

ALMA MATER STUDIORUM · UNIVERSITÀ DI
BOLOGNA

SCUOLA DI SCIENZE
Corso di Laurea in Informatica

Progettazione e sviluppo di una
piattaforma mobile per servizi di
meccanica a domicilio: Il caso
Ti-Riparo

Relatore:
Prof.
Federico Montori

Presentata da:
Baha Rababa

III Sessione
Anno Accademico 2024/2025

*"A chi resiste nel caos del mondo,
e sceglie di non avere paura"*

Sommario

La manutenzione automobilistica resta uno dei settori meno digitalizzati dell'economia dei servizi, nonostante la crescente domanda di soluzioni comode e trasparenti da parte degli automobilisti. Questa tesi documenta la progettazione, lo sviluppo e la validazione di Ti-Riparo, una piattaforma mobile per la prenotazione di interventi meccanici a domicilio nell'area metropolitana di Bologna. L'architettura si basa su React Native con Expo, Supabase come backend serverless e servizi cloud a costo zero, dimostrando che un singolo sviluppatore può realizzare un ecosistema completo, composto da app mobile cross-platform, dashboard amministrativa e sito web, con risorse minime. La piattaforma è stata validata con oltre 200 utenti registrati e un questionario di usabilità ispirato al System Usability Scale, ottenendo un punteggio composito di 4,35/5 e un Net Promoter Score di +64. L'analisi del modello di business evidenzia tuttavia criticità strutturali, tra cui un costo di acquisizione cliente elevato tramite pubblicità digitale, un tasso di conversione inferiore al 10% e un ROI annuale negativo. I risultati confermano la fattibilità tecnica e l'apprezzamento degli utenti per il servizio, ma mostrano che la qualità del prodotto, pur necessaria, non è sufficiente a garantire la sostenibilità economica senza un modello di acquisizione e conversione adeguato.

Introduzione

Portare l'auto in officina significa, nella maggior parte dei casi, perdere mezza giornata: telefonare per fissare un appuntamento, organizzare uno spostamento alternativo, attendere senza aggiornamenti, sperare che il preventivo non lieviti. Questa esperienza, familiare a milioni di automobilisti, appare sempre più anacronistica in un'epoca in cui ogni altro servizio è a portata di click. Dalla ristorazione alla mobilità, dall'assistenza domestica alla gestione bancaria, le piattaforme digitali hanno ridefinito il rapporto tra utente e fornitore, riducendo tempi di attesa, costi e complessità operative. Il settore della manutenzione automobilistica, tuttavia, resta largamente escluso da questa trasformazione. Da questa osservazione nasce Ti-Riparo, un'applicazione progettata per portare il servizio meccanico direttamente a domicilio. Il progetto, inizialmente concepito come sperimentazione tecnologica, si è evoluto in un servizio operativo con oltre 200 utenti registrati nell'area metropolitana di Bologna. La motivazione alla base del progetto presenta una duplice natura. Dal punto di vista tecnico, Ti-Riparo permette di esplorare l'integrazione tra framework di sviluppo ibrido e servizi cloud attraverso l'implementazione di moduli complessi. In termini di modello di business, il progetto propone un approccio innovativo per il settore della manutenzione veicolare, volto a ridurre le inefficienze dei processi esistenti. Un elemento centrale del lavoro riguarda l'applicazione pratica delle competenze acquisite durante il percorso di studi in Informatica per il Management. Ti-Riparo rappresenta la sintesi tra conoscenze teoriche e implementazione concreta, dimostrando come sia possibile trasformare un'idea in un servizio

funzionante e misurabile. La scelta di tecnologie open source e architetture serverless riflette inoltre la volontà di creare un modello sostenibile, accessibile e replicabile, elementi fondamentali per un progetto imprenditoriale in fase iniziale. Il Capitolo 1 presenta l'analisi del contesto e dello stato dell'arte, identificando il gap che questa tesi mira a colmare. I Capitoli 2 e 3 descrivono l'architettura e l'implementazione della piattaforma. Il Capitolo 4 analizza il modello di business e le metriche operative. Il Capitolo 5 presenta la valutazione dell'esperienza utente attraverso un questionario strutturato. Le Conclusioni sintetizzano i risultati e delineano le prospettive future.

Indice

Introduzione	iii
1 Analisi del contesto	1
1.1 Settore automotive e digitalizzazione dei servizi	1
1.2 Benchmark	3
1.2.1 Officlick – Il primo tentativo italiano	3
1.2.2 Fixter – L’approccio anglosassone	4
1.2.3 ClickMechanic – Il marketplace puro	5
1.2.4 Altri player minori	5
1.2.5 Lezioni apprese dall’analisi	6
1.2.6 Analisi comparativa e identificazione del gap	6
1.3 Valutazione della concorrenza locale	8
1.3.1 Le officine tradizionali	8
1.3.2 Le catene e i fast service	8
1.3.3 Tentativi di innovazione locale	9
1.3.4 Opportunità e posizionamento di Ti-Riparo	9
1.4 Contributo della tesi	10
1.5 Obiettivi del progetto	11
1.6 Metodologia di lavoro	11
1.6.1 Pianificazione e prototipazione	12
1.6.2 Sviluppo e versionamento	12
1.6.3 Testing e deployment	12
1.6.4 Approccio data-driven	13

2	Architettura del sistema	15
2.1	Requisiti funzionali	15
2.2	Scelta dello stack tecnologico	15
2.2.1	React Native con Expo	16
2.2.2	Supabase	17
2.2.3	Stripe	19
2.2.4	Tecnologie di supporto	19
2.2.5	Un ecosistema enterprise senza costi	20
2.3	Architettura generale	21
2.3.1	Overview architetturale	21
2.3.2	Flusso dati principale	22
2.3.3	Gestione dello stato	23
2.3.4	Sicurezza e autorizzazione	24
2.3.5	Scalabilità e performance	25
2.4	Modello dei dati	25
2.5	Descrizione dei moduli principali	26
2.5.1	Autenticazione e gestione utenti	26
2.5.2	Gestione prenotazioni e calendario	27
2.5.3	Sistema di preventivi e fatture	28
2.5.4	Notifiche push e mail	30
2.5.5	UI/UX design	30
3	Sviluppo e implementazione	35
3.1	Architettura del codice ed esempi	35
3.1.1	Struttura cartelle e organizzazione dei componenti	35
3.1.2	Componenti chiave	37
3.1.3	Gestione del flusso di prenotazione	38
3.2	L’ecosistema web: Dashboard e sito vetrina	39
3.2.1	Dashboard amministrativa con SvelteKit	39
3.2.2	Sito vetrina ti-riparo.it con Next.js	41
3.3	API e Edge Functions	41
3.3.1	Edge Function per la ricerca veicolo tramite targa	42

3.3.2	Integrazione con servizi esterni	43
3.4	Deploy, sicurezza e manutenzione	43
3.4.1	Pipeline di deploy con EAS Build	43
3.4.2	Over-the-air updates	44
3.4.3	Row Level Security e politiche di accesso	44
4	Management e strategia	47
4.1	Modello di business	47
4.2	Metriche di utilizzo	48
4.3	Analisi dei costi e ROI potenziale	49
4.3.1	Analisi di break-even	50
4.4	Rischi critici	51
4.5	Esito operativo e lezioni apprese	51
5	Valutazione dell'esperienza utente	53
5.1	Metodologia di valutazione	53
5.2	Profilo dei rispondenti	54
5.2.1	Frequenza di utilizzo	54
5.2.2	Canale di acquisizione	55
5.3	Usabilità generale	56
5.3.1	Userei Ti-Riparo frequentemente per la manutenzione della mia auto	57
5.3.2	Ho trovato l'app semplice da usare	58
5.3.3	Le funzioni dell'app sono ben organizzate e facili da trovare	58
5.3.4	Ho avuto bisogno di aiuto per capire come usare l'app .	59
5.3.5	Mi sono sentito sicuro nell'utilizzare le varie funzioni .	60
5.3.6	Ho dovuto imparare molte cose prima di riuscire ad usare l'app	61
5.3.7	Sintesi dell'usabilità generale	62
5.4	Valutazione delle singole funzionalità	62
5.5	Soddisfazione complessiva e confronto con metodi tradizionali	64

5.5.1	Soddisfazione complessiva	64
5.5.2	Net Promoter Score (NPS)	64
5.5.3	Confronto con metodi tradizionali	65
5.6	Analisi qualitativa e suggerimenti degli utenti	66
5.6.1	Cosa piace di più	66
5.6.2	Cosa migliorare	67
5.6.3	Bug segnalati	67
5.6.4	Funzionalità desiderate	67
5.7	Sintesi dei risultati e migliorie derivate	68
Conclusioni		71
Bibliografia		75

Elenco delle figure

2.1	Diagramma dei casi d'uso	16
2.2	Dashboard Supabase	18
2.3	Tabella accounts in Supabase	18
2.4	Architettura del sistema	21
2.5	Flusso di prenotazione	23
2.6	Diagramma ER del database	26
2.7	Prima versione UI	31
2.8	Flusso aggiunta veicolo	32
2.9	Flusso selezione servizio	32
3.1	Struttura del progetto	36
3.2	Dashboard Ti-Riparo	40
3.3	Interfaccia meccanico	40
3.4	Sito vetrina Ti-Riparo	41
5.1	Frequenza di utilizzo	55
5.2	Canale di acquisizione	56
5.3	Distribuzione delle risposte: Userei Ti-Riparo frequentemente per la manutenzione della mia auto	57
5.4	Distribuzione delle risposte: Ho trovato l'app semplice da usare	58
5.5	Distribuzione delle risposte: Le funzioni dell'app sono ben organizzate e facili da trovare	59
5.6	Distribuzione delle risposte: Ho avuto bisogno di aiuto per capire come usare l'app	59

5.7	Distribuzione delle risposte: Mi sono sentito sicuro nell'utilizzare le varie funzioni	60
5.8	Distribuzione delle risposte: Ho dovuto imparare molte cose prima di riuscire ad usare l'app	61
5.9	Valutazione funzionalità	63
5.10	Distribuzione NPS	64
5.11	Confronto con metodi tradizionali	65
5.12	Funzionalità richieste	68

Elenco delle tabelle

1.1	Confronto tra piattaforme per servizi meccanici	7
4.1	Confronto ricavi B2B vs B2C	48
4.2	Unit economics ottobre 2024	49
4.3	Analisi costi annuali	50
5.1	Distribuzione NPS	65
5.2	Migliorie derivate	69

Capitolo 1

Analisi del contesto

1.1 Settore automotive e digitalizzazione dei servizi

Il mercato italiano della manutenzione e riparazione automobilistica rappresenta un settore di dimensioni rilevanti. Secondo i dati ADIRA, il fatturato netto complessivo del comparto autoriparazione nel 2023 ha raggiunto i 20,8 miliardi di euro, di cui 5,5 miliardi sostenuti direttamente dalle famiglie [1]. L'autoriparazione italiana conta, a fine primo trimestre 2025, circa 89.000 imprese con circa 211.000 addetti [1]. Il settore è caratterizzato da una forte vocazione alla piccola imprenditoria: il 98% degli addetti opera in imprese con meno di 50 lavoratori e l'81,4% in realtà con meno di dieci addetti [3]. Le imprese artigiane costituiscono il 75,9% del totale delle imprese di autoriparazione [4].

