5.2 Architettura del frontend

L'architettura del frontend è stata progettata per garantire modularità, chiarezza organizzativa e facilità di manutenzione, seguendo un approccio scalabile e coerente con le linee guida dell'applicazione PoliTO Students.

L'interfaccia è suddivisa in più livelli logici e cartelle tematiche:

- **core**: contiene il navigatore principale dell'applicazione, provider globali, hook condivisi e il database simulato per la gestione dei dati fittizi.
- **components**: raccoglie tutti i componenti UI riutilizzabili (card, tab, icone, pulsanti), spesso ereditati o adattati dall'app studenti.
- **tabs**: sono presenti cinque cartelle principali, una per ciascun tab di navigazione (*Didattica*, *Agenda*, *Luoghi*, *Servizi*, *Profilo*). Ogni cartella include:
 - componenti specifici di sezione (non condivisi);
 - hook e contesti locali;
 - schermate (*screens*) corrispondenti alle varie viste.

Questa suddivisione favorisce una separazione netta tra logica condivisa e logica locale alle singole funzionalità.

5.2.1 Gestione della navigazione

La navigazione dell'app è stata gestita tramite la libreria **React Navigation**, che ha consentito la creazione di un sistema multilivello basato su:

- un **tab navigator principale**, che rappresenta le cinque sezioni principali dell'app;
- **stack navigator annidati** per gestire la navigazione interna alle sottosezioni;
- la possibilità di passaggio parametri tra schermate (es. ID corso, ID avviso) per accesso contestuale.

Sono state abilitate transizioni fluide e una gestione automatica della navigazione condizionata, ad esempio tramite notifiche o scorciatoie.

5.2.2 Gestione dei dati fittizi

Non essendo ancora disponibile un backend reale, è stato progettato un database fittizio, localizzato nella directory **core**. L'intero sistema è stato costruito e definito nel file **CoursesContext.tsx**, che funge da contesto centrale per la gestione dei dati. Al suo interno sono stati inizializzati mock data realistici, accessibili da tutte le schermate dell'applicazione. Questo approccio ha permesso di simulare un comportamento credibile, coerente con l'utilizzo reale dell'applicazione.

Inoltre, il contesto espone una serie di metodi per manipolare dinamicamente le informazioni, come ad esempio `addNotice`, `removeNotice`, `addLesson`, e altri ancora. Tali metodi consentono di eseguire tutte le principali operazioni di tipo CRUD (Create, Read, Update, Delete) in modo reattivo.

Per accedere facilmente a questi dati e metodi da qualsiasi componente, sono stati implementati appositi `hook` personalizzati, tra cui `useCoursesContext`. L'uso combinato di `context` e `hook` ha permesso di simulare con efficacia il flusso di dati e le interazioni tipiche di un'applicazione connessa a un server, rendendo possibile il test del prototipo anche in assenza di una vera integrazione lato backend.

5.2.3 Lingua e tema

La gestione della lingua e del tema visivo dell'applicazione è centralizzata tramite il contesto `PreferencesContext`, che conserva in memoria le preferenze utente. Per quanto riguarda la lingua dell'interfaccia, l'app supporta sia l'italiano che l'inglese, grazie all'integrazione della libreria `i18next`. Il sistema è progettato per adattarsi dinamicamente alla scelta dell'utente, consentendo così una fruizione più accessibile e inclusiva.

Analogamente, anche il tema visivo dell'interfaccia può essere personalizzato scegliendo tra modalità chiara (`light`), scura (`dark`) oppure adattata automaticamente alle impostazioni di sistema (`system`). Queste preferenze vengono poi propagate coerentemente a tutti i componenti dell'interfaccia tramite `hook` dedicati come `useTheme` e `useTranslation`. Questo approccio garantisce non solo uniformità visiva, ma anche un'esperienza coerente e accessibile a livello linguistico.

5.3 Scelte stilistiche e UI

L'interfaccia dell'applicazione **PoliTO Faculty** è stata progettata nel rispetto delle linee guida visive istituzionali del Politecnico di Torino, già adottate nell'app ufficiale per studenti [4]. Questa scelta ha garantito continuità e riconoscibilità dell'identità visiva all'interno dell'ecosistema digitale d'Ateneo.

La base stilistica è stata interamente ereditata dall'app *PoliTO Students*, senza introdurre variazioni ai colori, alla tipografia o alle dimensioni dei testi. L'utilizzo di componenti già esistenti ha permesso di accelerare lo sviluppo mantenendo coerenza e qualità visiva.

5.3.1 Adattamenti e personalizzazioni

Sebbene lo stile visivo sia rimasto invariato, sono stati effettuati diversi adattamenti a livello funzionale. In particolare, alcuni componenti dell'app studenti sono stati modificati per rispondere a esigenze specifiche dei docenti.

Un esempio significativo è il componente per la navigazione a tab all'interno dei corsi, arricchito con una nuova scheda "Studenti" e con il supporto a un *pulsante dinamico* per l'aggiunta contestuale di contenuti (avvisi, file, lezioni). Questo è stato realizzato integrando logiche condizionali sullo stato della navigazione, modificando la visibilità e il comportamento del pulsante in base alla sezione attiva.

Poiché l'app studenti non prevede funzionalità di creazione o modifica di contenuti, alcuni componenti – come pulsanti, moduli o sezioni dinamiche – sono stati adattati per consentire al docente di gestire in autonomia avvisi, materiale didattico e altri contenuti del corso, mantenendo comunque lo stile visivo originale.

5.3.2 Coerenza e leggibilità

L'app mantiene una struttura gerarchica chiara e una gerarchia visiva ben definita, grazie all'uso dei medesimi font istituzionali e di un layout responsive che si adatta a dispositivi con dimensioni differenti. Gli elementi chiave (titoli, pulsanti, icone) risultano immediatamente riconoscibili e leggibili, contribuendo a un'interazione intuitiva anche per utenti meno esperti.

5.3.3 Uso di NativeWind per lo stile

Per la definizione dello stile è stata utilizzata la libreria **NativeWind**, che consente di scrivere classi in stile **TailwindCSS** direttamente nei componenti React Native. Questo ha permesso di:

- mantenere coerenza con la logica funzionale e dichiarativa del progetto;
- evitare file di stile esterni, mantenendo lo stile vicino al markup;
- riutilizzare facilmente pattern stilistici comuni tra componenti.

In sintesi, le scelte stilistiche adottate hanno puntato a valorizzare la continuità con l'ecosistema PoliTO, assicurando un'interfaccia visivamente coerente, accessibile e ben strutturata.

Capitolo 6

Implementazione dei servizi e componenti

In questo capitolo viene descritto nel dettaglio lo sviluppo delle principali sezioni dell'applicazione **PoliTO Faculty**, con un focus specifico sui singoli tab funzionali introdotti nella fase di progettazione.

Per ciascuna sezione (*Profilo*, *Servizi*, *Luoghi*, *Agenda*, *Didattica*) verranno analizzati:

- le **funzionalità** implementate;
- l'**architettura dei componenti** sviluppati;
- le **componenti** adottati;
- gli eventuali **problemi riscontrati** in fase di sviluppo e le **soluzioni implementate**.

L'obiettivo è illustrare come le funzionalità definite nel prototipo *mid-fidelity* siano state tradotte in componenti effettivamente funzionanti e coerenti con le linee guida istituzionali.

Ogni paragrafo fornisce inoltre esempi concreti, porzioni di codice e schermate esplicative, al fine di documentare in modo trasparente l'approccio tecnico e progettuale adottato.

6.1 Profilo

La sezione **Profilo** consente al docente di consultare e gestire le proprie informazioni personali, contatti e impostazioni. Essa è composta da tre schermate principali: *Profilo*, *Modifica Profilo* e *Impostazioni*, collegate tra loro tramite un navigatore dedicato chiamato **ProfileNavigator**.

6.1.1 Funzionalità implementate

Nella schermata principale dell'applicazione, denominata **ProfileScreen**, l'utente ha accesso a una serie di informazioni personali e istituzionali. In primo luogo è visibile la propria **smartcard universitaria**, utile per identificarsi all'interno del sistema PoliTO. Sono poi consultabili le **informazioni personali**, tra cui la residenza, la residenza fiscale e il codice IBAN. Infine, l'utente può accedere ai propri **contatti**, che includono recapiti telefonici e indirizzi email istituzionali e privati. Questa schermata rappresenta un punto di accesso centralizzato e immediato alle informazioni più rilevanti per il docente.

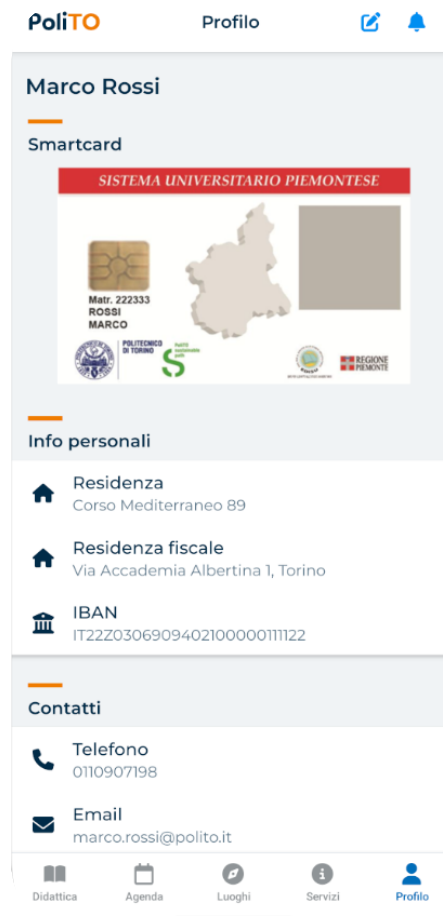
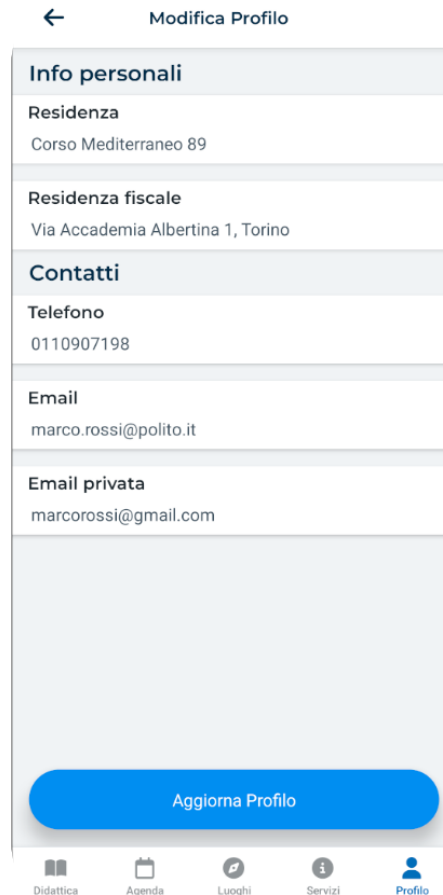


Figura 6.1: Schermata principale del profilo con smartcard e dati personali.

L'utente ha inoltre la possibilità di modificare i propri dati personali. Cliccando sull'icona di modifica presente in alto a destra nella schermata del profilo, viene reindirizzato alla schermata *Modifica Profilo* (denominata `ProfileForm` nel progetto). In questa sezione può aggiornare campi come la **residenza** e la **residenza fiscale**, il **numero di telefono**, nonché l'indirizzo **email istituzionale** e **privata**. Questa funzionalità consente agli utenti di mantenere sempre aggiornate le proprie informazioni anagrafiche all'interno dell'ecosistema digitale dell'Ateneo.



← Modifica Profilo

Info personali

Residenza
Corso Mediterraneo 89

Residenza fiscale
Via Accademia Albertina 1, Torino

Contatti

Telefono
0110907198

Email
marco.rossi@polito.it

Email privata
marcorossi@gmail.com

Aggiorna Profilo

Didattica Agenda Luoghi Servizi **Profilo**

Figura 6.2: Form di modifica delle informazioni personali.

Infine, tramite l'apposita schermata delle **Impostazioni** (chiamata `SettingsScreen`)| è possibile cambiare il **tema** dell'interfaccia, scegliendo tra modalità chiara, scura o sistema; modificare la **lingua** selezionando tra italiano e inglese; e accedere all'opzione di rimozione della **cache dei file di corso**, funzionalità al momento non ancora implementata.

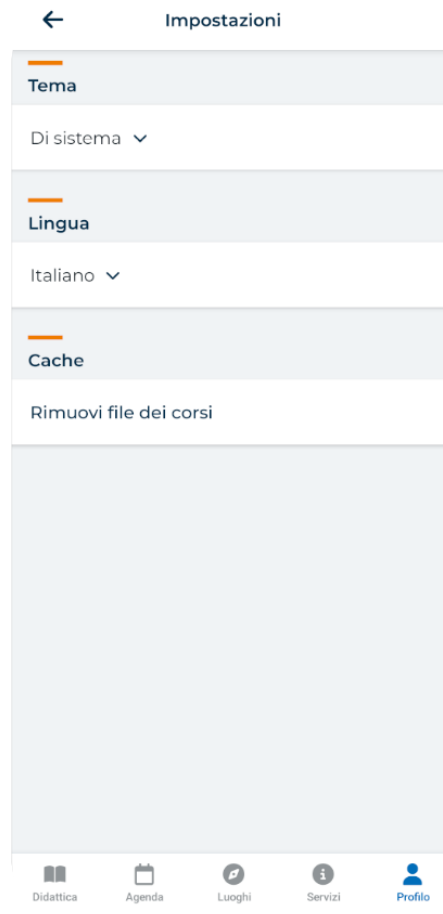


Figura 6.3: Schermata delle impostazioni del profilo.