Nonostante queste dimensioni considerevoli, il comparto rimane uno dei meno digitalizzati dell'economia dei servizi. La struttura stessa del settore contribuisce a questa resistenza: la maggior parte delle officine italiane sono piccole realtà familiari, spesso a conduzione diretta, dove il rapporto personale con il cliente è considerato prioritario rispetto all'efficienza operativa. Il meccanico di fiducia, figura radicata nell'immaginario collettivo italiano, basa il proprio valore sulla relazione diretta, sulla disponibilità e sulla co-

noscenza storica del veicolo del cliente. In questo contesto, la tecnologia è percepita più come una minaccia che come un'opportunità.

I segnali di cambiamento sono tuttavia evidenti. Le analisi di mercato mostrano una crescita costante dell'interesse online verso i servizi di assistenza, confermata dai trend di ricerca registrati su Google Trends per termini come “meccanico vicino a me”, che dal 2019 evidenziano un aumento rilevante in molte aree urbane. Secondo le indagini sulla mobilità digitale [5], una quota crescente di automobilisti, in particolare gli under 40, preferisce utilizzare canali digitali per prenotare servizi rispetto al contatto telefonico tradizionale. Anche le indagini internazionali sulla customer experience nel post-vendita indicano che una parte significativa dei rispondenti sarebbe disposta a pagare un sovrapprezzo per servizi che eliminano la necessità di recarsi fisicamente in officina, come pick-up, delivery o interventi a domicilio.

La pandemia ha accelerato questa trasformazione. Durante il lockdown, molte officine hanno dovuto improvvisare soluzioni digitali per mantenere il contatto con i clienti. Alcune hanno iniziato a offrire videoconsulenze per diagnosi preliminari, altre hanno implementato sistemi di ritiro e consegna a domicilio. Questi esperimenti, nati dalla necessità, hanno dimostrato che il settore può evolversi senza perdere la propria identità.

Nel complesso, questi dati delineano un mercato in evoluzione, caratterizzato da un'apertura crescente verso soluzioni digitali e da un potenziale ancora parzialmente inespresso per servizi innovativi nel settore della riparazione e manutenzione [2].

Le grandi catene hanno già iniziato a muoversi. Norauto, Euromaster e altre realtà strutturate offrono app di prenotazione e sistemi di gestione digitale, ma si concentrano principalmente su interventi standardizzati come cambio gomme o tagliandi. Manca ancora una soluzione che porti la stessa semplicità e trasparenza alle officine indipendenti, che rappresentano il cuore del mercato italiano. È proprio in questo spazio che si inseriscono le nuove piattaforme digitali, con l'obiettivo di colmare il gap senza snaturare il rapporto di fiducia tra meccanico e cliente.

1.2 Benchmark

Il panorama delle piattaforme digitali per servizi meccanici è ancora frammentato, con pochi player che hanno raggiunto una scala significativa. L'analisi di questi competitor è stata fondamentale per comprendere cosa funziona, cosa manca e dove Ti-Riparo potesse differenziarsi.

1.2.1 Officlick – Il primo tentativo italiano

Officlick è un'officina mobile italiana operata da Ecoprogram Flotte, considerata una delle prime realtà in Italia a portare il concetto di officina “a domicilio”. Secondo il sito ufficiale, il servizio è stato lanciato nel 2018 per anticipare il trend dell'home delivery [6].

A differenza di una classica piattaforma di comparazione preventivi, Officlick non mette in competizione diverse officine: è Ecoprogram stessa a gestire officine mobili attrezzate, inviate direttamente presso il cliente. Il funzionamento prevede che il cliente compili un form con la targa e la localizzazione, dopo di che il customer care contatta l'utente per pianificare l'intervento. Viene quindi inviato un preventivo via email e, al termine dell'intervento, rilasciato un verbale digitale.

Le officine mobili di Officlick sono compatte (circa 20 m², equivalenti a due posti auto) ma completamente equipaggiate per svolgere una gamma ampia di interventi: tagliandi, cambio gomme, smart repair, diagnostica, manutenzione ordinaria e straordinaria.

Per quanto riguarda l'assetto operativo, Officlick ha realizzato numeri rilevanti sulle flotte aziendali: secondo Ecoprogram, nei primi anni ha effettuato oltre 40.000 interventi tramite 22 officine mobili attive su scala nazionale [6].

Dal punto di vista strategico, questo modello presenta punti di forza quali centralizzazione, controllo della qualità e digitalizzazione della prenotazione. Tuttavia emergono anche limiti: non si tratta di un vero marketplace, quindi non esiste comparazione tra differenti officine locali; inoltre, il processo è strettamente legato alle flotte (modello B2B), risultando meno orienta-

to all'utente privato. Infine, Officlick non sembra prevedere un sistema di pagamento completamente integrato online.

In sintesi, Officlick si configura non tanto come un confronto preventivi, quanto come un servizio di “officina mobile” gestito in-house, con un modello che funziona per le flotte aziendali ma risulta meno adatto a clienti privati che cercano un'ampia scelta di officine concorrenti.

1.2.2 Fixter – L'approccio anglosassone

Fixter, fondata nel 2016 e operativa dal 2017 nel Regno Unito, rappresenta un modello più maturo e strutturato. Supportata inizialmente da Kamet Ventures (il braccio di venture building di AXA), la piattaforma ha raccolto circa 14 milioni di dollari in finanziamenti ed è stata acquisita dal Gruppo Renault nel giugno 2022, a testimonianza della validità del modello proposto [7].

L'approccio di Fixter è radicalmente diverso da Officlick. La piattaforma non si limita a intermediare, ma gestisce l'intero processo. Fixter ha stretto partnership con officine selezionate, implementato prezzi trasparenti per interventi standard e, soprattutto, offerto un servizio di pick-up and delivery che elimina il problema logistico per il cliente. L'utente prenota online, un driver ritira l'auto, la porta in officina e la riconsegna a lavoro ultimato. Il servizio è attivo in oltre 100 città nel Regno Unito [7].

L'analisi del modello di business rivela scelte interessanti. I prezzi trasparenti eliminano l'incertezza, elemento critico per conquistare la fiducia dei clienti. La garanzia di 12 mesi su tutti gli interventi trasmette professionalità e affidabilità. Il sistema di aggiornamenti in tempo reale, che permette di seguire lo stato del proprio veicolo, porta trasparenza in un settore tradizionalmente opaco.

Tuttavia, anche Fixter presenta limiti. Il modello pick-up and delivery funziona bene in città dense come Londra, ma diventa economicamente meno sostenibile in aree meno urbanizzate. I costi operativi sono elevati: due trasferimenti per ogni intervento raddoppiano i tempi morti. Inoltre, la

standardizzazione dei prezzi, se da un lato semplifica, dall'altro può risultare poco competitiva per interventi semplici dove il prezzo di mercato è inferiore.

1.2.3 ClickMechanic – Il marketplace puro

ClickMechanic, anch'essa britannica, ha scelto un approccio ancora diverso: il marketplace puro per meccanici mobili. La piattaforma mette in contatto automobilisti con meccanici che si spostano per effettuare riparazioni a domicilio. Con oltre 1.000 meccanici attivi e copertura nazionale, il modello ha dimostrato una certa capacità di scalare.

I punti di forza sono evidenti: l'eliminazione totale del problema logistico, poiché l'auto non si muove ed è il meccanico a spostarsi; i costi ridotti, dovuti all'assenza di una sede fisica, che si traducono in prezzi competitivi; e la flessibilità degli orari, con interventi anche nel weekend.

Emergono tuttavia limitazioni significative. Non tutti gli interventi possono essere effettuati a domicilio: operazioni che richiedono ponte o attrezzature specifiche sono escluse. Il controllo qualità è complesso quando si hanno centinaia di operatori indipendenti. La stagionalità rappresenta un ulteriore problema: lavorare all'aperto in inverno non è ideale, specialmente nel clima britannico.

1.2.4 Altri player minori

Durante la ricerca sono state analizzate anche realtà minori ma interessanti. YourMechanic negli Stati Uniti ha raccolto oltre 50 milioni di dollari ma ha faticato a raggiungere la profittabilità. Autobutler in Danimarca si concentra sul segmento premium con buoni margini ma volumi limitati. In Italia, oltre a Officlick, esistono tentativi locali come CercaOfficina o MeccanicoDomicilio, ma nessuno ha raggiunto massa critica.

1.2.5 Lezioni apprese dall'analisi

Dall'analisi dei competitor emergono pattern chiari. I modelli che funzionano risolvono il problema end-to-end, senza limitarsi a digitalizzare un singolo passaggio del processo. La fiducia è fondamentale: garanzie, recensioni verificate e prezzi trasparenti sono elementi essenziali. Il controllo qualità non può essere delegato completamente: serve un bilanciamento tra scalabilità e standard di servizio. La sostenibilità economica è critica: molte startup del settore consumano capitale senza trovare un modello profittevole.

Ti-Riparo nasce da una constatazione semplice: ogni modello esistente risolve un pezzo del problema, ma ne ignora un altro. Il servizio mobile elimina l'ostacolo logistico, è il tecnico che va dal cliente, non il contrario. Partire con un team interno, anziché affidarsi subito a una rete di partner, permette di tenere sotto controllo ogni riparazione. Concentrarsi su un'area geografica definita non è un limite, è una scelta: serve densità operativa perché i numeri funzionino. E una piattaforma tecnologica snella tiene i costi fissi al minimo, evitando di bruciare capitale prima di aver validato il modello. In sintesi: partire piccoli, crescere solo quando ha senso farlo.

1.2.6 Analisi comparativa e identificazione del gap

La Tabella 1.1 sintetizza le caratteristiche principali delle piattaforme analizzate, evidenziando le aree coperte e scoperte da ciascun competitor.

Dal confronto emergono tre gap significativi che nessuna soluzione esistente colma simultaneamente.

In primo luogo, l'assenza di una piattaforma mobile nativa per il mercato italiano B2C. Officlick opera esclusivamente con form web e customer care telefonico, rivolgendosi principalmente a flotte aziendali. Non esiste in Italia un'app mobile che permetta a un utente privato di prenotare un intervento meccanico a domicilio con la stessa immediatezza con cui si ordina un pasto o si prenota un taxi.