6.1.2 Architettura del componente

La sezione **Profilo** è gestita tramite un navigatore dedicato denominato `ProfileNavigator`, che collega tra loro le schermate:

- `ProfileScreen` — schermata principale con i dati del docente;
- `ProfileForm` — schermata per la modifica dei dati;
- `SettingsScreen` — schermata per le impostazioni dell'interfaccia.

Tutte le schermate sono state progettate secondo un'architettura modulare e riutilizzano componenti esistenti dell'app studenti, opportunamente adattati per il contesto docente. In particolare, la schermata `ProfileForm` è stata sviluppata ex novo, poiché le funzionalità di editing non erano presenti nella versione studenti.

6.1.3 Componenti adottati

L'interfaccia utente della sezione **Profilo** è stata progettata per garantire semplicità e coerenza visiva con l'app PoliTO Students. I principali componenti UI adottati includono:

- `SectionHeader` — utilizzato per separare logicamente le sezioni del profilo. A differenza dell'app studenti, in questa sezione può includere link per accedere ad altre schermate (es. `StaffScreen` nella didattica).



Figura 6.4: Componente `SectionHeader` con titolo e azione.

- `ListItem` — usato in modo uniforme per visualizzare dati strutturati (es. residenza, IBAN).

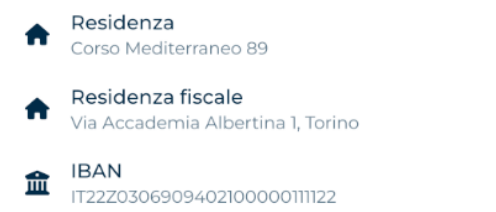


Figura 6.5: Esempio di `ListItem`.

- **IconButton** — pulsanti con icone per azioni rapide come modifica profilo o apertura notifiche.



Figura 6.6: IconButton in alto a destra.

- **Card**, **TextInput**, **CtaButton** — componenti usati nel form di modifica, derivati dalla struttura dell'app studenti.
- **Select** — utilizzato nelle impostazioni per cambiare lingua e tema.

Tutti questi componenti condividono il font istituzionale del PoliTO, una palette colori coerente e interazioni omogenee con feedback visivo.

6.2 Didattica

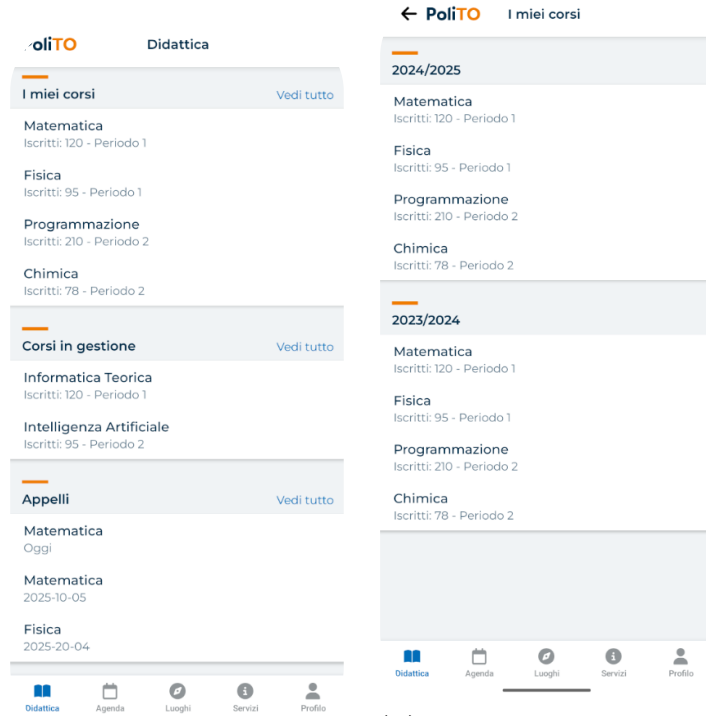
La sezione **Didattica** costituisce il cuore funzionale dell'applicazione **PoliTO Faculty**. Rispetto alle altre aree, presenta la maggiore complessità in termini di struttura, interazioni e componenti sviluppati, poiché concentra tutte le funzionalità legate alla gestione didattica del docente, dalla consultazione dei corsi fino alla gestione di avvisi, materiali e studenti. È stata progettata con l'obiettivo di offrire un'interfaccia modulare e coerente con le esigenze concrete dei docenti, pur mantenendo un elevato livello di coerenza visiva e architettonica con l'applicazione studenti.

6.2.1 Funzionalità implementate

La schermata principale della sezione si suddivide in tre blocchi informativi:

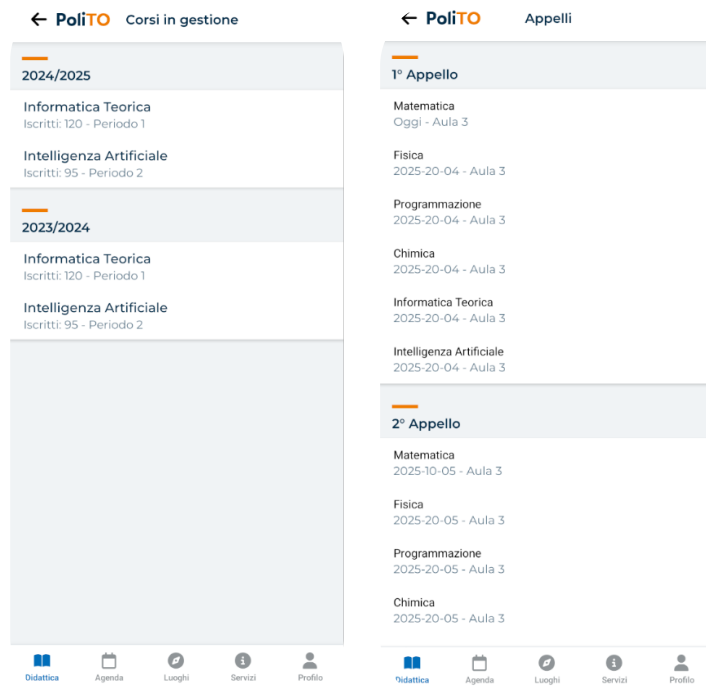
- **I miei corsi** — Elenco dei corsi in cui il docente è titolare.
- **Corsi in gestione** — Elenco dei corsi dove il docente ricopre un ruolo di supporto.
- **Appelli** — Visualizzazione degli esami imminenti relativi ai propri corsi.

Ogni sezione visualizza al massimo 4 elementi. Per consultare la lista completa è presente un link “Vedi tutto” nel **SectionHeader**, che rimanda a schermate specifiche per ogni categoria.



(a) Pagina principale

(b) Schermata “I miei corsi”



(c) Corsi in gestione

(d) Appelli dei corsi

Figura 6.7: Navigazione iniziale della sezione Didattica.

All'interno di ciascun corso selezionato, l'utente accede a una schermata organizzata tramite una barra di navigazione superiore a tab. Ogni tab rappresenta una sezione tematica autonoma del corso.

- **Info** – Contiene le informazioni generali sul corso: periodo, CFU, collaboratori e loro permessi, prossimi appelli, link alla guida del corso.
- **Avvisi** – Permette di creare, modificare e gestire avvisi pubblicati agli studenti.
- **Materiale** – Sezione per il caricamento, l'organizzazione in cartelle e la gestione dei materiali didattici.
- **Lezioni** – Registro delle lezioni tenute, con possibilità di specificare data, argomento, aula, staff e lingua.
- **Studenti** – Consente la visualizzazione, ricerca, filtro e gestione degli studenti iscritti al corso.
- **Elaborati** – Permette di consultare e scaricare gli elaborati caricati dagli studenti.

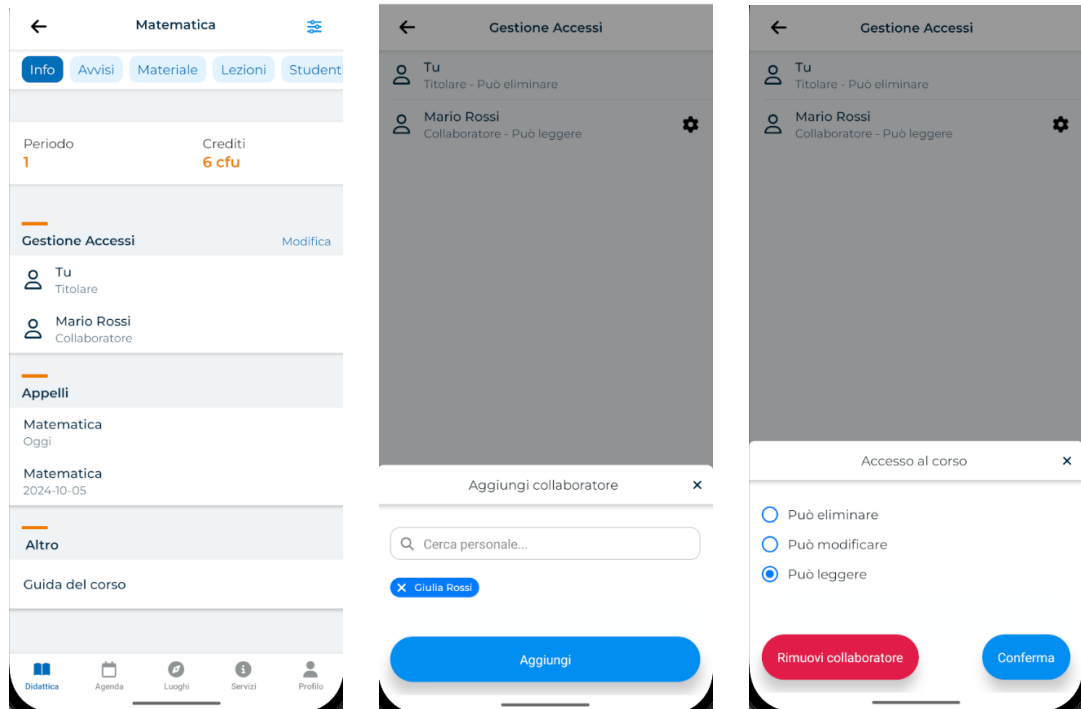
6.2.2 Architettura del componente

La sezione **Didattica** si basa su una struttura di navigazione composta da due navigatori principali:

- **TeachingNavigator**, che gestisce la navigazione generale all'interno della sezione Didattica, dalla schermata principale ai corsi specifici;
- **CourseNavigator**, attivo quando l'utente accede a un singolo corso. Utilizza un sistema a tab orizzontali, implementato tramite `createMaterialTopTabNavigator` della libreria `@react-navigation/material-top-tabs`.

Le sezioni tematiche del corso descritte in precedenza sono implementate ciascuna attraverso una tab dedicata nel **CourseNavigator**. Di seguito, per ogni tab vengono indicati i relativi componenti tecnici e lo schermo corrispondente.

Info corso — CourseInfoTab Come detto prima, questa sezione mostra le informazioni generali del corso, tra cui la possibilità di gestire i collaboratori del corso (nello schermo denominato **StaffScreen**). In particolare, è possibile aggiungere collaboratori e modificare o rimuovere i relativi permessi.