Caratteristica	Officlick	Fixter	ClickMechanic	Ti-Riparo
Servizio a domicilio	Sì	No (pick-up)	Sì	Sì
App mobile nativa	No	Sì	Sì	Sì
Target B2C privati	No (B2B)	Sì	Sì	Sì
Prezzi trasparenti in-app	No	Sì	Sì	Sì
Ricerca automatica targa	No	No	No	Sì
Preventivi in-app	No	No	Sì	Sì
Fatturazione digitale	No	No	No	Sì
Pagamento digitale	No	Sì	Sì	In roadmap
Operativo in Italia	Sì	No (UK)	No (UK)	Sì
Costo infrastruttura ~ 0	No	No	No	Sì

Tabella 1.1: Analisi comparativa delle piattaforme per servizi meccanici digitali

In secondo luogo, la frammentazione del ciclo di servizio. Anche nelle piattaforme più mature come Fixter e ClickMechanic, la gestione documentale (preventivi dettagliati, fatturazione digitale) resta separata dal flusso di prenotazione. L'utente riceve conferme via email, preventivi in PDF allegati, fatture da richiedere separatamente. Nessuna piattaforma analizzata integra l'intero ciclo, dalla richiesta alla fattura, in un'unica interfaccia mobile.

In terzo luogo, la sostenibilità tecnologica per operatori indipendenti. Le soluzioni esistenti richiedono investimenti significativi in infrastruttura tecnologica oppure dipendono da piattaforme proprietarie con costi ricorrenti elevati. Manca un modello che dimostri come un singolo sviluppatore, utilizzando strumenti open source e tier gratuiti di servizi cloud, possa costruire un ecosistema completo (app mobile, dashboard gestionale, sito web) con costi di infrastruttura prossimi allo zero.

1.3 Valutazione della concorrenza locale

Bologna rappresenta un caso di studio ideale per testare un servizio innovativo nel settore automotive. Con circa 390.000 abitanti e un'area metropolitana che supera il milione, la città combina dimensioni sufficienti per garantire domanda con una complessità gestibile per una fase pilota. Secondo i dati del report “I Numeri di Bologna Metropolitana” [8], il parco veicolare dell'area metropolitana è in crescita.

Per quanto riguarda le officine, esistono decine di centri assistenza a Bologna e provincia; ad esempio, Virgilio riporta circa 149 autofficine e centri assistenza nella zona [9]. Tuttavia, non è stato possibile reperire dati pubblici attendibili che confermino una suddivisione precisa tra officine indipendenti, catene e officine autorizzate, né una statistica sul numero di dipendenti per officina.

1.3.1 Le officine tradizionali

Durante la fase di ricerca sono state visitate personalmente oltre 10 officine nell'area bolognese con l'obiettivo di comprendere il loro approccio al servizio. L'esperienza è stata significativa. La maggior parte degli esercizi opera ancora secondo modalità consolidate da decenni: il cliente telefona per fissare un appuntamento, porta l'auto la mattina presto, attende l'intera giornata senza aggiornamenti e ritira il veicolo la sera.

I prezzi variano in modo considerevole. Per un semplice cambio olio, sono stati ricevuti preventivi compresi tra 45 e 120 euro, senza che fosse chiaro cosa giustificasse tale differenza. La trasparenza risulta limitata: raramente viene fornito un preventivo scritto dettagliato, i tempi sono approssimativi, e il prezzo finale include spesso “imprevisti” scoperti durante l'intervento.

1.3.2 Le catene e i fast service

Le catene come Norauto, Formula Driver e Euromaster rappresentano il tentativo più strutturato di modernizzazione del settore. Offrono prezzi fissi

per interventi standard, possibilità di prenotazione online (anche se spesso il sistema rimanda comunque a una chiamata telefonica), e tempi di attesa generalmente più brevi. Tuttavia, il servizio risulta impersonale e limitato a operazioni di routine.

Dalla verifica diretta del servizio di diverse catene, l'esperienza, seppur più prevedibile rispetto alle officine tradizionali, presenta ancora limiti. Le attese sono lunghe, specialmente il sabato. Il personale, spesso giovane e con turnover elevato, non sempre dispone dell'esperienza necessaria per diagnosi complesse. I prezzi, pur essendo trasparenti, non risultano particolarmente competitivi se si considera il livello di servizio offerto.

1.3.3 Tentativi di innovazione locale

Nel territorio bolognese sono emersi negli anni vari tentativi minori di innovazione nel settore dell'assistenza meccanica e della prenotazione online, spesso iniziative di operatori locali senza diffusione significativa. Alcuni progetti sperimentali, focalizzati sul servizio a domicilio o sulla digitalizzazione delle officine indipendenti, non hanno tuttavia superato la fase iniziale, spesso a causa di un valore aggiunto limitato o dell'assenza di un modello scalabile.

In assenza di piattaforme consolidate, Bologna rappresenta ancora oggi un contesto adatto a nuove soluzioni digitali realmente integrate.

1.3.4 Opportunità e posizionamento di Ti-Riparo

L'assenza di servizi a domicilio professionali lascia scoperto un segmento di clienti, professionisti, famiglie con bambini piccoli, anziani con difficoltà di spostamento, disposti a pagare per la comodità. La mancanza di trasparenza sui prezzi genera sfiducia e domanda repressa: molti clienti rimandano interventi proprio per l'incertezza sui costi. I tempi di attesa indefiniti rappresentano un ulteriore pain point in un'epoca in cui è possibile tracciare in tempo reale qualsiasi consegna.

In questo contesto, Ti-Riparo si posiziona su un segmento definito: interventi di manutenzione ordinaria e riparazioni medie effettuabili a domicilio, per clienti che valorizzano tempo e comodità. Il vantaggio competitivo principale risiede nell’eliminazione del problema logistico: è il servizio a raggiungere l’auto, non il contrario. Dimostrare che molti interventi possono essere svolti con professionalità anche in mobilità richiede tuttavia un investimento significativo in comunicazione e costruzione della fiducia.

Bologna, con le sue caratteristiche demografiche ed economiche, rappresenta il terreno di prova ideale per validare questo modello prima di un’eventuale espansione.

1.4 Contributo della tesi

L’analisi dello stato dell’arte, sia a livello internazionale (Sezione 1.2) sia locale (Sezione 1.3), evidenzia un problema comune: non esiste nel mercato italiano una soluzione che combini simultaneamente servizio meccanico a domicilio, applicazione mobile nativa per utenti privati, gestione digitale end-to-end del ciclo di servizio e un modello tecnologico replicabile a costo ridotto.

Il contributo di questa tesi si articola su tre livelli. Sul piano progettuale, il lavoro documenta l’intero processo dalla definizione dei requisiti alla pubblicazione sugli store, integrando sviluppo software, analisi di mercato e gestione operativa. Sul piano tecnologico, presenta un’architettura completa basata su React Native, Supabase e servizi cloud a costo zero, dimostrando che un ecosistema professionale può essere realizzato da un singolo sviluppatore con risorse minime. Sul piano della validazione, include la misurazione dell’usabilità attraverso un questionario ispirato al SUS [10] somministrato a utenti reali, chiudendo il ciclo progettazione-implementazione-valutazione.

1.5 Obiettivi del progetto

Partendo dai gap identificati nelle sezioni precedenti, il progetto si articola attorno a cinque obiettivi:

1. Sviluppare un'applicazione mobile funzionante per la prenotazione di servizi meccanici a domicilio, coprendo l'intero ciclo preventivo-prenotazione-fatturazione.
2. Dimostrare la fattibilità tecnica di un ecosistema completo (app, dashboard, sito web) realizzato con stack open source e costi di infrastruttura prossimi allo zero.
3. Validare il modello operativo con utenti reali nell'area metropolitana di Bologna.
4. Misurare l'usabilità della piattaforma attraverso un questionario strutturato ispirato al System Usability Scale [10].
5. Analizzare la sostenibilità del modello di business, identificando criticità e strategie di evoluzione.

L'architettura è stata inoltre progettata in ottica di crescita futura verso un modello marketplace, in cui altre officine possano offrire i propri servizi attraverso la piattaforma.

1.6 Metodologia di lavoro

Il percorso di sviluppo di Ti-Riparo non ha seguito un andamento lineare. La necessità di conciliare lo sviluppo con impegni lavorativi paralleli ha richiesto un adattamento delle metodologie classiche ai vincoli temporali del progetto. L'approccio adottato è stato necessariamente pragmatico: invece di seguire rigidamente sprint di due settimane come previsto dalle metodologie Agile, si è lavorato per obiettivi mensili più flessibili.

1.6.1 Pianificazione e prototipazione

La fase iniziale, durata circa due mesi, è stata fondamentale per definire l'ambito del prodotto. Si è proceduto attraverso sessioni di brainstorming, schizzi e prototipi realizzati su Figma. L'obiettivo era ottenere una visione chiara del prodotto finale senza investire tempo eccessivo in documentazione destinata a cambiare durante lo sviluppo.

1.6.2 Sviluppo e versionamento

Lo sviluppo si è protratto per circa sei mesi, con un ritmo discontinuo dettato dalla disponibilità di tempo. Alcuni mesi sono stati dedicati interamente a una singola funzionalità complessa, come il sistema di autenticazione OAuth, mentre in altri periodi è stato possibile avanzare su più fronti contemporaneamente.

Il versionamento con Git è stato essenziale per mantenere ordine. Ogni avanzamento veniva committato con descrizioni dettagliate, permettendo di riprendere il lavoro anche dopo pause di diversi giorni senza perdere il filo. I branch sono stati usati principalmente per sperimentazione: venivano creati per testare idee che poi venivano abbandonate o integrate nel main a seconda del risultato.

1.6.3 Testing e deployment

Il testing è avvenuto in modo incrementale. Non disponendo di risorse per un team di QA, sono stati coinvolti amici e familiari come beta tester. Questi momenti si sono rivelati preziosi: osservare persone reali utilizzare l'applicazione ha evidenziato problemi di usabilità che non sarebbero stati identificati altrimenti.

La fase di deployment è stata particolarmente intensa. Dopo mesi di sviluppo in locale, la pubblicazione sugli store ha richiesto di affrontare numerosi aspetti tecnici e procedurali. I primi bug in produzione sono emersi rapidamente, ma la struttura di aggiornamenti over-the-air implementata con Expo

ha permesso di risolvere le criticità in tempi brevi, senza attendere i tempi di revisione degli store.

1.6.4 Approccio data-driven

Un aspetto fondamentale della metodologia è stato l'approccio data-driven alle decisioni. Fin dall'inizio è stato implementato un sistema di analytics che traccia ogni interazione significativa. Questi dati hanno guidato le priorità di sviluppo: funzionalità che sembravano essenziali si sono rivelate poco utilizzate, mentre altre aggiunte quasi per caso sono diventate centrali nell'esperienza utente.

Nel complesso, questa metodologia adattiva ha permesso di mantenere flessibilità operativa senza compromettere il raggiungimento degli obiettivi, privilegiando un approccio pratico orientato al risultato.”

Capitolo 2

Architettura del sistema

2.1 Requisiti funzionali

Prima di descrivere le scelte tecnologiche, è utile definire le funzionalità offerte dalla piattaforma ai tre attori principali: cliente, meccanico e amministratore. La Figura 2.1 sintetizza i casi d'uso del sistema.

Il cliente può registrarsi, aggiungere veicoli tramite targa, richiedere preventivi, prenotare interventi selezionando data e ora, e consultare ordini e fatture. Il meccanico gestisce gli ordini assegnati, formula preventivi personalizzati, invia fatture e amministra la propria disponibilità. L'amministratore supervisiona l'intero sistema, gestisce il catalogo servizi e monitora le statistiche operative.

2.2 Scelta dello stack tecnologico

La selezione delle tecnologie per Ti-Riparo è stata guidata da considerazioni pratiche. Dovendo sviluppare l'applicazione da solo, con tempo limitato e budget praticamente nullo, ogni scelta doveva massimizzare il rapporto tra risultato e sforzo richiesto.



Figura 2.1: Diagramma dei casi d'uso di Ti-Riparo

2.2.1 React Native con Expo

La decisione di utilizzare React Native attraverso Expo è arrivata dopo aver valutato diverse alternative, tra cui Flutter e Swift [13]. La conoscenza pregressa di JavaScript e React ha reso React Native la scelta più naturale per accelerare lo sviluppo.

Expo ha rappresentato un fattore decisivo. Invece di dedicare settimane alla configurazione di ambienti di sviluppo nativi, build pipeline e certificati, è stato possibile concentrarsi direttamente sul codice dell'applicazione. Il

sistema di sviluppo con Expo Go ha permesso un ciclo di iterazione estremamente rapido: le modifiche al codice venivano riflesse istantaneamente sul dispositivo fisico, senza necessità di compilazioni o deploy intermedi [14].

La scelta di Expo SDK 50 (poi aggiornato alla 55) ha portato vantaggi concreti. L'integrazione nativa con molte funzionalità del dispositivo — camera, notifiche push, geolocalizzazione — era già pronta all'uso. Il sistema di Over-The-Air updates si è rivelato fondamentale: è stato possibile correggere bug critici in produzione in minuti invece che giorni. EAS Build ha semplificato la generazione delle build per gli store, processo che altrimenti avrebbe richiesto competenze specifiche per iOS e Android.

I compromessi principali riguardano le dimensioni del bundle (circa 25 MB in più rispetto a un'app nativa equivalente, dovuti al runtime JavaScript) e la difficoltà di accedere a moduli nativi non supportati da Expo senza ricorrere a un development build personalizzato. Per un MVP orientato alla validazione del concept, questi trade-off sono stati ritenuti accettabili.

2.2.2 Supabase

La scelta di Supabase come backend è nata dalla ricerca di un'alternativa a Firebase che offrisse maggiore apertura e flessibilità [15]. Supabase si presenta come una piattaforma open source costruita su PostgreSQL, e nella pratica si è rivelata una soluzione significativamente più completa di quanto inizialmente previsto.

L'adozione di PostgreSQL come database sottostante è stata determinante. A differenza dei database NoSQL proprietari, Supabase consente di utilizzare SQL standard, linguaggio già acquisito durante i corsi di basi di dati del percorso universitario. La possibilità di definire relazioni, vincoli e trigger direttamente a livello di database ha permesso di implementare logiche complesse senza la necessità di un layer backend separato.

Le Edge Functions hanno sostituito completamente la necessità di un server tradizionale. Scritte in TypeScript e deployate direttamente tramite Supabase, gestiscono l'intera business logic critica del sistema: la ricerca veicolo

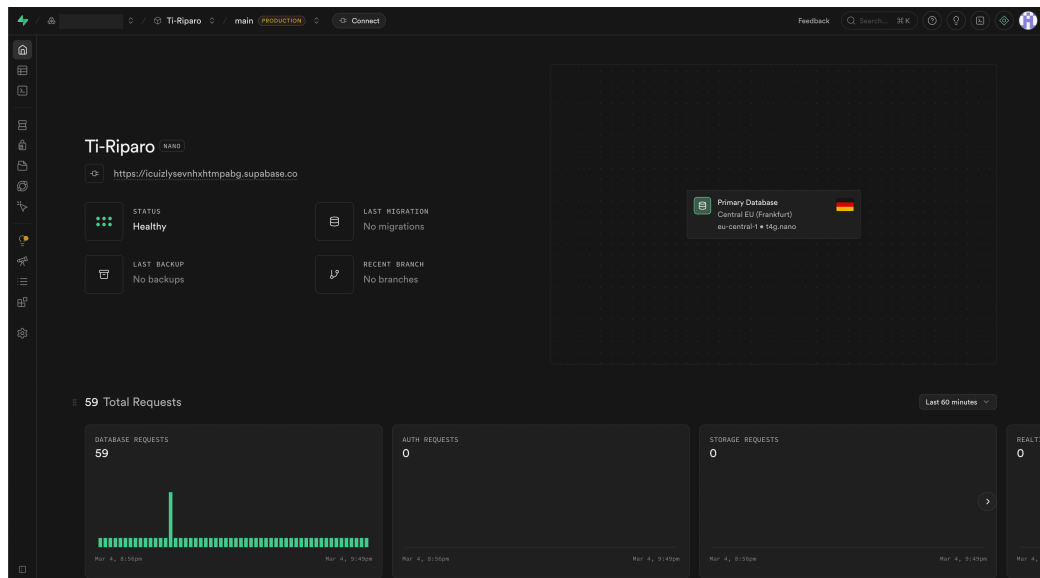


Figura 2.2: Pannello principale della dashboard Supabase del progetto Ti-Riparo

The screenshot shows the Supabase Table Editor for the 'accounts' table. The table structure is defined with columns: id (integer), created_at (timestamp), full_name (text), email (text), phone (text), account_type (text), and account_type_amount (integer). The data view shows a list of 20 records, each representing a user account with their ID, creation timestamp, name, email, phone number, and account type.

id	created_at	full_name	email	phone	account_type	account_type_amount
216	2025-09-24 18:39:10.374438+00	Feliciano			user	NULL
217	2025-09-26 13:35:03.971913+00	Armando			user	NULL
218	2025-09-30 14:23:04.039679+00	Ionut Mu			user	NULL
219	2025-09-30 21:44:12.542834+00	Caterina I			user	NULL
220	2025-10-02 12:58:07.665437+00	Daniele T.			user	NULL
221	2025-10-03 15:50:09.163986+00	Francesco			user	NULL
222	2025-10-03 20:37:38.184739+00	Singh Ma			user	NULL
223	2025-10-03 21:53:18.598546+00	Lorenzo V.			user	NULL
224	2025-10-07 09:18:29.354396+00	Giada De			user	NULL
225	2025-10-07 14:21:39.465088+00	Giuseppe			user	NULL
226	2025-10-13 04:43:24.336213+00	Filippo Co			user	NULL
227	2025-10-13 04:59:58.156257+00	Annamari			user	NULL
228	2025-10-13 05:56:21.245814+00	Senena Be			user	NULL
229	2025-10-13 07:44:09.593311+00	Veronica			user	NULL
230	2025-10-13 09:22:06.358079+00	Elias gott			user	NULL
231	2025-10-13 10:43:58.710785+00	Alessio V.			user	NULL
232	2025-10-13 18:06:03.080534+00	Mario De			user	NULL
233	2025-10-13 19:17:07.179354+00	Enrico Su			user	NULL
234	2025-10-13 19:22:49.770254+00	Ayoub Je			user	NULL
235	2025-10-14 05:07:48.122221+00	Nicola Pa			user	NULL
236	2025-10-14 07:27:56.037821+00	Michele E			user	NULL

Figura 2.3: Struttura della tabella accounts nel database PostgreSQL di Supabase

tramite targa, l'invio di email transazionali e la gestione dei pagamenti attraverso Stripe. L'esecuzione su runtime Deno garantisce prestazioni elevate e un ambiente di esecuzione sicuro.

Il sistema di autenticazione integrato ha semplificato notevolmente uno degli aspetti più critici dello sviluppo. Supabase Auth supporta nativamente il login tramite Magic Link, l'autenticazione OAuth con Google e Apple, e la gestione automatica delle sessioni. A livello di sicurezza, Row Level Security (RLS) di PostgreSQL garantisce l'isolamento dei dati per utente, implementando il controllo degli accessi direttamente a livello di database anziché a livello applicativo.

2.2.3 Stripe

Per la gestione dei pagamenti, la scelta di Stripe è stata naturale. Implementare un sistema di pagamenti internamente avrebbe richiesto la conformità allo standard PCI-DSS, la gestione diretta delle carte di credito, protocolli di sicurezza dedicati e la gestione delle dispute. Stripe astrae l'intera complessità attraverso API ben documentate e affidabili.

L'integrazione si articola su tre componenti principali: Stripe Elements per l'interfaccia di inserimento dei dati di pagamento, Stripe Connect per la gestione dei pagamenti verso le future officine partner, e un sistema di webhook per la sincronizzazione in tempo reale dello stato delle transazioni con il backend.

2.2.4 Tecnologie di supporto

Oltre allo stack principale, diverse tecnologie complementari hanno reso possibile Ti-Riparo nella sua forma attuale.

Resend è stato adottato per l'invio di email transazionali, sostituendo configurazioni SMTP complesse con API REST semplici. Il piano gratuito (3.000 email al mese) si è rivelato più che adeguato per le esigenze iniziali,

offrendo anche gestione automatica di bounce e complaint, oltre a template React per email responsive.

OpenAPI Automotive è un servizio italiano che permette di ottenere le informazioni di un veicolo a partire dalla targa: marca, modello, anno, cilindrata. L'utente inserisce la targa e l'app compila automaticamente tutti i dati del veicolo. Il costo di 0,10€ per ricerca è minimo rispetto al valore percepito dall'utente.

Register.it gestisce il dominio ti-riparo.it. La gestione DNS, inclusi i record MX per le email, i CNAME per i sottodomini e i TXT per le verifiche dei servizi, richiede una configurazione accurata che Register.it rende gestibile attraverso un'interfaccia semplice.