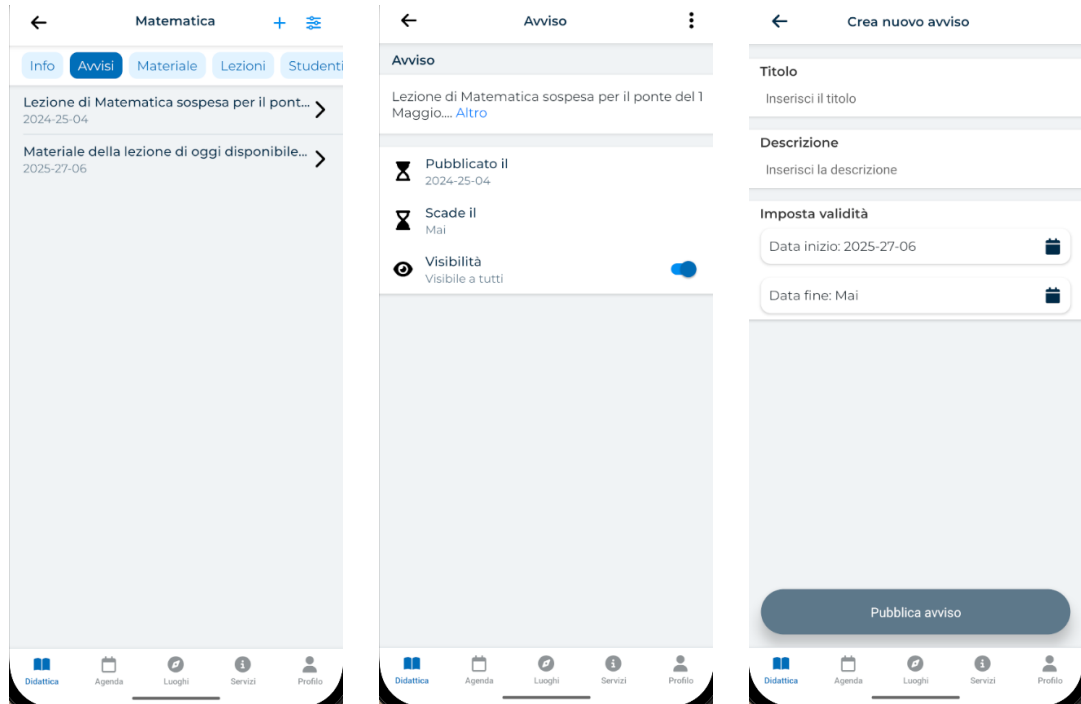
(a) Pagina principale dentro CourseInfoTab

(b) Aggiunta collaboratore dentro StaffScreen

(c) Gestione permessi dentro StaffScreen

Figura 6.8: Panoramica sulla sezione di Info Corso

Avvisi — CourseNoticesTab Permette la gestione completa degli avvisi pubblicati dal docente, inclusa la possibilità di consultare la lista degli avvisi, visualizzare nel dettaglio ciascun avviso con i relativi dati (incluso lo stato di visibilità) e accedere al form per la modifica o la creazione di un nuovo avviso.



(a) Pagina principale dentro CourseNoticesTab

(b) Avviso nel dettaglio dentro NoticeScreen

(c) Creazione di un avviso dentro NoticeFormScreen

Figura 6.9: Panoramica sulla sezione Avvisi

Materiale — CourseFilesTab La sezione dedicata alla gestione del materiale didattico offre una doppia modalità di visualizzazione: una modalità “flat”, che mostra tutti i file in un unico elenco, e una modalità organizzata per cartelle. Il materiale può essere aggiunto tramite il pulsante “+”, così come modificato o rimosso in qualsiasi momento.

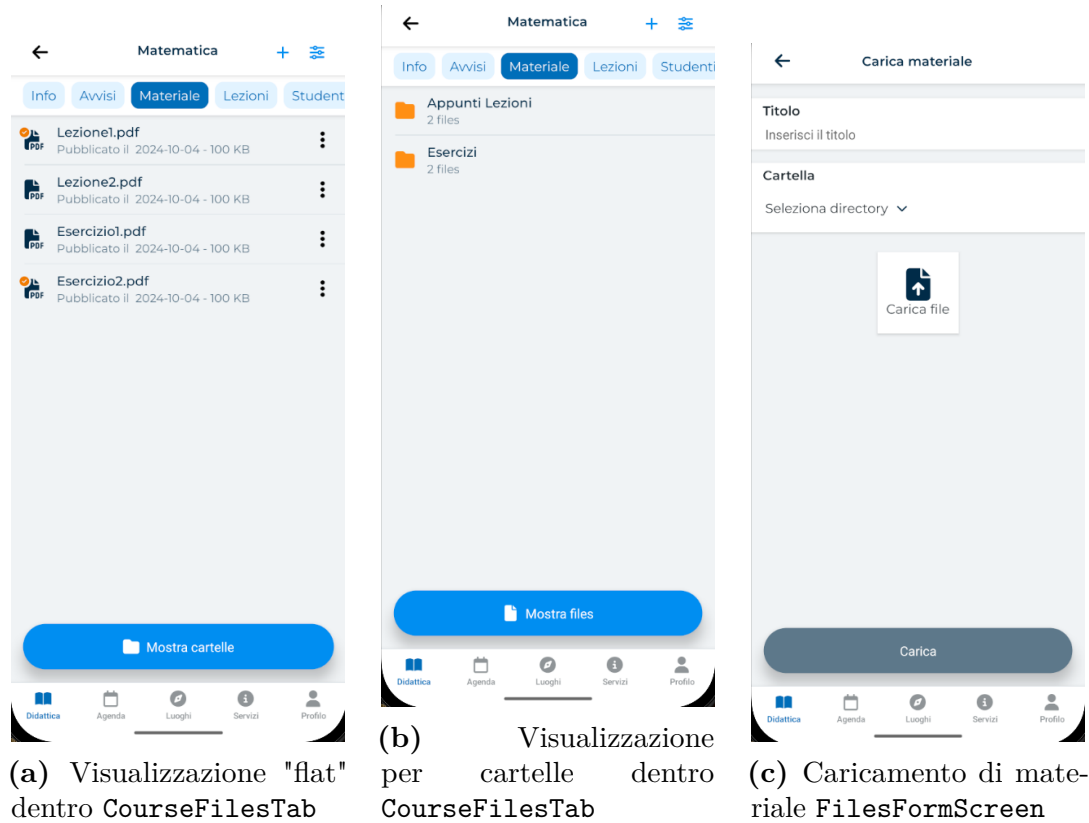
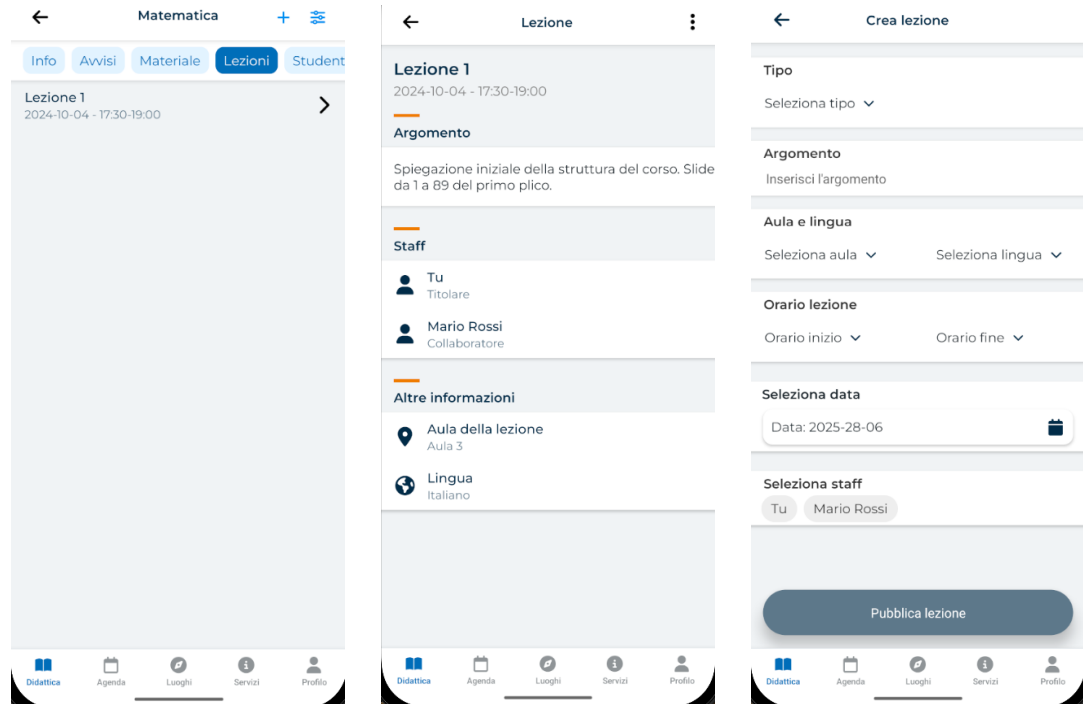


Figura 6.10: Panoramica sulla sezione Materiale

Lezioni — CourseLecturesTab Come detto precedentemente, questa sezione funziona come registro del docente. È possibile visualizzare la lista completa delle lezioni e accedere al dettaglio di ciascuna, con informazioni specifiche quali titolo, data, orario, argomento, staff coinvolto, aula assegnata e lingue. Inoltre, viene offerta la possibilità di aggiungere, modificare o eliminare le lezioni in modo flessibile.



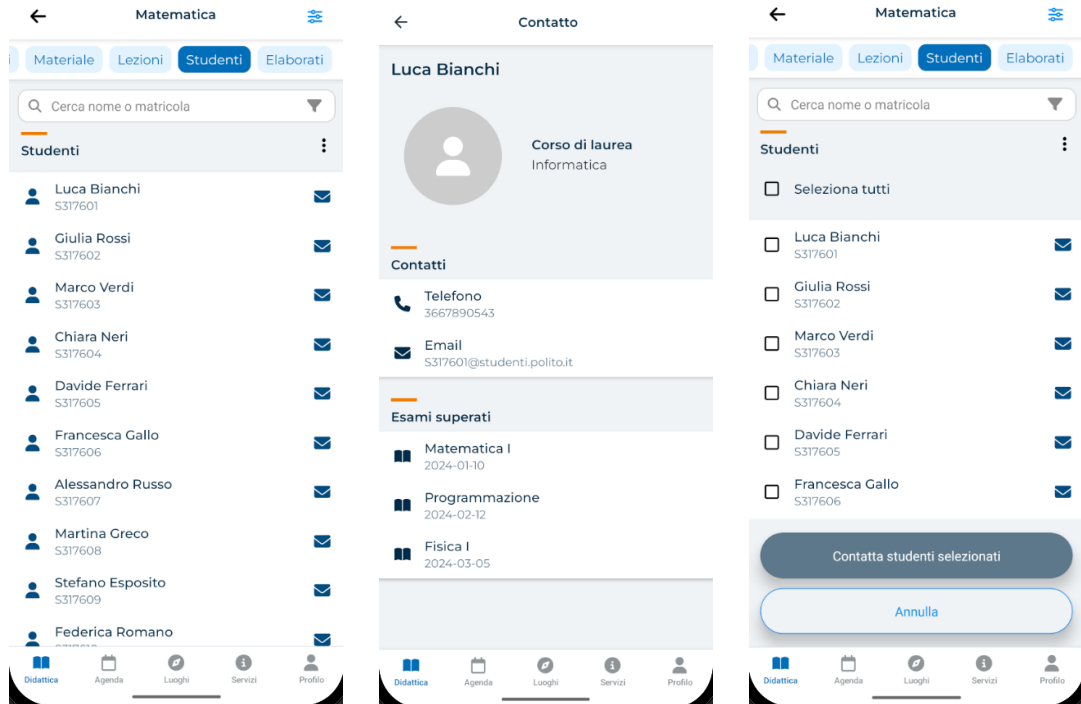
(a) Lista lezioni dentro CourseLecturesTab

(b) Dettaglio lezione dentro LectureScreen

(c) Aggiunta lezione dentro LectureFormScreen

Figura 6.11: Panoramica sulla sezione Lezioni

Studenti — CourseStudentsTab Consente la gestione avanzata degli studenti iscritti, offrendo la possibilità di visualizzare la lista completa degli studenti, consultare la scheda contatto dei singoli, contattare sia un singolo studente che un gruppo intero e aggiungere nuovi studenti all’elenco.



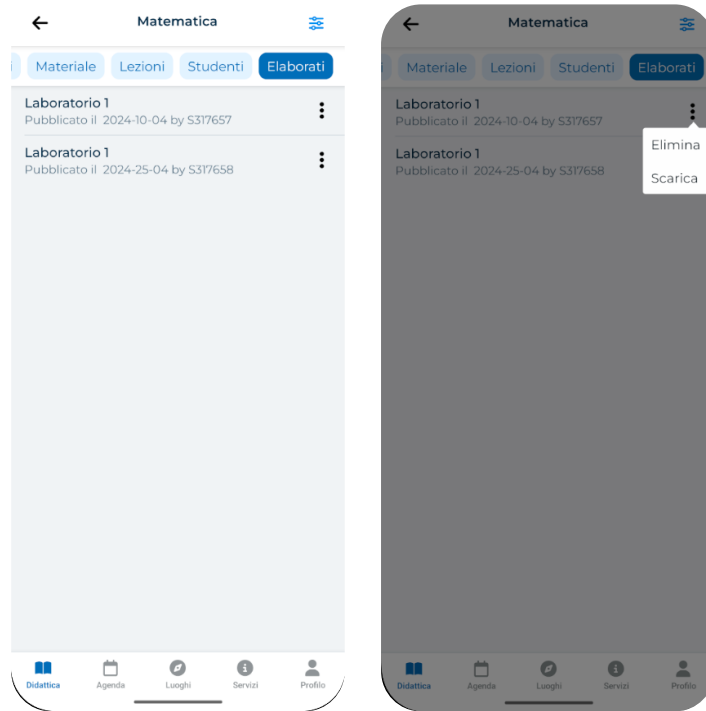
(a) Lista studenti dentro CourseStudentsTab

(b) Contatto studente dentro StudentContact

(c) Modalità di selezione studenti per contattarli

Figura 6.12: Panoramica sulla sezione Studenti

Elaborati — **CourseAssignmentsTab** Permette di consultare e scaricare gli elaborati caricati dagli studenti. È anche possibile eliminarli. Non è prevista la possibilità di aggiunta da parte del docente. s



(a) Lista elaborati dentro CourseAssignmentsTab

(b) Lista opzioni dentro CourseAssignmentsTab

Figura 6.13: Panoramica sulla sezione Elaborati

6.2.3 Componenti adottati

Anche la sezione **Didattica** segue le linee guida visive e interattive dell'app PoliTO Students, mantenendo coerenza nella tipografia, palette colori e struttura modulare. Tuttavia, per rispondere alle esigenze del ruolo docente, introduce nuove interazioni e componenti non presenti nell'app per studenti.