Per il design, la combinazione di Figma per i mockup e NativeWind (TailwindCSS per React Native) ha accelerato l'iterazione UI/UX, permettendo di applicare classi di utilità familiari dal mondo del web development.

Expo Application Services (EAS) merita una menzione specifica. Build cloud per iOS senza necessità di un Mac, deployment automatizzato e Over-The-Air updates hanno trasformato un processo di DevOps potenzialmente complesso in un flusso gestibile. Per la fase MVP è stato utilizzato il piano gratuito, sufficiente per generare build e pubblicare aggiornamenti.

2.2.5 Un ecosistema enterprise senza costi

Grazie ai piani gratuiti dei servizi descritti, il costo operativo dell'intero ecosistema è stato praticamente nullo nella fase iniziale. Si tratta di una dimostrazione concreta di come l'attuale panorama SaaS permetta anche a un singolo sviluppatore di costruire prodotti che fino a pochi anni fa avrebbero richiesto team dedicati e budget elevati. Per validare un'idea e costruire un MVP, questo approccio si è rivelato estremamente efficace.

2.3 Architettura generale

L'architettura di Ti-Riparo segue un pattern client-server moderno, con alcune peculiarità dettate dalle tecnologie scelte e dai requisiti specifici del dominio.

2.3.1 Overview architetturale

Il sistema si compone di tre layer principali: il client mobile, il backend Supabase e i servizi esterni integrati. La comunicazione avviene principalmente attraverso API REST e WebSocket per le funzionalità real-time.

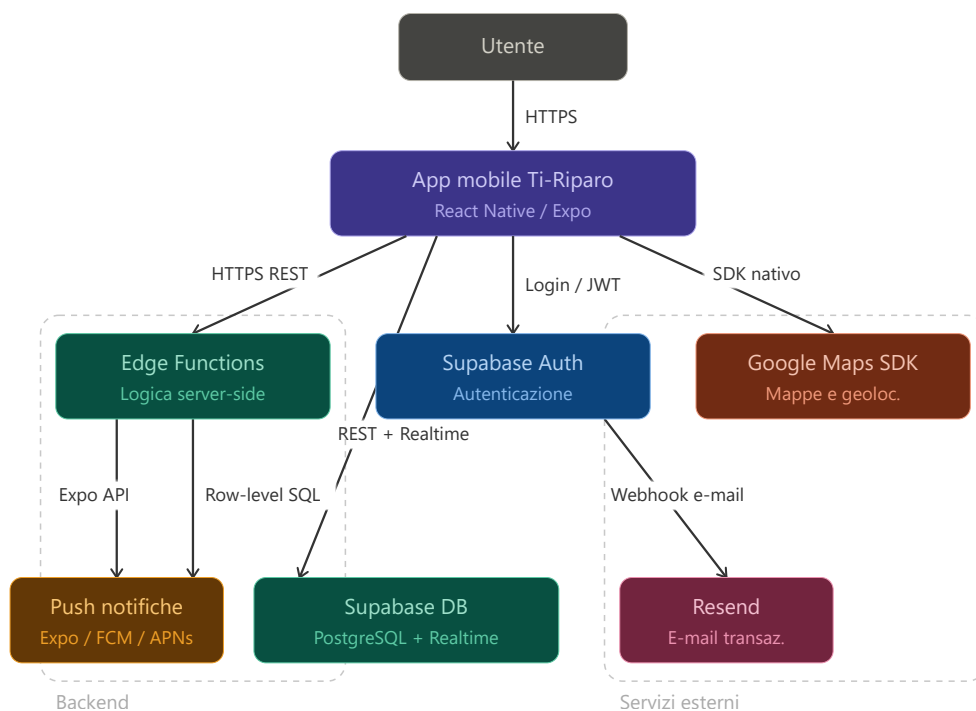


Figura 2.4: Diagramma dell'architettura generale di Ti-Riparo

2.3.2 Flusso dati principale

Il client mobile, costruito con React Native/Expo, è responsabile dell'interfaccia utente e della logica di presentazione. Mantiene uno stato locale minimale, sincronizzando costantemente con il backend per garantire consistenza. L'approccio "optimistic UI" migliora la responsività: le azioni dell'utente vengono riflesse immediatamente nell'interfaccia, mentre in background avviene la sincronizzazione con il server.

Il backend Supabase fornisce quattro servizi fondamentali: database PostgreSQL per la persistenza, autenticazione per la gestione utenti, storage per immagini e documenti, e Edge Functions per la business logic. Questi componenti comunicano internamente attraverso trigger e function PostgreSQL, garantendo consistenza transazionale.

Il flusso di una prenotazione tipica illustra l'interazione tra i componenti. L'utente apre l'app e si autentica: Supabase verifica le credenziali e restituisce un JWT token che il client utilizzerà per tutte le richieste successive. Il token ha una durata di un'ora e si rinnova automaticamente finché l'utente è attivo.

Quando l'utente richiede un preventivo, il client raccoglie le informazioni (tipo di intervento, foto o descrizione del problema, posizione dell'auto e informazioni sul veicolo) e le invia a una Edge Function. Questa valida i dati, calcola il prezzo base consultando la tabella dei servizi, applica eventuali maggiorazioni (urgenza, distanza, weekend) e salva il preventivo nel database. Un trigger PostgreSQL notifica il sistema di amministrazione che processa la richiesta.

Se il preventivo viene accettato, si attiva il flusso di pagamento. Il client richiede a Stripe la creazione di un Payment Intent, riceve un client secret e presenta l'interfaccia di pagamento. Quando l'utente inserisce i dati della carta, Stripe processa il pagamento e notifica il backend attraverso webhook. Una Edge Function aggiorna lo stato della prenotazione e invia conferme via email e notifica push.

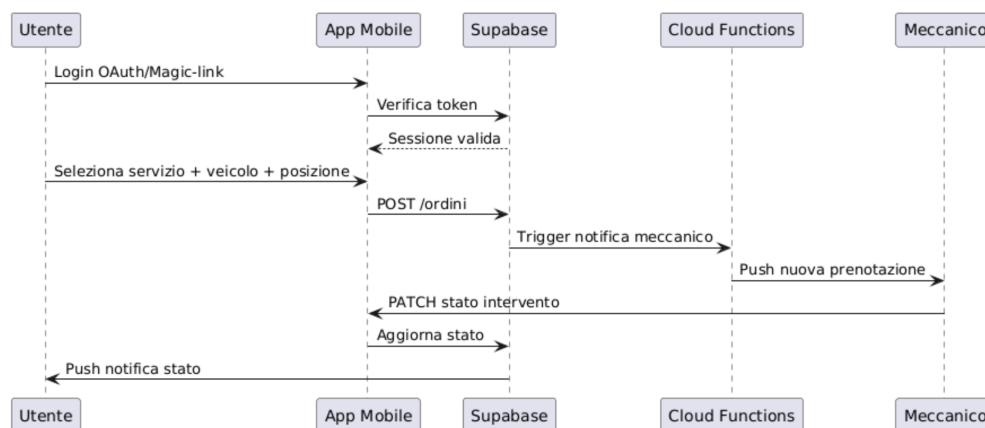


Figura 2.5: Diagramma di sequenza del flusso di prenotazione

2.3.3 Gestione dello stato

La gestione dello stato adotta un approccio misto. Redux, tramite Redux Toolkit, è utilizzato esclusivamente per lo stato di autenticazione: sessione utente, dati del profilo e flag di onboarding. La scelta di centralizzare queste informazioni in uno store globale è motivata dal fatto che la sessione deve essere accessibile da qualsiasi punto dell'applicazione, sia per le chiamate API (che richiedono il JWT token) sia per il rendering condizionale dell'interfaccia in base al ruolo dell'utente.

Per il resto dello stato applicativo, ordini, veicoli, preventivi, fatture, non è stato introdotto un layer di state management globale. I dati vengono recuperati direttamente da Supabase tramite query nei singoli componenti, utilizzando gli hook nativi di React (`useState`, `useEffect`). Questo approccio, più semplice e con meno codice boilerplate, si è rivelato adeguato per la scala attuale dell'applicazione. Un eventuale passaggio a una soluzione di caching come React Query o SWR è previsto qualora la crescita del numero di schermate e di query parallele rendesse necessaria una gestione più strutturata di cache, refetch e stati di caricamento.

La Context API è utilizzata per ambiti circoscritti, come la gestione del

caricamento dei font personalizzati, dove uno store Redux sarebbe sovradimensionato.

2.3.4 Sicurezza e autorizzazione

La sicurezza è implementata a più livelli. A livello di rete, tutte le comunicazioni avvengono su HTTPS con certificate pinning per prevenire attacchi man-in-the-middle. I JWT token hanno vita breve e vengono rinnovati frequentemente.

```
1 ALTER POLICY "select order only authenticated or
   Mechanic"
2 ON "public"."order"
3 TO authenticated
4 USING (
5     (EXISTS (
6         SELECT 1 FROM accounts
7         WHERE accounts.auth_user_id = (SELECT auth.uid())
8         AND accounts.account_type =
9             'mechanic'::account_type_enum
10    ))
11 OR
12 ((SELECT auth.uid()) = (
13     SELECT accounts.auth_user_id
14     FROM accounts
15     WHERE accounts.id = "order".account_id
16 ))
17 );
```

Listing 2.1: Policy Row Level Security per la tabella ordini

A livello di database, Row Level Security (RLS) garantisce l'isolamento dei dati. Ogni tabella ha policy che limitano l'accesso in base all'utente autenticato. Un utente può visualizzare solo le proprie prenotazioni, mentre un meccanico accede esclusivamente a quelle assegnate a lui.

Le Edge Functions validano ogni richiesta, verificando non solo l'autenticazione ma anche l'autorizzazione per l'azione specifica. Un utente non può modificare prenotazioni altrui, né accedere a funzionalità di amministrazione anche conoscendo gli endpoint.

2.3.5 Scalabilità e performance

L'architettura è progettata per scalare orizzontalmente. Supabase gestisce automaticamente il load balancing e lo scaling del database. Le Edge Functions sono stateless e possono essere replicate senza limiti. Il client mobile utilizza caching aggressivo sui dati statici, riducendo le richieste al server.

Con circa 200 utenti, il sistema non ha evidenziato colli di bottiglia. Non sono stati tuttavia condotti load test formali, e una crescita verso migliaia di utenti richiederebbe una valutazione specifica, in particolare sulla gestione delle connessioni WebSocket per le subscription real-time.

2.4 Modello dei dati

Il database PostgreSQL di Ti-Riparo è organizzato attorno a cinque entità principali, le cui relazioni sono illustrate nella Figura 2.6.

La tabella **accounts** estende le informazioni di autenticazione gestite da **auth.users** con dati applicativi: nome, telefono, indirizzo, token Expo per le notifiche push e tipo di account (cliente, meccanico, amministratore). La tabella **cars** memorizza i veicoli associati a ciascun utente, con i dati ottenuti dalla ricerca per targa (marca, modello, anno, cilindrata). La tabella **orders** rappresenta il cuore del sistema, collegando utente, veicolo, servizio richiesto, posizione, data, stato e riferimenti a preventivi e fatture. Le tabelle **quotes** e **invoices** gestiscono rispettivamente i preventivi e le fatture, ciascuna con il dettaglio delle voci di costo e i riferimenti ai documenti PDF archiviati su Supabase Storage.

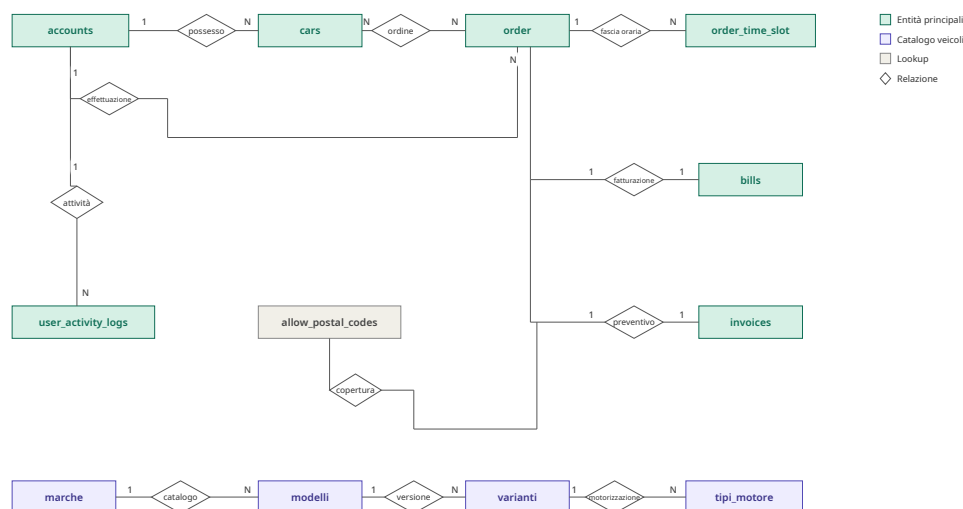


Figura 2.6: Diagramma Entity-Relationship del database Ti-Riparo

Le relazioni tra le entità seguono un modello relazionale classico: un account può avere più veicoli (uno-a-molti), un account può generare più ordini (uno-a-molti), ogni ordine è associato a un veicolo, e ogni ordine può avere un preventivo e una fattura (uno-a-uno).

2.5 Descrizione dei moduli principali

2.5.1 Autenticazione e gestione utenti

Il modulo di autenticazione è il cuore della sicurezza di Ti-Riparo. Costruito sopra Supabase Auth, gestisce registrazione, login, Magic Link, OAuth (Apple, Google) e gestione profili con attenzione particolare all'esperienza utente e alla sicurezza.

La registrazione segue un flusso ottimizzato per ridurre l'abbandono. Inizialmente venivano richieste molte informazioni: nome, cognome, telefono,

indirizzo, modello auto. Il tasso di completamento era molto basso. Il form è stato quindi ridotto a soli email e nome, rimandando la raccolta di ulteriori informazioni a quando l'utente è già all'interno dell'applicazione.

Il sistema supporta due modalità di autenticazione: Magic Link e OAuth con Google e Apple. Quest'ultima è particolarmente diffusa, rappresentando il 60% delle registrazioni. La parte più complessa è stata la gestione del deep linking per il Magic Link: intercettare il link, attivare l'utente e completare automaticamente la creazione della sessione.

La gestione del profilo utente è strutturata su due livelli. La tabella `auth.users` di Supabase contiene le credenziali e i metadati di sistema. La tabella `public.accounts` contiene i dati applicativi: nome, telefono, indirizzo, token Expo per le notifiche push e altri dati necessari al funzionamento. Questa separazione permette di mantenere il sistema di autenticazione generico personalizzando al contempo il dominio applicativo.

La gestione dei ruoli prevede tre tipologie di utenti: clienti, meccanici e amministratori. È stato adottato un approccio semplice: un campo `account_type` nel profilo determina cosa l'utente può visualizzare e quali azioni può compiere. Le policy RLS e le Edge Functions rispettano questi ruoli, garantendo la separazione delle responsabilità.

La sessione utente è gestita attraverso refresh token con rotazione automatica. All'accesso, l'utente riceve un access token (valido 1 ora) e un refresh token (valido 30 giorni). L'app rinnova silenziosamente l'access token quando necessario. Alla scadenza del refresh token, l'utente deve ri-autenticarsi.

2.5.2 Gestione prenotazioni e calendario

Il sistema di prenotazioni bilancia le richieste degli utenti con la disponibilità dei meccanici, ottimizza i percorsi per minimizzare gli spostamenti, e gestisce modifiche e cancellazioni.

Il calendario utilizza una struttura a slot temporali. Ogni giornata è divisa in fasce orarie di 3-4 ore (8-12, 14-17 e simili). Quando un utente prenota,

seleziona uno slot disponibile. Il sistema verifica l'assenza di sovrapposizioni e conferma la prenotazione o suggerisce alternative.

La gestione delle disponibilità è dinamica. I meccanici possono contrassegnare slot come non disponibili per ferie, malattia o impegni personali. Il sistema aggiorna automaticamente le disponibilità mostrate agli utenti. In caso di indisponibilità improvvisa, le prenotazioni vengono riassegnate oppure l'utente viene notificato per riprogrammare.

Le modifiche e cancellazioni seguono policy precise. Cancellazione gratuita fino a 24 ore prima dell'intervento, 50% di penale tra 24 e 2 ore, 100% sotto le 2 ore. Queste policy sono automatizzate: il sistema calcola eventuali rimborsi e li processa attraverso Stripe. Le modifiche di orario sono gratuite se lo slot desiderato è disponibile.

2.5.3 Sistema di preventivi e fatture

Il modulo preventivi e fatture è stato ripensato diverse volte prima di trovare il giusto equilibrio tra automazione e controllo manuale. La versione attuale riflette la realtà operativa di Ti-Riparo: un servizio gestito da un meccanico professionista che mantiene il controllo sulla qualità e sui prezzi, supportato dalla tecnologia per velocizzare i processi.

Il catalogo servizi è il punto di partenza. È stata definita una lista di interventi standard con prezzi fissi: diagnosi computerizzata (50€), cambio olio e filtri (da 80€), tagliando completo (da 150€), ricarica clima (90€), sostituzione batteria (30€ più costo della batteria), e altri interventi comuni. Ogni servizio ha una descrizione dettagliata che specifica cosa include, quanto tempo richiede e eventuali note importanti. Questa standardizzazione elimina l'incertezza: l'utente conosce immediatamente la spesa prevista.

Per problemi non catalogati, l'utente può richiedere un preventivo personalizzato. In questo caso, il meccanico riceve una notifica e valuta la richiesta. Attraverso la vista dedicata, può consultare i dettagli del veicolo (marca, modello, anno, storico interventi), visualizzare eventuali foto del problema e formulare un preventivo specifico.

L'interfaccia del meccanico è stata progettata per l'uso in mobilità. La dashboard mostra gli ordini suddivisi per stato: in arrivo, accettati, in lavorazione e completati. Ogni ordine presenta le informazioni essenziali: cliente, veicolo, servizio richiesto, ubicazione, urgenza. L'accettazione o il rifiuto avvengono con un singolo tap. In caso di accettazione, l'ordine entra nel calendario del meccanico e l'utente riceve conferma immediata.

Quando il preventivo viene generato, l'utente riceve una notifica push che lo informa della disponibilità della stima. Aprendo l'applicazione, viene automaticamente visualizzata una bottom sheet che presenta il preventivo in maniera sintetica e comprensibile, includendo il servizio richiesto, le singole voci di costo e il prezzo totale. L'interfaccia mette a disposizione due azioni principali: accettare il preventivo e procedere con la prenotazione oppure rifiutarlo. Nel caso di accettazione, l'utente viene immediatamente indirizzato alla selezione della data dell'intervento, tramite un calendario che mostra esclusivamente gli slot effettivamente disponibili. L'intero flusso, dalla visualizzazione del preventivo alla conferma della prenotazione, è progettato per essere completato in pochi passaggi, riducendo al minimo il tempo e lo sforzo richiesto all'utente.

Il pagamento è attualmente gestito in loco con POS mobile, una scelta pragmatica per la fase iniziale. Il meccanico processa il pagamento al termine dell'intervento, quando il cliente ha verificato il lavoro svolto. Questo approccio rassicura chi è diffidente verso i pagamenti anticipati online, specialmente per importi significativi.

La fatturazione segue il flusso operativo reale. Completato l'intervento e ricevuto il pagamento, il meccanico invia la fattura direttamente dall'app. I dati sono precompilati: informazioni cliente dal profilo, dettagli veicolo dal database, descrizione servizio dall'ordine, importo dal preventivo accettato. Il meccanico può aggiungere note su lavori aggiuntivi effettuati, pezzi di ricambio utilizzati e sconti applicati. Il PDF viene salvato su Supabase Storage con naming convention anno/mese/numero-progressivo.pdf per facilitare l'archiviazione.

La roadmap futura prevede l'implementazione dei pagamenti in-app per completare il ciclo digitale. L'utente potrà pagare al momento della prenotazione o al termine del servizio direttamente dall'applicazione, eliminando la frizione del POS fisico e abilitando funzionalità come split payment e pagamento rateale.

2.5.4 Notifiche push e mail

Il sistema di notifiche è fondamentale per mantenere gli utenti informati e coinvolti. Ti-Riparo utilizza un approccio multicanale: notifiche push per le comunicazioni urgenti, email per documenti e conferme.

Le notifiche push sono gestite attraverso Expo Push Notification Service. Al momento dell'installazione, l'app richiede il permesso per le notifiche e registra il token del dispositivo. Questo viene salvato nel profilo utente e utilizzato per invii mirati.

Le notifiche seguono una gerarchia di priorità. Alta priorità: prenotazione confermata, meccanico in arrivo, pagamento ricevuto. Media priorità: nuovo preventivo disponibile, promemoria appuntamento (24 ore prima). Bassa priorità: promozioni e aggiornamenti dell'app.

L'implementazione delle email è gestita da Resend. I template sono realizzati con React Email, permettendo di creare email HTML responsive con la stessa tecnologia dell'app.

2.5.5 UI/UX design

Il design di Ti-Riparo ha attraversato due fasi distinte: una versione iniziale minimale per validare il concept, e un redesign completo che ha portato l'app alla sua forma attuale.