- **SectionHeader** — utilizzato per separare logicamente le sezioni, spesso corredato da azioni contestuali come “Vedi tutto” o “Modifica”.

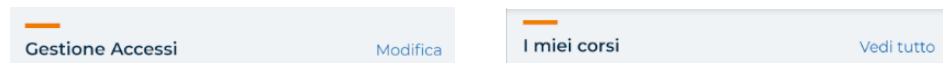


Figura 6.14: Esempi di **SectionHeader** con opzioni.

- **ListItem** — impiegato in tutte le schermate elenco per presentare in modo uniforme elementi modificabili (avvisi, materiali, lezioni, studenti...).
- **createMaterialTopTabNavigator** — libreria usata per la navigazione tra tab nelle schermate di corso (**CourseNavigator**), come mostrato in Fig. 6.15.



Figura 6.15: Navigazione a tab all’interno di un corso.

- **Popover** — introdotto per far capo alle necessità dei docenti, consente di mostrare azioni contestuali come modifica ed elimina, associate ai singoli elementi. È stato realizzato tramite la libreria `react-native-popover-view`.

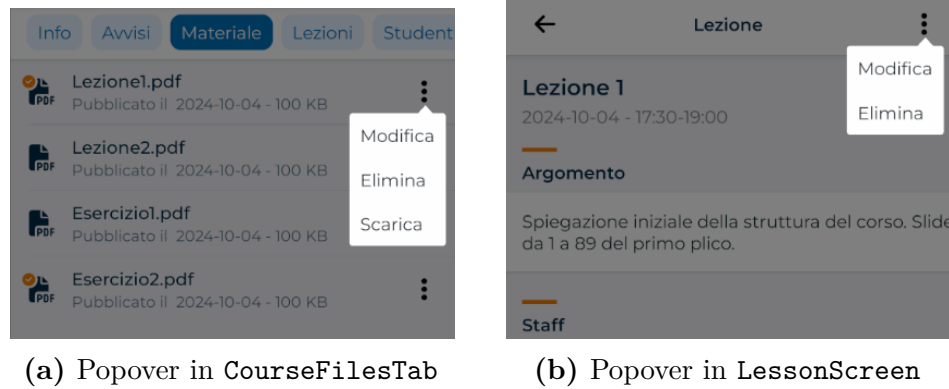


Figura 6.16: Menu contestuali visualizzati con Popover.

- **CtaButton** — usato in tutte le schermate di creazione/modifica per confermare un'azione, o anche in altre schermate per eliminare un elemento.

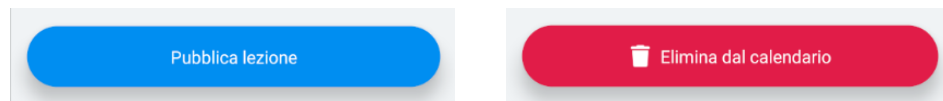


Figura 6.17: CtaButtons nel registro lezioni o nel calendario.

- **Switch** — presente negli avvisi per controllarne la visibilità agli studenti.



Figura 6.18: Esempio di utilizzo del componente Switch negli avvisi.

- **BottomModal** — utilizzato ad esempio per modificare i permessi dei collaboratori, si apre dal basso verso l'alto e permette di fare delle azioni senza dover caricare una nuova pagina.



Figura 6.19: Gestione permessi tramite BottomModal.

- **Select** — utilizzato nei form per permettere la selezione da una lista predefinita, ad esempio per scegliere lingua, ruolo o categoria durante la compilazione di avvisi, lezioni o gestione collaboratori.

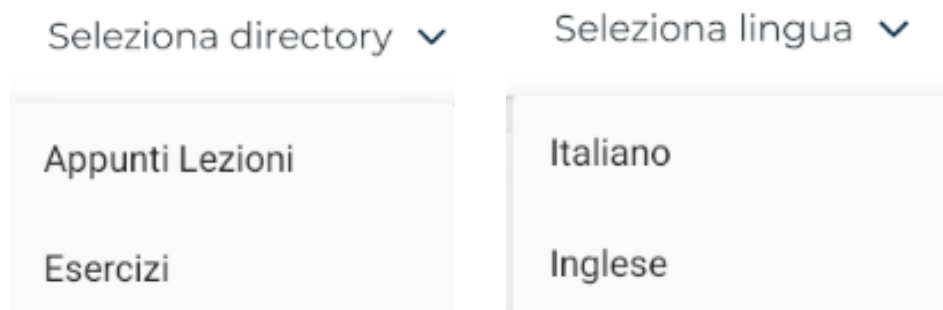


Figura 6.20: Esempio di Select nei vari form.

Nelle sezioni dove è prevista la creazione di contenuti (avvisi, materiali, lezioni, ecc.), è presente un pulsante “+” nell’header. L’app, grazie allo stack di navigazione attivo, riconosce il contesto e apre automaticamente il form corretto.

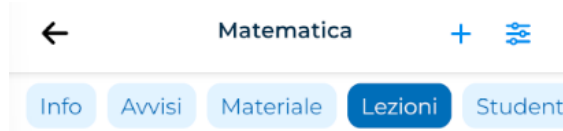


Figura 6.21: In questo caso, il pulsante “+” porta al form per la creazione di una Lezione per il registro.

6.2.4 Problemi riscontrati e soluzioni

Lo sviluppo della sezione **Didattica** ha richiesto particolare attenzione per via della sua articolazione interna e delle numerose funzionalità esclusive del contesto docente. Le principali criticità affrontate sono state:

- **Navigazione complessa** — L’architettura annidata tra **TeachingNavigator**, **CourseNavigator** e le sottoschermate interne ai singoli tab ha richiesto un’attenta gestione dello stato di navigazione per evitare conflitti e comportamenti imprevisti. Il problema è stato risolto razionalizzando le route e separando con chiarezza le responsabilità dei navigatori.
- **Gestione dinamica del pulsante “+”** — Poiché il pulsante di aggiunta è presente solo in alcune tab (es. **Materiale**, **Avvisi**, **Lezioni**, **Studenti**), è stato necessario implementare una logica centralizzata per permettere al navigatore di determinare dinamicamente quale form aprire, a seconda del contesto attivo.
- **Sezione Studenti** — L’intera tab **CourseStudentsTab** è stata sviluppata da zero, in quanto non esisteva una controparte nell’app studenti. Sono state progettate e implementate ex novo tutte le funzionalità: visualizzazione, ricerca, filtri, contatto diretto o multiplo, aggiunta manuale.

Tutti gli altri problemi minori sono stati risolti durante il normale ciclo di sviluppo e testing e non hanno avuto un impatto significativo sul progetto complessivo.

6.3 Agenda

La sezione **Agenda** consente ai docenti di consultare in modo efficace e intuitivo i propri impegni settimanali e giornalieri. Essa fornisce una vista unificata delle lezioni, degli appuntamenti personali e di altri eventi istituzionali, con l'obiettivo di migliorare la pianificazione e la gestione del tempo. Sebbene l'integrazione diretta con il calendario Outlook sia prevista nelle future implementazioni, l'attuale versione prototipale offre comunque una panoramica completa e interattiva delle attività accademiche.

6.3.1 Funzionalità implementate

All'interno della sezione **Agenda**, l'utente ha la possibilità di visualizzare tutte le **lezioni** associate ai corsi di cui è docente, insieme agli eventuali **appuntamenti** personali, che possono essere inseriti manualmente tramite un apposito form. Per ogni evento visualizzato, sono disponibili una serie di dettagli: il **titolo**, l'**orario**, la **data**, una breve **descrizione**, il **luogo** in cui si svolge e, nel caso di una lezione, anche il **corso associato**.

L'interfaccia permette di alternare tra due modalità di visualizzazione:

- **Layout settimanale** – per una panoramica degli impegni dell'intera settimana;
- **Layout giornaliero** – per concentrarsi su un singolo giorno.

In entrambe le modalità, ogni evento è selezionabile per accedere a una vista dettagliata. Nel caso in cui il luogo dell'evento corrisponda a una struttura dell'ateneo (es. un'aula), è possibile, tramite l'interazione con il `ListItem`, essere reindirizzati automaticamente alla mappa della sezione **Luoghi**, con le indicazioni per raggiungerla. Allo stesso modo, se l'evento è una lezione associata a un corso, il `ListItem` consente di accedere rapidamente alla pagina del corso corrispondente nella sezione **Didattica**.

In alto a destra è presente il consueto pulsante “+”, che consente di accedere a un form dedicato per l'aggiunta di un nuovo appuntamento. Le lezioni invece vengono mostrate automaticamente in base ai dati dei corsi, e possono essere rimosse ma non create direttamente da questa sezione.

6.3.2 Architettura del componente

La sezione **Agenda** è gestita dal navigatore dedicato `AgendaNavigator`, che controlla la navigazione tra le due modalità di visualizzazione, giornaliera e settimanale, e consente l'accesso al dettaglio dei singoli eventi. La struttura di base e la logica di rendering sono derivate direttamente dall'agenda dell'app PoliTO Students, ma sono state introdotte modifiche specifiche per adattare la funzionalità alle esigenze del personale docente. Tra queste, figurano l'integrazione di un **form per la creazione** di nuovi appuntamenti e la possibilità di **eliminare** una lezione direttamente dal calendario.

I principali screen che compongono la sezione sono:

AgendaScreen Rappresenta la visualizzazione giornaliera dell'agenda. Consente di vedere nel dettaglio tutti gli eventi della giornata, con possibilità di selezione e interazione.

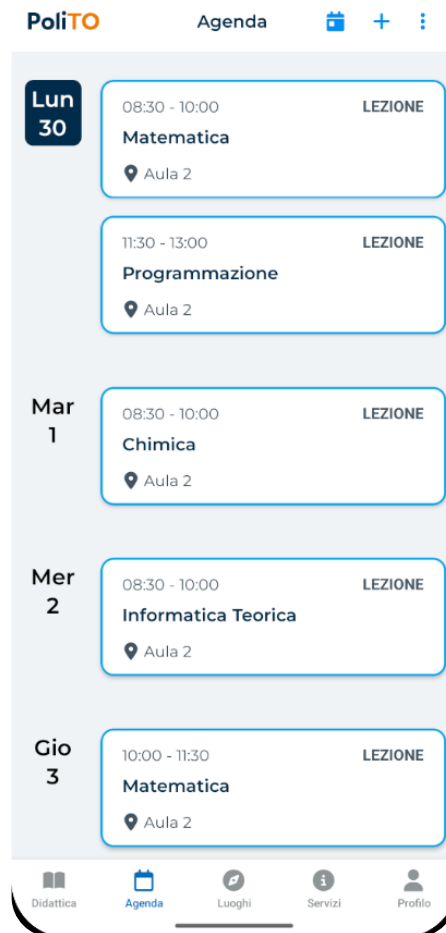


Figura 6.22: Schermata giornaliera: AgendaScreen.

AgendaWeekScreen Mostra l'intera settimana accademica, offrendo una panoramica più ampia e utile per pianificazioni a medio termine.

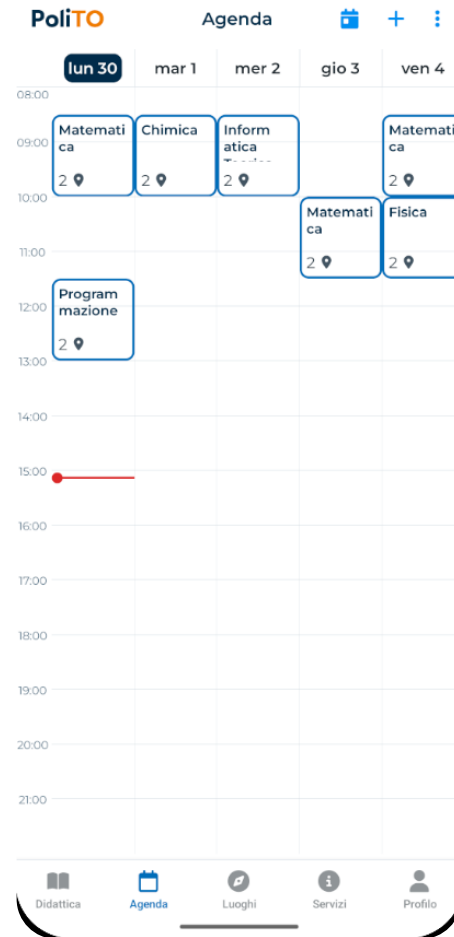


Figura 6.23: Schermata settimanale: AgendaWeekScreen.

NoteForm È lo screen collegato al pulsante “+” e permette la creazione di un nuovo appuntamento. Al suo interno è presente un form con tutti i campi necessari alla definizione dell’impegno (data, orario, titolo, descrizione, luogo, ecc.).

The screenshot displays the 'Crea evento' (Create event) screen. At the top, there is a back arrow and the title 'Crea evento'. The form consists of several sections: 'Tipo di evento' (Event type) with a dropdown menu showing 'Appuntamento'; 'Titolo' (Title) with a text input field; 'Descrizione' (Description) with a text input field; 'Imposta data' (Set date) with a date picker showing 'Data: 2025-07-01'; and 'Seleziona slot orario' (Select time slot) with two dropdown menus for 'Orario inizio' and 'Orario fine'. Below these fields is a large blue button labeled 'Carica evento'. At the bottom, there is a navigation bar with five icons: 'Didattica' (teaching), 'Agenda' (calendar), 'Luoghi' (locations), 'Servizi' (services), and 'Profilo' (profile).

Figura 6.24: Form per nuovo appuntamento: NoteForm.

SingleElementScreen È lo screen utilizzato per mostrare un singolo evento in dettaglio. Contiene informazioni su luogo, orario, eventuale corso associato e descrizione. Offre anche la possibilità di eliminarlo.

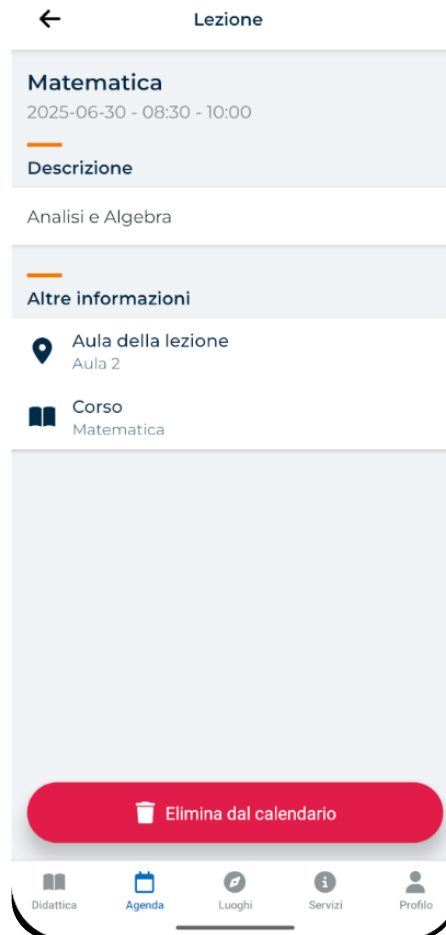


Figura 6.25: Dettaglio di un evento: SingleElementScreen.

6.3.3 Componenti adottati

La sezione **Agenda** riprende molti componenti e pattern già utilizzati nelle sezioni precedenti, adattandoli al contesto del calendario docente. La visualizzazione degli eventi si basa su un sistema ibrido, composto da elementi statici (form e dettagli) e una struttura dinamica per la resa giornaliera e settimanale degli impegni.

I principali componenti riutilizzati includono:

- **TextInput**, **Card**, **CtaButton** — impiegati all'interno del form **NoteForm** per la creazione di un nuovo appuntamento.
- **SectionHeader** e **ListItem** — usati nella schermata di dettaglio di una lezione per strutturare logicamente i contenuti e fornire interazioni rapide (es. navigazione al luogo sulla mappa o alla pagina del corso).
- **Popover** — utilizzato per azioni contestuali, come ad esempio il passaggio tra layout giornaliero e settimanale o l'aggiornamento di un evento.

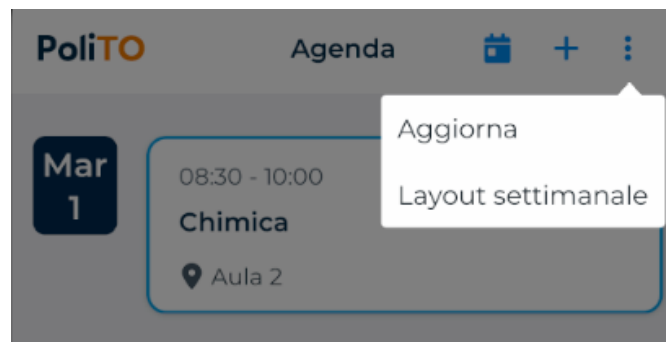


Figura 6.26: Popover per la gestione del layout.

Per la rappresentazione visiva dell'agenda sono stati invece introdotti alcuni componenti personalizzati:

- **AgendaCard** — una card modulare utilizzata per mostrare visivamente eventi, lezioni e appuntamenti all'interno della vista calendario. Gestisce icone, orari, titolo e luogo, ed è ottimizzata sia per il layout compatto (in visualizzazione settimanale) sia per quello esteso.

- **LectureCard** — estensione di **AgendaCard**, specifica per le lezioni. Aggiunge supporto per le aule virtuali e una breve descrizione dell'evento.

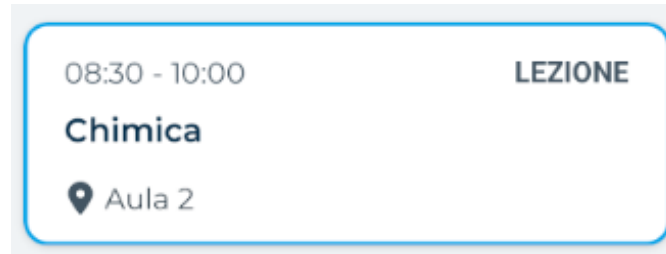


Figura 6.27: Esempio di LectureCard con descrizione e aula virtuale.



Figura 6.28: Esempio di LectureCard con descrizione e aula virtuale.

- **Calendar** — è il componente principale che costruisce e gestisce l'intera griglia del calendario. Gestisce:
 - il calcolo delle date da mostrare (giorno, settimana, ecc.);
 - le interazioni dell'utente (tap, swipe, evidenziazione);
 - il rendering di intestazione e corpo del calendario.

Internamente, il componente si divide in:

- **CalendarHeader** — mostra i giorni (o date) nella parte superiore, evidenziando il giorno attivo o corrente;
- **CalendarBody** — costruisce la vera e propria griglia temporale, suddivisa in ore e colonne giornaliere, e posiziona graficamente gli eventi (lezioni o appuntamenti).

6.4 Servizi

La sezione **Servizi** rappresenta un punto di accesso centralizzato a una serie di funzionalità eterogenee messe a disposizione dal Politecnico per i docenti. Essa non presenta una struttura complessa come la sezione **Didattica**, ma è progettata per offrire rapidità e immediatezza nella fruizione dei servizi più ricorrenti, integrando sia contenuti statici (es. documentazione, numeri di emergenza) sia dinamici (es. rubriche, prenotazioni e firme digitali).

L'obiettivo principale è facilitare le attività operative del personale docente, fornendo strumenti digitali efficienti e integrati con i sistemi esistenti.

6.4.1 Funzionalità implementate

Di seguito vengono illustrate le principali funzionalità accessibili tramite la sezione **Servizi**:

Mail Consente al docente di accedere rapidamente alla propria casella email istituzionale. Il servizio reindirizza l'utente al portale web dedicato alla gestione della posta, aprendo automaticamente il browser.

News & Eventi Collega il docente alla pagina ufficiale contenente notizie e aggiornamenti sugli eventi dell'ateneo, mantenendolo informato sulle attività accademiche e istituzionali in corso.

Persone Funziona da vera e propria *rubrica di ateneo*, permettendo la ricerca di qualsiasi persona afferente al Politecnico, come docenti, ricercatori e personale tecnico. Una volta selezionata una persona, viene visualizzata una scheda contenente le informazioni di contatto e il profilo istituzionale.

Firma Digitale Servizio innovativo che consente la firma di documenti direttamente da smartphone, eliminando la necessità di accedere da PC. La firma avviene mediante autenticazione con password e OTP (*One-Time Password*), abilitando una gestione veloce e sicura di circolari e comunicazioni ufficiali.

Segnala Guasto Permette di inviare una segnalazione relativa a un guasto tecnico all'interno delle strutture del Politecnico, specificando la zona o l'aula interessata. Questo servizio mira a ottimizzare la gestione degli interventi di manutenzione.

Prenotazione Spazi Fornisce la possibilità di prenotare spazi dell'ateneo per diverse finalità: aule per lezione, sale riunioni, spazi per conferenze o incontri ristretti. L'interfaccia guida l'utente nella compilazione della richiesta, adattandosi al tipo di ambiente e all'esigenza specifica.

Emergenze Sezione informativa dedicata alla gestione delle situazioni di emergenza. Contiene istruzioni da seguire in caso di eventi come incendi o terremoti, distinte per tipologia. Include inoltre shortcut rapidi per contattare i servizi di emergenza.

Supporto audio-video in aula Raccoglie indicazioni per richiedere assistenza tecnica durante le lezioni, in caso di problemi con le attrezzature audio-video. Include il numero telefonico da contattare per ricevere supporto immediato da parte dello staff tecnico.

6.4.2 Architettura del componente

La sezione **Servizi** è gestita tramite il navigatore principale **ServiceNavigator**, che permette di navigare tra una varietà di schermate, ciascuna corrispondente a un servizio specifico.

Alcuni servizi, come **Mail** e **News&Eventi**, non sono attualmente implementati come schermate native, in quanto già presenti e consolidati all'interno dell'applicazione per studenti. Verranno eventualmente integrati in una fase successiva tramite redirect o integrazione diretta.

Tutti i servizi descritti di seguito, invece, rappresentano funzionalità **sviluppate ex-novo** specificamente per l'applicazione docente, in risposta a necessità concrete non coperte dalla precedente architettura.

ServiceScreen Contiene le **Card** con tutti i servizi disponibili. Ogni servizio può essere contrassegnato come preferito tramite un'icona a forma di stella, che consente di evidenziarlo nella parte superiore della schermata.

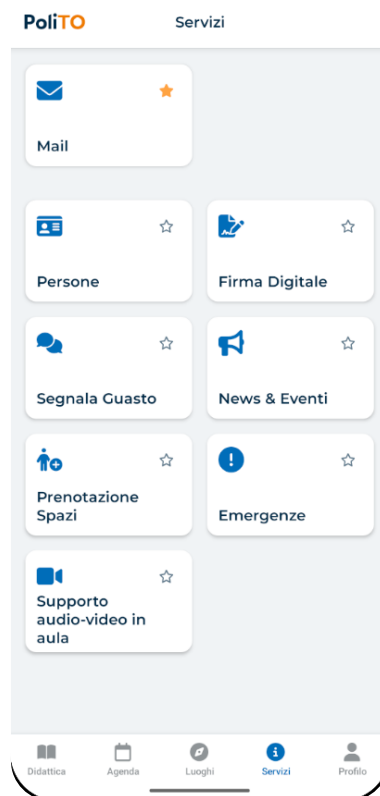


Figura 6.29: Schermata servizi principali

PersoneScreen Dedicata al servizio **Persone**, funge da rubrica per la ricerca di qualsiasi figura istituzionale (es. docenti, ricercatori). Comprende una barra di ricerca e una sezione di preferiti. Selezionando un contatto, si viene reindirizzati alla **ContactScreen**, dove è possibile visualizzarne i dettagli e aggiungerlo o rimuoverlo dai preferiti.

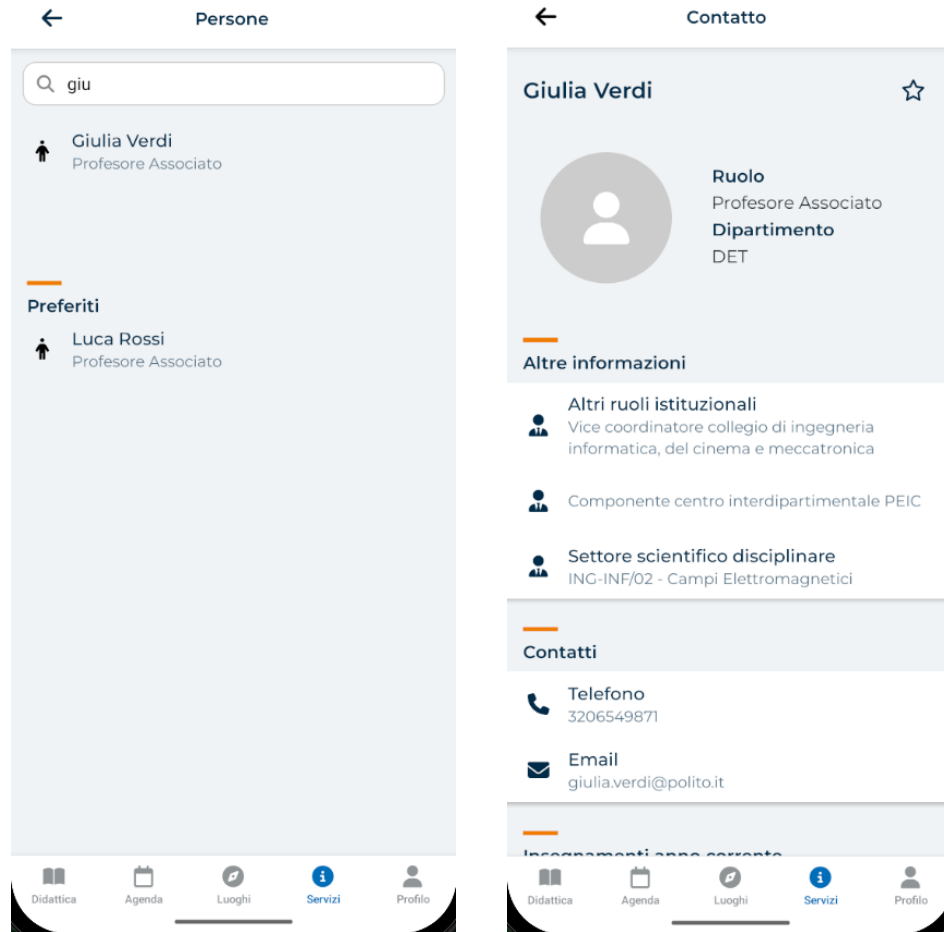


Figura 6.30: Schermi principali del servizio Persone.