La prima versione era volutamente minimale. L'obiettivo era validare il concept rapidamente, limitando l'implementazione al minimo indispensabile: quattro schermate in totale con selezione servizio, inserimento posizione, scelta della data e riepilogo finale. Nessuna animazione, nessuna transizio-

ne elaborata, solo funzionalità. Il risultato, pur essendo privo di rifinitura estetica, era funzionale e ha permesso di raccogliere i primi feedback.

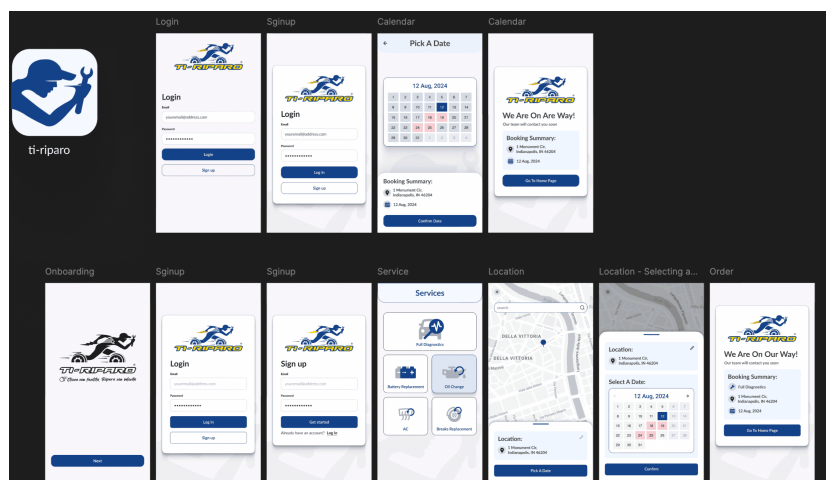


Figura 2.7: Mockup della prima versione dell'interfaccia su Figma

La svolta è arrivata con l'implementazione della bottom sheet dinamica, probabilmente la decisione di design più significativa del progetto. Invece di navigare tra schermate separate, l'intera esperienza avviene in un unico ambiente fluido. L'utente apre l'app e visualizza immediatamente la mappa della propria zona con la bottom sheet che sale dal basso. Non si tratta di un elemento statico: si adatta al contenuto, si espande e si contrae seguendo il flusso delle azioni.

Il primo impatto è immediato: la mappa mostra la posizione dell'utente, la bottom sheet presenta i servizi disponibili. Selezionando un servizio, la sheet si abbassa rivelando la sezione veicoli. Se non è stato ancora aggiunto un veicolo, appare un invito ad aggiungerlo. L'aggiunta del veicolo tramite ricerca targa — inserendo la targa, l'app interroga il database e popola automaticamente marca, modello e anno — è una delle funzionalità più apprezzate dagli utenti.

Il menu, accessibile dalla home, è volutamente secondario rispetto al flusso principale. Account, veicoli, ordini passati, preventivi, fatture, contatti

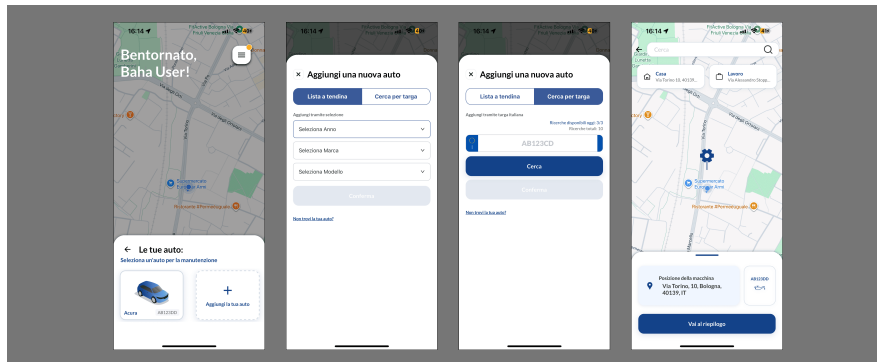


Figura 2.8: Flusso di aggiunta veicolo tramite ricerca targa

e logout sono organizzati in una lista con icone intuitive, senza distrarre dall'esperienza di prenotazione.

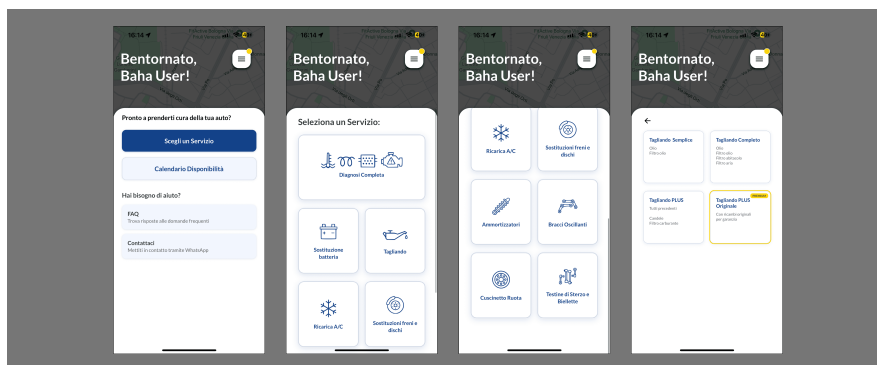


Figura 2.9: Flusso di selezione di un servizio nella versione attuale

Nel flusso principale, dopo la selezione del veicolo la bottom sheet si posiziona a metà altezza. La mappa resta visibile sopra, mentre sotto appare un riepilogo compatto con targa e servizio scelto, accompagnato da una barra di ricerca per l'indirizzo. Se il servizio è a catalogo, la sheet presenta immediatamente il calendario per scegliere data e ora. Se serve un preventivo personalizzato, l'interfaccia si adatta mostrando la conferma di invio della richiesta con una stima dei tempi di risposta.

Le animazioni sono misurate: la bottom sheet che scorre con un leggero rimbalzo, le card che rispondono al tocco. Ogni animazione ha uno scopo preciso: confermare un'azione, guidare l'attenzione, rendere l'attesa meno percepibile. L'ottimizzazione per iOS e Android ha richiesto compromessi continui: la bottom sheet su iOS segue le convenzioni native con angoli arrotondati e barra per il drag, mentre su Android aderisce alle Material Design guidelines.

Capitolo 3

Sviluppo e implementazione

3.1 Architettura del codice ed esempi

3.1.1 Struttura cartelle e organizzazione dei componenti

La Figura 3.1 illustra l'organizzazione modulare del progetto. I componenti UI riutilizzabili risiedono in `components/`, con oltre 15 varianti di `BottomSheet` per le diverse fasi del flusso di prenotazione. Il layer di accesso ai dati in `query/` separa le chiamate a Supabase dalla logica di presentazione. La navigazione in `navigation/` orchestra tre navigator (autenticazione, app principale e deep linking) attraverso un `RootNavigation` centrale.

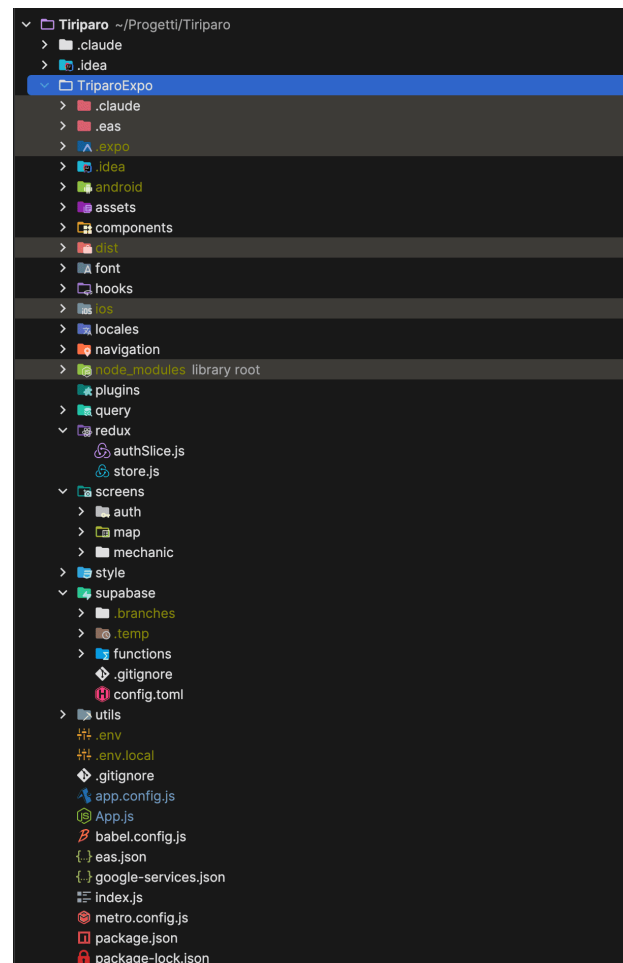


Figura 3.1: Organizzazione delle cartelle del progetto Ti-Riparo

3.1.2 Componenti chiave

Il componente più articolato è il sistema di BottomSheet dinamici: non un singolo elemento, ma una famiglia di sheet che si scambiano dati e si coordinano tra loro. Il codice seguente mostra la struttura principale del BottomSheet nella schermata home:

```
1 <BottomSheet
2   ref={sheetRef}
3   index={2}
4   snapPoints={homeSnapPoints}
5   enablePanDownToClose={false}
6   handleComponent={() => <CustomHandle />}
7   backgroundStyle={{borderRadius: 32}}
8   onChange={handleSheetChanges}
9   backdropComponent={renderBackdrop}
10 >
11   {!isSelectedCar ? (
12     <BottomSheetSelectACar
13       onSelectACar={handleCarSelect}
14       onBackToServices={handleBackToServices}
15     />
16   ) : (
17     <BottomSheetContent
18       address={address}
19       onPickDate={showDatePickerModal}
20       selectedCar={selectedCar}
21       selectedService={selectedService}
22       onConfirm={handleSummary}
23     />
24   )}
25 </BottomSheet>
```

Listing 3.1: Struttura del BottomSheet nella schermata home

La logica condizionale determina quale contenuto mostrare: se l'utente non ha ancora selezionato un veicolo, appare il componente di selezione; una volta scelto il veicolo, la sheet presenta il contenuto per completare la prenotazione.

La gestione della mappa ha rappresentato una delle sfide tecniche più rilevanti. Il `MapComponent` integra Google Maps con overlay personalizzati per mostrare la posizione selezionata e l'area di copertura del servizio. Oltre alla logica standard (permessi di geolocalizzazione, fallback in assenza di GPS, interazione con il sistema di ricerca indirizzi), è stato necessario gestire differenze significative di comportamento tra Android e iOS. Per ottenere un funzionamento uniforme su entrambe le piattaforme, è stata integrata direttamente la Google Maps SDK, adattando l'implementazione ai rispettivi ambienti.