DigitalSignatureScreen Relativa al servizio **Firma Digitale**, mostra la lista dei documenti in attesa di firma o già firmati. Ogni documento può richiedere più firme: un **Badge** indica se la propria firma è stata apposta e se sono necessarie ulteriori firme (“Attesa Firma”) oppure se il processo è completo (“Firme completate”). Tramite la selezione del documento si accede alla **SignatureScreen**, dove è possibile firmare utilizzando una password fissa e un OTP, dopo aver cliccato su “Firma”.

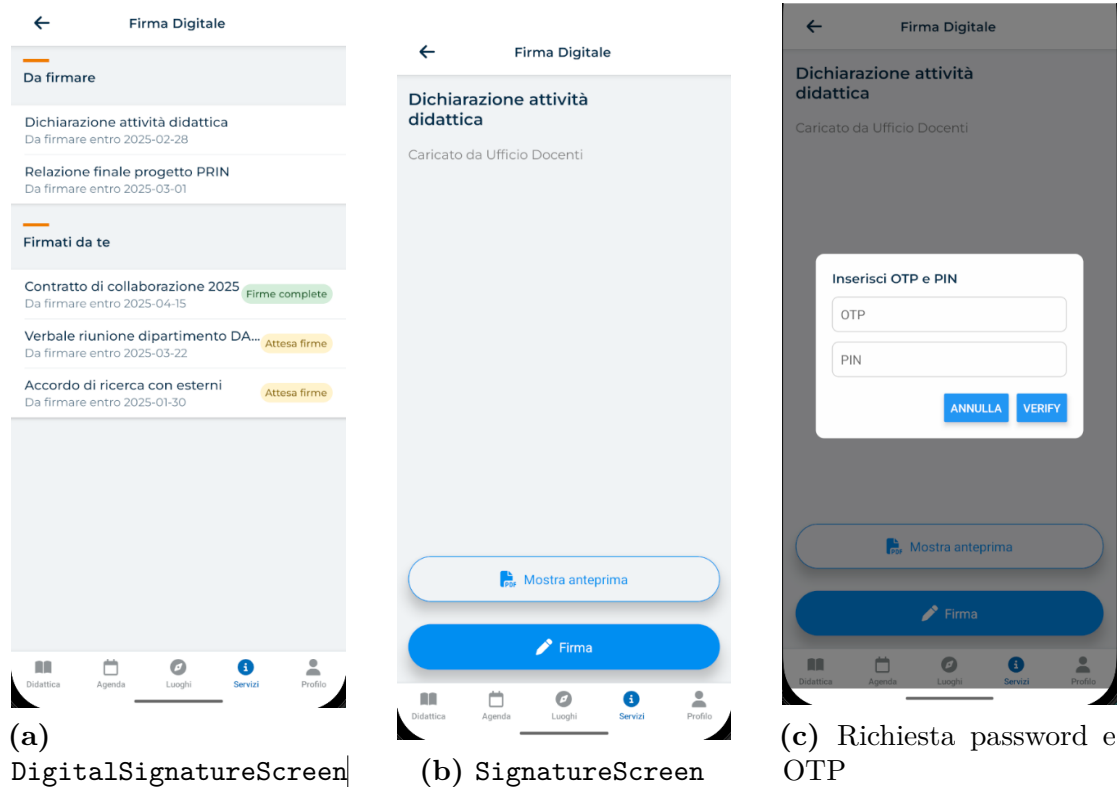


Figura 6.31: Schermi principali del servizio Firma Digitale.

IssueReport Per il servizio **Segnala Guasto**, mostra l'elenco delle segnalazioni effettuate. Ogni voce è accompagnata da un **Badge** che ne indica lo stato (“In attesa”, “Risolta”, “Respinta”). Il pulsante **CtaButton** permette di accedere al form (**IssueReportForm**) per creare una nuova segnalazione. Selezionando una voce si apre la **IssueDetails**, dove si visualizzano data, luogo e dettagli della segnalazione, con la possibilità di annullarla se ancora in attesa.

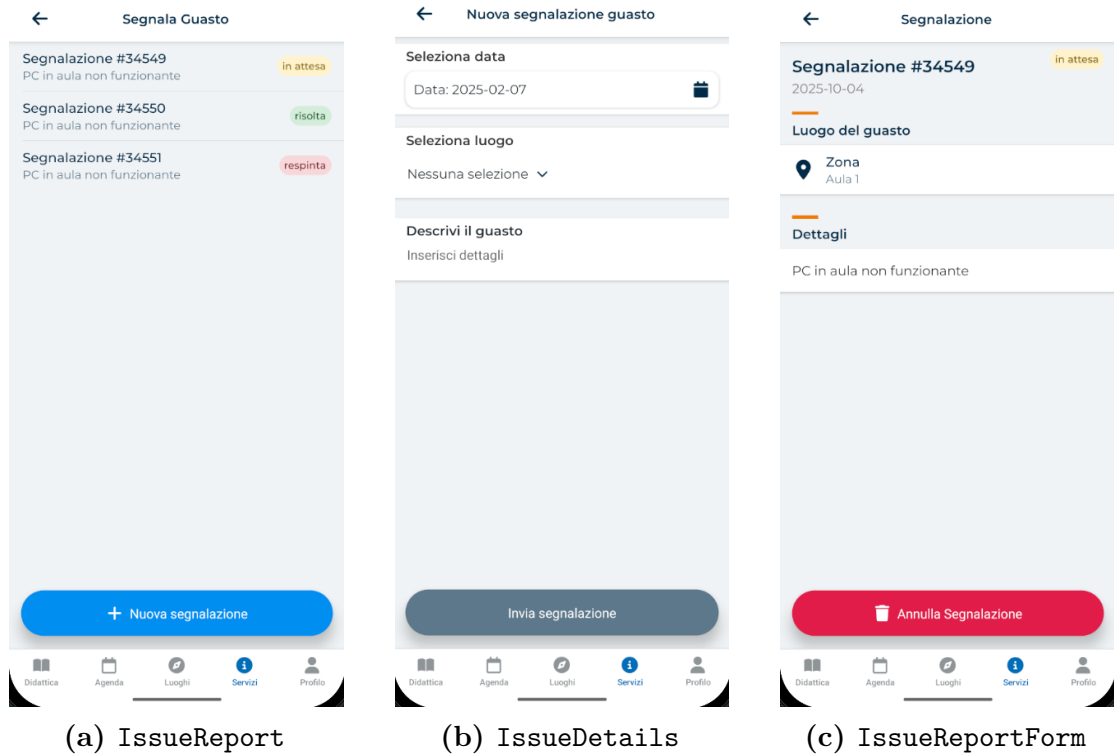


Figura 6.32: Schermi principali del servizio Segnala Guasto.

BookingScreen Per il servizio **Prenotazione Spazi**, permette la scelta del tipo di prenotazione: aule, spazi eventi o strutture dipartimentali. A seconda della selezione, si accede a schermate dedicate ad esempio **BookStructureRoomScreen** per le sale di dipartimento (come sale riunioni), dove si possono consultare le richieste inviate e crearne di nuove tramite form (**BookStructureForm**), indicando data, slot orario, locale da prenotare tra quelli disponibili e la motivazione. Anche qui le richieste sono tracciate tramite badge (“In attesa”, “Accettata”, “Respinta”), e se in attesa possono essere annullate.

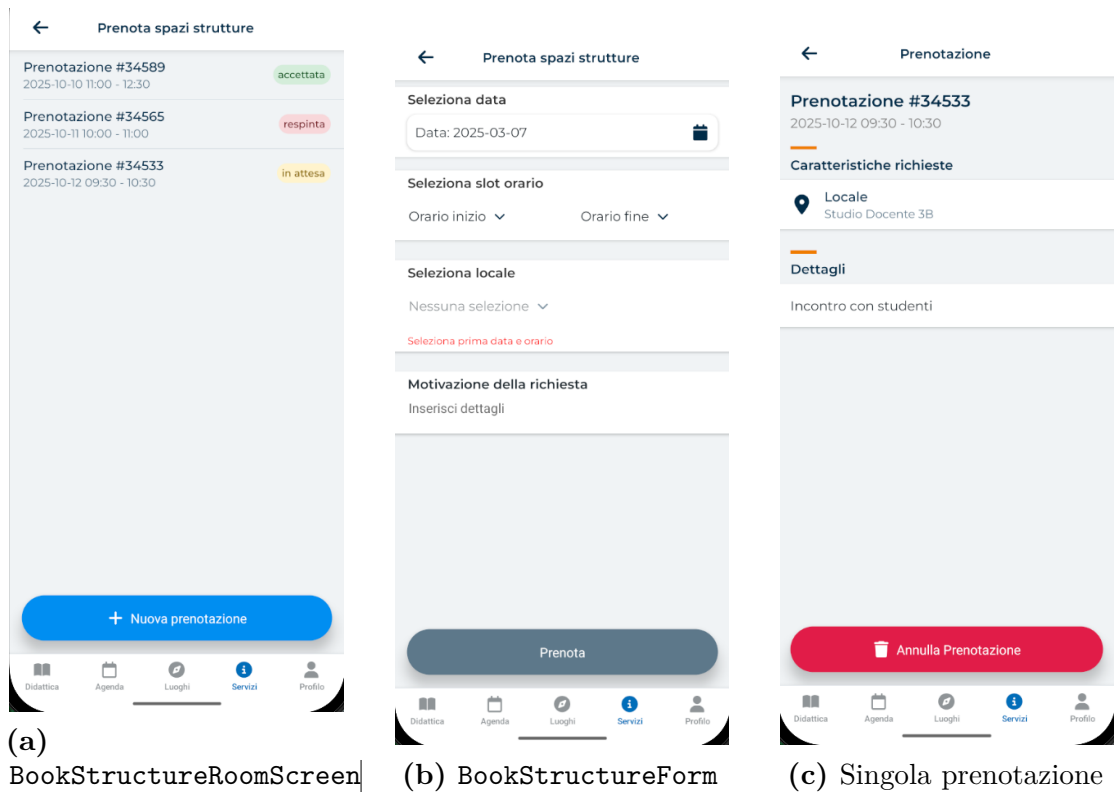


Figura 6.33: Schermi principali del servizio Prenotazione Spazi.

EmergencyScreen Per il servizio **Emergenze**, raccoglie le istruzioni per la gestione di diverse situazioni critiche (incendio, evacuazione, terremoto, rapina, sostanze pericolose, infortunio). Sono presenti anche shortcut per contattare i numeri di emergenza. Cliccando su ciascuna voce, si accede alla schermata dettagliata della specifica emergenza.

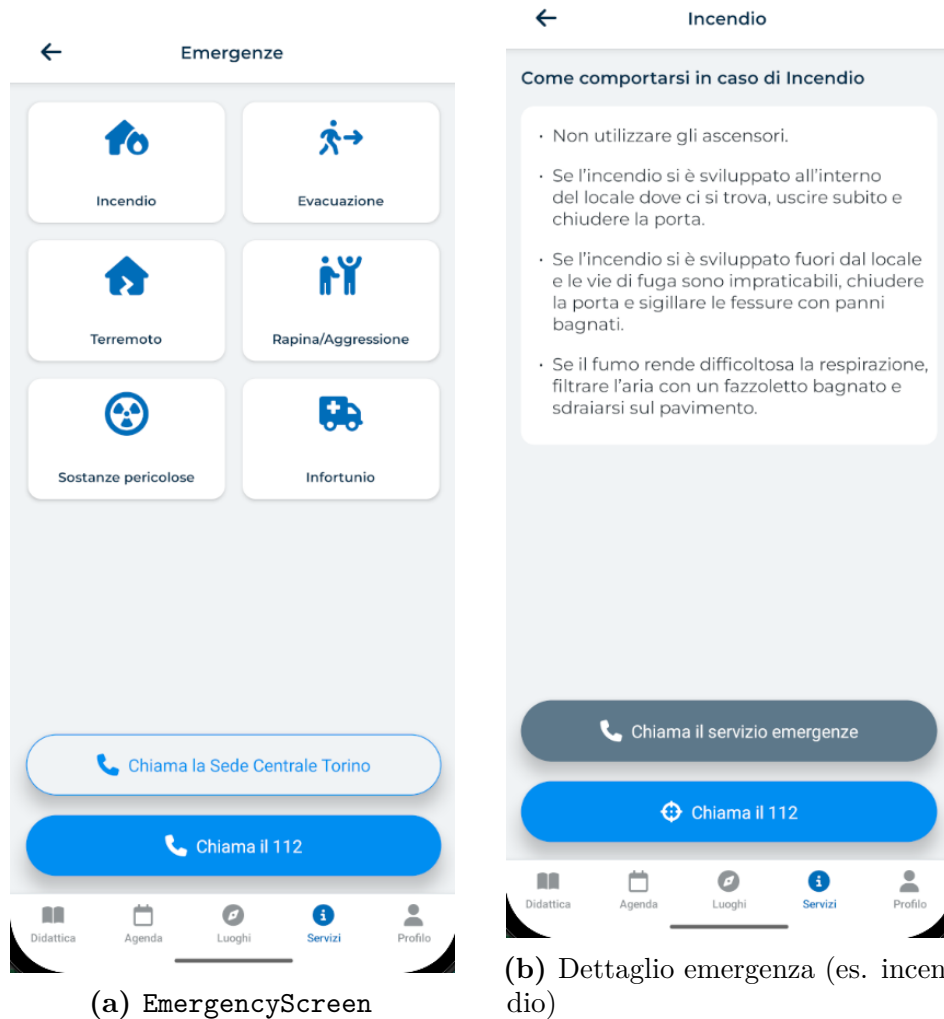


Figura 6.34: Schermi principali del servizio Emergenze.

SupportScreen Per il servizio **Supporto audio-video in aula**, fornisce gli orari di disponibilità e i numeri per contattare tempestivamente i tecnici.

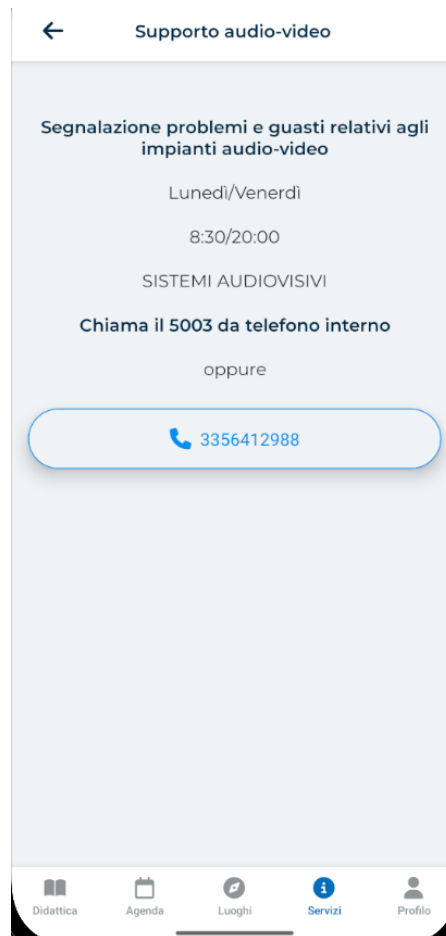


Figura 6.35: Schermo della sezione Supporto audio-video

6.4.3 Componenti adottati

Per la sezione **Servizi** sono stati impiegati numerosi componenti già descritti in precedenza, tra cui:

- **TextInput**, **Card**, **CtaButton** — utilizzati nei vari form per inserimento e invio di dati;
- **SectionHeader** e **ListItem** — per rappresentare intestazioni e liste nei servizi che prevedono elementi ripetuti (prenotazioni, segnalazioni, firme...).

Tuttavia, i componenti che caratterizzano in modo peculiare questa sezione sono due:

- **ServiceCard** — componente che rappresenta graficamente ciascun servizio. Ogni card mostra un'icona associata, un titolo e, opzionalmente, un badge numerico (es. documenti da firmare). È possibile contrassegnare i servizi come “preferiti” cliccando sull'icona a forma di stella, che ne provoca lo spostamento nella parte superiore dello schermo **ServiceScreen**. Il componente sfrutta elementi come **Icon**, **Text**, **IconButton** e **TouchableCard** per definire un layout ricco e interattivo.

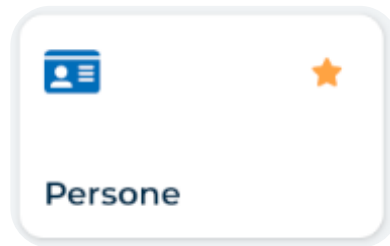


Figura 6.36: Una **ServiceCard** contrassegnata come preferita.

- **Badge** — utilizzato per indicare lo stato di una segnalazione o di una prenotazione, oppure per mostrare un contatore (es. firme in attesa). Ha uno stile facilmente distinguibile grazie ai colori di sfondo e testo personalizzabili. Può includere un'icona opzionale, affiancata a un testo descrittivo.



Figura 6.37: Esempi di utilizzo del componente **Badge**.

Questi elementi contribuiscono a rendere la sezione Servizi fortemente riconoscibile visivamente, migliorando l'usabilità grazie a pattern ricorrenti e alla visibilità dello stato di ciascuna interazione.

6.4.4 Luoghi

La sezione **Luoghi** rappresenta una componente già consolidata e perfettamente funzionante, ereditata direttamente dall'app studenti del Politecnico, senza modifiche. È stata integrata all'interno dell'applicazione docente per offrire un'esperienza completa e coerente, ma non ha richiesto interventi progettuali o di sviluppo, in quanto già curata da un altro team.

Tale sezione consente di **consultare le mappe dell'ateneo** per ottenere informazioni dettagliate sui luoghi, come aule, laboratori, uffici o spazi comuni. Ogni punto d'interesse è visualizzabile su mappa interattiva e, una volta selezionato, fornisce una scheda descrittiva contenente il **nome del luogo**, il **piano e l'edificio di riferimento**, la **capienza dell'ambiente** e le **dotazioni presenti in aula**, rendendo più semplice la pianificazione di attività didattiche o logistiche.

Oltre alla sua funzione autonoma di ricerca e consultazione dei luoghi, è **prevista un'integrazione con la sezione Agenda**: quando un evento (come una lezione o un appuntamento) contiene un luogo associato, è possibile cliccare direttamente sul nome del luogo per essere reindirizzati alla mappa nella sezione **Luoghi**, posizionandosi esattamente nel punto corrispondente. Questa funzionalità migliora l'esperienza utente, facilitando l'orientamento all'interno dell'ateneo e ottimizzando l'accesso alle informazioni logistiche.

In sintesi, sebbene non sia stata oggetto di sviluppo diretto, la sezione Luoghi riveste un ruolo fondamentale nella navigazione e nell'integrazione dei servizi, e per questo motivo è stata mantenuta e inclusa nel progetto docente.

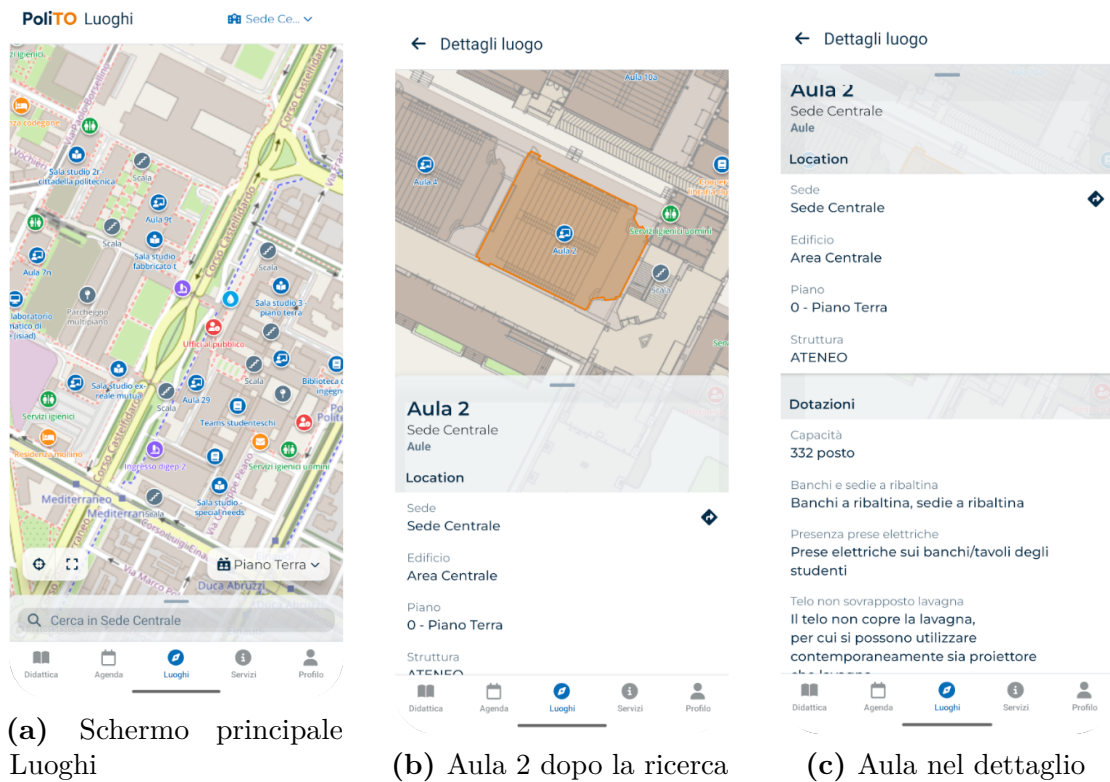


Figura 6.38: Schermi principali del servizio Luoghi.

Capitolo 7

Usability Testing

La valutazione dell'usabilità rappresenta una fase fondamentale nel processo di sviluppo di un'interfaccia, in quanto consente di verificare l'efficacia, l'efficienza e la soddisfazione con cui gli utenti riescono a raggiungere i propri obiettivi. Nel contesto dell'applicazione *Polito Faculty*, è stato condotto un test di usabilità con l'obiettivo di validare l'esperienza d'uso del sistema, individuare eventuali criticità e raccogliere suggerimenti per miglioramenti futuri.

La metodologia adottata per la pianificazione e conduzione dei test di usabilità è ispirata alle indicazioni fornite da Rubin e Chisnell [5].

Il test ha coinvolto un gruppo ristretto ma rappresentativo di utenti reali (docenti e ricercatori del Politecnico di Torino), ed è stato progettato per simulare situazioni realistiche e attività frequenti. I dati raccolti durante le sessioni hanno fornito informazioni sia quantitative che qualitative, utili per rifinire l'interfaccia e confermare le scelte progettuali effettuate.

7.1 Obiettivi e metodologia del test

L'obiettivo principale del test di usabilità è stato quello di validare le funzionalità implementate nell'applicazione *Polito Faculty* e accertarsi che esse fossero intuitive, accessibili e coerenti con le aspettative degli utenti finali. Come in ogni test di usabilità, lo scopo non era valutare i partecipanti, bensì l'interfaccia: ogni difficoltà o errore è stato considerato un indicatore di miglioramento per il sistema.

Le sessioni si sono svolte in presenza, in modalità one-to-one, e sono durate in media circa 20 minuti. Ogni sessione ha coinvolto due o tre membri del team: una persona (l'autore) con il ruolo di facilitatore, che dialogava con il partecipante e guidava l'esecuzione dei compiti, mentre le altre si occupavano di prendere appunti su azioni, difficoltà e tempi.

Durante lo svolgimento, è stato richiesto ai partecipanti di verbalizzare i propri pensieri (“think aloud”) per facilitare la comprensione delle motivazioni e delle aspettative che guidavano le loro interazioni.

7.2 Descrizione dei partecipanti

Al test hanno partecipato cinque persone, selezionate tra coloro che avevano risposto al questionario iniziale e avevano dato disponibilità a essere contattati come beta tester. La selezione è stata effettuata in modo casuale, con l’unico vincolo di garantire diversità in termini di età e dipartimento di appartenenza.

Tutti i partecipanti avevano già avuto almeno un’esperienza d’uso con la versione attuale dell’applicazione docenti, sebbene la maggior parte di loro dichiarasse di preferire il portale web per le attività quotidiane.

Partecipante	Età	Dipartimento e Ruolo
Partecipante 1	51	Dipartimento DAD, professore/professoressa
Partecipante 2	64	Dipartimento DISAT, professore/professoressa
Partecipante 3	34	Dipartimento DAD, ricercatore
Partecipante 4	31	Dipartimento DET, professore/professoressa
Partecipante 5	51	Dipartimento DENERG, professore/professoressa

Tabella 7.1: Profilo dei partecipanti alle interviste

7.3 Scenari di utilizzo testati

Durante le sessioni di test sono stati proposti ai partecipanti diversi scenari d’uso, ciascuno articolato in uno o più task concreti. Per esempio, in un caso è stato chiesto a un docente di modificare il proprio numero di telefono nel profilo (Task T0) oppure, di aggiungere uno studente a un appello e contattare altri partecipanti (Task T7), simulando una situazione in cui serve aggiornare dati personali e gestire comunicazioni urgenti prima di un esame. Le attività simulate riproducevano situazioni quotidiane nell’esperienza dei docenti, con l’obiettivo di verificare sia la copertura funzionale dell’applicazione sia la coerenza e l’intuitività dei flussi interattivi.

Le task sono state costruite per coprire tutte le principali funzionalità dell'applicazione. Non è stato necessario testare ogni singola funzione, poiché molte di esse si basano su pattern ricorrenti già inclusi nei task (es. aggiunta, modifica o rimozione di elementi). Di conseguenza, alcune funzionalità ridondanti sono state omesse senza compromettere la completezza del test.

Le task sono stati costruiti per coprire tutte le principali funzionalità dell'applicazione. Non è stato necessario testare ogni singola funzione, poiché molte di esse si basano su pattern ricorrenti già inclusi nei task (es. aggiunta, modifica o rimozione di elementi). Di conseguenza, alcune funzionalità ridondanti sono state omesse senza compromettere la completezza del test.

Per ciascun task sono stati osservati e annotati tre aspetti fondamentali:

1. **Errori** — eventuali errori commessi o momenti di confusione durante l'interazione.
2. **Feedback qualitativo** — commenti spontanei, esitazioni, osservazioni verbali.
3. **Successo del task** — valutazione sintetica del completamento del task, su una scala a 4 livelli.

Nella tabella alla pagina successiva sono illustrate le varie task e i criteri di successo per ognuna. La varietà dei task ha permesso di esplorare tutte le sezioni principali dell'app, fornendo così una panoramica rappresentativa dell'esperienza complessiva dell'utente.

Tabella 7.2: Task proposte durante il test di usabilità

ID	Obiettivo	Criterio di successo atteso
T0	Modificare il numero di telefono nel profilo	Accedere al profilo, cliccare sull'icona di modifica, aggiornare il numero e salvarlo
T1	Sostituire un collaboratore a un corso	Rimuovere il collaboratore esistente, aggiungerne uno nuovo e assegnargli i permessi adeguati
T2	Pubblicare un nuovo avviso e nascondere uno esistente	Creare un nuovo avviso con scadenza e nascondere un altro avviso già presente
T3	Correggere materiale didattico errato	Eliminare il file errato e caricare quello corretto nella sezione Materiale
T4	Aggiungere una lezione manuale nel registro	Navigare nella sezione Lezioni, cliccare su “+” e compilare il form per la nuova lezione
T5	Aggiungere uno studente a un corso e consultare i suoi contatti	Aggiungere uno studente e accedere alla sua scheda contatto per visualizzarne le info
T6	Contattare un gruppo di studenti	Selezionare più studenti e usare l'opzione per inviare un messaggio collettivo
T7	Aggiungere uno studente a un appello e contattare altri studenti	Iscrivere lo studente all'appello e contattare contemporaneamente più partecipanti
T8	Verificare il luogo di una lezione	Accedere all'Agenda, selezionare la lezione e cliccare sul luogo per aprire la mappa
T9	Aggiungere una lezione dal calendario	Aprire il calendario, cliccare su “+”, inserire i dati della lezione e salvarla
T10	Trovare il contatto email di un docente	Usare la sezione “Persone” per cercare il docente e leggere la sua email
T11	Firmare un contratto digitalmente	Accedere alla sezione “Firma Digitale”, selezionare il contratto, inserire password e OTP
T12	Segnalare un guasto in aula	Usare l'apposita sezione per inviare una nuova segnalazione compilando il modulo
T13	Prenotare una sala per un incontro	Accedere alla sezione “Prenotazione spazi” e completare la richiesta di prenotazione

7.4 Risultati del test

L'analisi dei risultati raccolti durante le sessioni di usability testing ha permesso di evidenziare con chiarezza il livello di usabilità generale dell'applicazione, individuando punti di forza e criticità. Per ciascun task, i valutatori hanno assegnato un punteggio su scala da 1 a 4, dove:

- 1 = il compito non è stato completato;
- 2 = completato con gravi difficoltà o incomprensioni;
- 3 = completato con problemi minori (incertezze, rallentamenti);
- 4 = completato correttamente e senza difficoltà.

Di seguito è riportata una sintesi delle osservazioni emerse per ciascun task, con il relativo punteggio medio:

T0 — Modifica numero di telefono nel profilo personale La maggior parte dei partecipanti ha inizialmente cercato di modificare il numero cliccando direttamente sul testo visualizzato. Solo successivamente hanno individuato il pulsante di modifica generale. Questo comportamento suggerisce un'aspettativa non soddisfatta nell'interfaccia. Il task è stato completato in tutti i casi, ma con incertezza iniziale.

Punteggio medio: 3,2

T1 — Gestione collaboratori Tutti i partecipanti hanno riscontrato un bug nella selezione del nuovo collaboratore: il primo clic chiudeva la tastiera ma non selezionava l'elemento. Inoltre, l'interfaccia per l'assegnazione dei permessi non era sempre chiara.

Punteggio medio: 3,0

T2 — Gestione avvisi Task completato con facilità. Tuttavia, alcuni partecipanti hanno suggerito di rendere più visibile lo stato di visibilità degli avvisi (es. icona "visibile"/"nascosto").

Punteggio medio: 3,8

T3 — Gestione materiale didattico Completato correttamente da tutti, ma un utente non ha inserito un campo obbligatorio e non ha compreso subito l'errore, segnalando la mancanza di feedback espliciti.

Punteggio medio: 3,6

T4 — Inserimento lezione nel registro Più della metà dei partecipanti ha avuto difficoltà a causa del nome della sezione “Lezioni” al posto di “Registro”. Il task è stato completato, ma con disorientamento iniziale.

Punteggio medio: 2,8

T5 — Aggiunta studente e consultazione contatto L'icona a tre puntini per l'aggiunta non è stata interpretata immediatamente. Inoltre, alcuni hanno segnalato l'assenza di ordinamento alfabetico nella lista.

Punteggio medio: 3,2

T6 — Contatto con gruppo di studenti Una sola persona ha avuto difficoltà nella selezione multipla a causa della hitbox delle checkbox. Gli altri hanno completato facilmente.

Punteggio medio: 3,8

T7 — Gestione iscrizione appelli Completato agevolmente da tutti i partecipanti.

Punteggio medio: 4,0

T8 — Verifica del luogo di una lezione Nessuna difficoltà significativa. Solo un utente non ha immediatamente capito che il luogo era cliccabile.

Punteggio medio: 3,8

T9 — Creazione lezione tramite calendario Confusione iniziale per alcuni partecipanti che cercavano la funzione all'interno del corso invece che nel calendario. Risolto rapidamente.

Punteggio medio: 3,2

T10 — Ricerca contatto docente Task completato senza difficoltà da tutti.

Punteggio medio: 4,0

T11 — Firma digitale di un contratto Completato in modo corretto da tutti. Un solo partecipante ha inizialmente cercato nel posto sbagliato.

Punteggio medio: 3,8

T12 — Segnalazione guasto Nessuna criticità osservata.

Punteggio medio: 4,0

T13 — Prenotazione sala per incontro Completato in tutti i casi, ma la terminologia (spazi struttura / eventi / aule) è stata giudicata poco chiara. Sugeriti esempi o descrizioni più esplicite.

Punteggio medio: 3,2

Nel complesso, tutti i task sono stati completati con successo. Tuttavia, alcuni di essi hanno mostrato margini di miglioramento, in particolare rispetto alla terminologia, ai feedback in caso di errore e all'intuitività di alcune icone o pulsanti. Le principali proposte migliorative saranno discusse nella sezione successiva.

Proposte di miglioramento

L'analisi dei task ha portato all'identificazione di alcuni problemi di usabilità minori ma ricorrenti, per i quali si propongono le seguenti soluzioni:

Modifica del profilo utente Per rendere l'interfaccia più aderente alle aspettative degli utenti, si potrebbe considerare la possibilità di attivare la modifica tenendo premuto sul campo desiderato. Tuttavia, una volta compreso il funzionamento del pulsante di modifica generale, l'interazione diventa rapidamente familiare, quindi non è considerato un intervento prioritario.

Bug nella selezione di studenti e collaboratori Quando si effettua una ricerca nella `BottomModal`, il primo tocco su un risultato chiude la tastiera ma non seleziona l'elemento. È necessario un secondo tocco per completare l'azione. È raccomandato risolvere questo comportamento, permettendo la selezione con un singolo tocco.

Stato visibilità avvisi È stato suggerito di aggiungere un'icona direttamente nel `ListItem` degli avvisi che indichi chiaramente se un avviso è visibile o nascosto, evitando ambiguità visive.

Feedback nei form Attualmente i pulsanti di invio nei form rimangono disattivati fino al completamento di tutti i campi obbligatori. Si consiglia di integrare un messaggio di alert al click sul pulsante disattivato, in modo da indicare esplicitamente all'utente quali campi risultano mancanti.

Terminologia della sezione registro Diversi partecipanti hanno confuso la sezione "*Lezioni*" con quella per l'aggiunta di lezioni manuali nel registro. Si suggerisce di rinominare la sezione in "*Registro*" per riflettere meglio la sua funzione e migliorare la riconoscibilità.

Precisione delle hitbox In alcune interazioni (es. selezione studenti o frecce di navigazione), le hitbox risultano troppo piccole o mal posizionate, causando click accidentali. È consigliato aumentare l'area interattiva delle checkbox e dei pulsanti di navigazione per migliorare la precisione del tocco.

Chiarezza nella prenotazione spazi La distinzione tra “prenota aula”, “spazi per eventi” e “spazi struttura” è risultata poco chiara. Sarebbe utile inserire un sottotitolo descrittivo sotto ogni voce, specificando a quali esigenze o capienze si rivolge ciascuna sezione, per aiutare l'utente a scegliere correttamente.

Nel complesso, le problematiche emerse sono facilmente risolvibili e non compromettono l'usabilità generale dell'app. Le soluzioni proposte mirano a migliorare ulteriormente l'efficienza e la chiarezza dell'interazione, allineando l'interfaccia alle aspettative degli utenti e rendendo l'esperienza complessiva più fluida e intuitiva.

Capitolo 8

Conclusioni

La realizzazione del frontend dell'applicazione *PoliTO Faculty* ha rappresentato un'opportunità concreta per affrontare una sfida reale all'interno dell'ecosistema digitale del Politecnico di Torino, contribuendo a colmare un divario funzionale e progettuale rispetto all'applicazione dedicata agli studenti.

Il percorso sviluppato in questa tesi ha combinato aspetti di analisi, progettazione e implementazione, seguendo un approccio user-centered. L'indagine preliminare ha consentito di raccogliere esigenze autentiche da parte del corpo docente, mentre la fase di prototipazione e sviluppo ha permesso di tradurre tali necessità in interfacce coerenti, accessibili e in linea con le linee guida istituzionali.

Il testing condotto ha confermato l'efficacia delle scelte compiute, evidenziando punti di forza ma anche spunti per riflessioni e miglioramenti. L'applicazione rappresenta un primo passo verso una digitalizzazione più mirata e partecipata dei servizi per il personale accademico. Progetti come questo mostrano l'importanza di adottare metodologie iterative e partecipative nella progettazione di strumenti digitali, sottolineando come il valore di un'interfaccia non risieda solo nella sua estetica, ma nella capacità di rispondere concretamente ai bisogni degli utenti.

8.1 Sviluppi futuri

Gli sviluppi più immediati dovrebbero concentrarsi sull'implementazione dei miglioramenti emersi durante le sessioni di test. In particolare, è opportuno intervenire sui punti critici riscontrati in termini di chiarezza, fluidità di navigazione e visibilità di alcune funzionalità. Questi interventi costituirebbero un primo affinamento importante, capace di aumentare l'efficacia complessiva dell'interfaccia e la soddisfazione degli utenti.

In parallelo, la prossima fase cruciale sarà l'implementazione del backend, indispensabile per fornire all'applicazione dati reali, funzionalità persistenti e integrazione con i servizi informativi d'Ateneo. La progettazione dell'infrastruttura server dovrà garantire efficienza, scalabilità e sicurezza, costituendo la base per una distribuzione effettiva dell'applicazione.

Proseguire con cicli regolari di testing e raccolta di feedback sarà fondamentale per garantire che l'app evolva in modo coerente con le reali esigenze del corpo docente e con i cambiamenti nell'organizzazione dei servizi accademici.

Bibliografia

- [1] Arlene Fink. *How to Manage, Analyze and Interpret Survey Data*. 2^a ed. The Survey Kit. SAGE Publications, 2017 (cit. a p. 5).
- [2] David Engelberg e Ahmed Seffah. «A Framework for Rapid Mid-Fidelity Prototyping of Web Sites». In: *Conference Paper* (2002) (cit. a p. 16).
- [3] Meta Platforms, Inc. *React Native Documentation*. 2025. URL: <https://reactnative.dev/docs/getting-started> (cit. a p. 26).
- [4] Politecnico di Torino. *PoliTO Students App*. <https://github.com/polito/students-app>. Repository github opensource. 2024 (cit. a p. 28).
- [5] Jeffrey Rubin e Dana Chisnell. *Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests*. 2nd. Wiley, 2008 (cit. a p. 69).