Il `FloatingMenu`, posizionato sopra la mappa, offre accesso rapido a tutte le funzionalità secondarie: account, veicoli, ordini, preventivi, fatture e contatti. L'animazione di apertura utilizza React Native Reanimated per garantire fluidità anche su dispositivi meno recenti.

3.1.3 Gestione del flusso di prenotazione

Il flusso di prenotazione costituisce il percorso critico dell'applicazione. Parte dalla selezione del servizio, prosegue con la scelta o l'aggiunta del veicolo, l'inserimento della posizione e culmina con la conferma nel `SummaryScreen`.

Ogni passaggio è progettato per ridurre l'attrito. Alla selezione di un servizio, il `BottomSheet` mostra la lista dei veicoli salvati. In assenza di veicoli, appare automaticamente l'opzione per aggiungerne uno. La ricerca targa, integrata con un'API esterna, popola automaticamente marca, modello e anno.

La selezione della posizione combina ricerca testuale e interazione con la mappa. Il `SearchComponent` integra Google Places con TomTom come fallback, garantendo la disponibilità dell'autocompletamento in ogni circo-

stanza. Confermata la posizione, il sistema verifica che rientri nell'area di copertura.

La verifica dell'area di copertura avviene tramite una funzione che confronta il codice postale dell'indirizzo selezionato con la tabella `allowed_postal_codes` nel database. Se il CAP non è presente tra quelli abilitati, l'app mostra una modale informativa che comunica all'utente l'indisponibilità del servizio nella zona selezionata. Questa scelta riflette il vincolo operativo della fase attuale: con un singolo meccanico basato a Bologna, l'area di copertura è limitata ai CAP dell'area metropolitana bolognese. L'architettura è tuttavia predisposta per un'estensione: l'aggiunta di nuove zone operative richiede semplicemente l'inserimento dei relativi codici postali nella tabella, senza modifiche al codice dell'applicazione.

3.2 L'ecosistema web: Dashboard e sito vetrina

3.2.1 Dashboard amministrativa con SvelteKit

Parallelamente all'app mobile, è stata sviluppata una dashboard per la gestione operativa del servizio, accessibile all'indirizzo `dashboard.ti-riparo.it`. La scelta di SvelteKit è stata motivata dalla volontà di esplorare un framework diverso da React e dalle sue performance native superiori.

La dashboard consente al meccanico e agli amministratori di gestire ordini, visualizzare statistiche, controllare i pagamenti e generare report. L'interfaccia è ottimizzata per desktop ma responsive per tablet, considerando che viene spesso consultata in mobilità.

L'integrazione con Supabase è diretta: stesso database e stesse API dell'app mobile, ma con un layer di autorizzazione più restrittivo che limita l'accesso ai soli utenti con ruolo amministrativo.

L'aspetto più interessante dal punto di vista tecnico è il funzionamento in tempo reale. Attraverso le subscription di Supabase, la dashboard si aggiorna

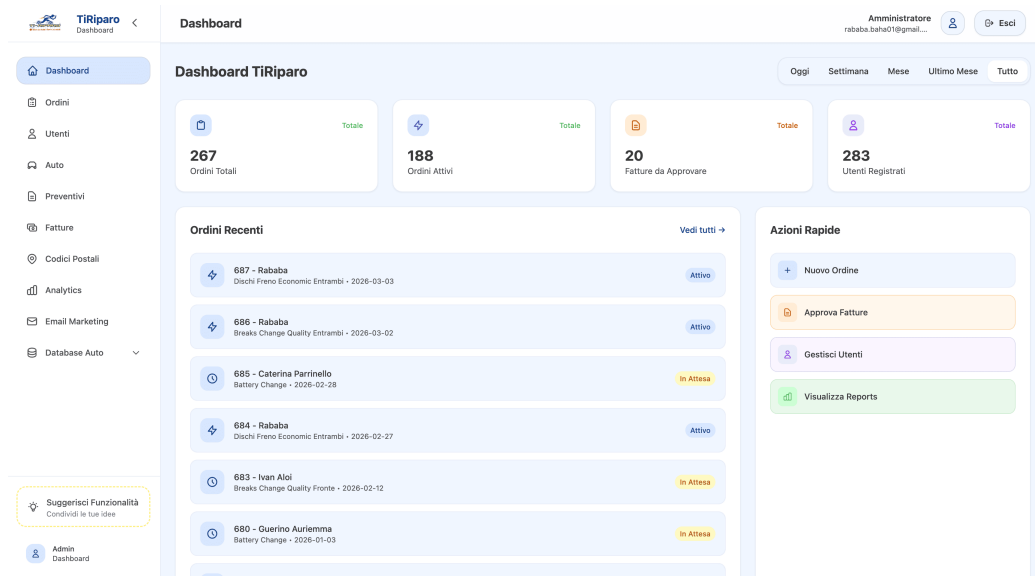


Figura 3.2: Home della dashboard amministrativa per la gestione degli ordini

automaticamente all'arrivo di nuovi ordini, al cambio di stato o all'aggiunta di commenti, senza necessità di ricaricare la pagina.

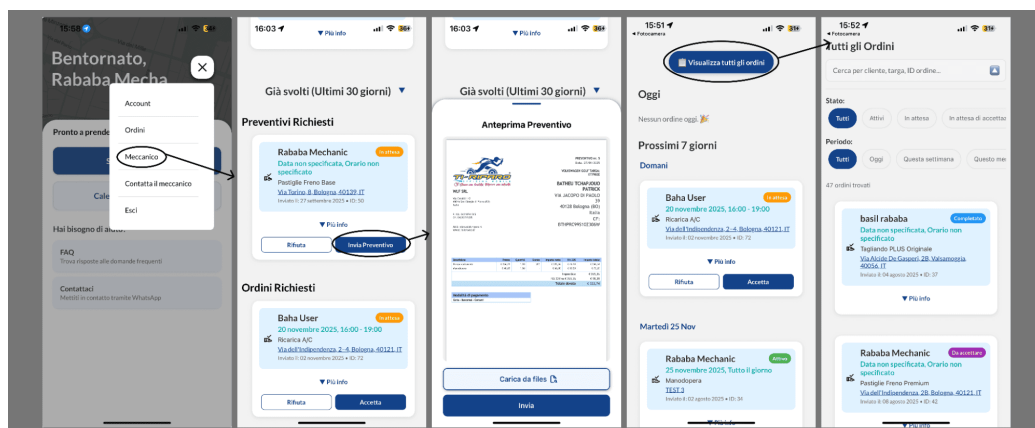


Figura 3.3: Vista dell'interfaccia dedicata al meccanico per la gestione degli ordini

3.2.2 Sito vetrina ti-riparo.it con Next.js

Il sito `ti-riparo.it` rappresenta la presenza pubblica del servizio. Sviluppato con Next.js, presenta l'offerta, mostra i prezzi, permette di scaricare l'app e ospita le landing page per le campagne di marketing.

La scelta di Next.js è stata strategica: il rendering server-side garantisce una corretta indicizzazione da parte di Google, aspetto fondamentale per l'acquisizione organica di utenti. Le API routes gestiscono i form di contatto e l'iscrizione alla newsletter.



Figura 3.4: Home page del sito vetrina ti-riparo.it

3.3 API e Edge Functions

Le Edge Functions di Supabase, scritte in TypeScript ed eseguite su runtime Deno, costituiscono il layer server-side di Ti-Riparo. Gestiscono tutte le operazioni che non possono risiedere nel client per ragioni di sicurezza o di accesso a credenziali esterne.

3.3.1 Edge Function per la ricerca veicolo tramite targa

Un esempio significativo di Edge Function è quella che gestisce la ricerca veicolo tramite targa. Questa funzione riceve la targa dal client, la valida, interroga il servizio esterno OpenAPI Automotive e restituisce i dati strutturati del veicolo. Il Listato 3.2 mostra l'implementazione completa.

```

1 Deno.serve(async (req) => {
2   try {
3     if (req.method !== "POST") {
4       return new Response("Method not allowed", { status: 405 });
5     }
6
7     const secretKey = req.headers.get("x-edge-function-key");
8     const validKey = Deno.env.get("EDGE_FUNCTION_SECRET_KEY");
9
10    if (!secretKey || secretKey !== validKey) {
11      return new Response("Unauthorized", { status: 401 });
12    }
13
14    let requestBody;
15    try {
16      requestBody = await req.json();
17    } catch (err) {
18      return new Response("Invalid JSON in request body",
19        { status: 400 });
20    }
21
22    const { licensePlate } = requestBody;
23    if (!licensePlate) {
24      return new Response("Missing license plate",
25        { status: 400 });
26    }
27
28    const isTest = Deno.env.get("USE_TEST_API") === "true";
29    const token = isTest
30      ? Deno.env.get("SANDBOX_AUTOMOTIVE_TOKEN")
31      : Deno.env.get("AUTOMOTIVE_TOKEN");
32
33    if (!token) {
34      return new Response("Server configuration error",
35        { status: 500 });
36    }
37
38    const hostname = isTest
39      ? "test.automotive.openapi.com"
40      : "automotive.openapi.com";
41    const url =
42      `https://${hostname}/IT-car/${licensePlate}`;
43
44    let response;
45    try {
46      response = await fetch(url, {
47        method: "GET",
48        headers: {
49          "accept": "application/json",
50          "Authorization": `Bearer ${token}`
51        },
52      });
53    } catch (err) {
54      return new Response(
55        `Failed to connect to vehicle data API: ${err}`,
56        { status: 502 });
57    }
58
59    if (!response.ok) {
60      const errorText = await response.text();
61      return new Response(
62        `Failed to fetch vehicle data: ${errorText}`,
63        { status: response.status });
64    }
65
66    const vehicleData = await response.json();
67    return new Response(JSON.stringify(vehicleData), {
68      status: 200,
69      headers: { "Content-Type": "application/json" },
70    });

```

```
71 } catch (error) {  
72   return new Response(  
73     'Internal server error: ${error}', { status: 500 });  
74   }  
75 };
```

Listing 3.2: Edge Function per la ricerca veicolo tramite targa

La funzione implementa diversi pattern di sicurezza: autenticazione tramite chiave segreta nell'header, validazione dell'input, gestione degli errori a più livelli e supporto per un ambiente sandbox di test. La distinzione tra ambiente di test e produzione avviene tramite variabili d'ambiente, permettendo di sviluppare e testare senza rischi né costi aggiuntivi.

3.3.2 Integrazione con servizi esterni

Le Edge Function gestiscono anche l'integrazione con Stripe (creazione del Payment Intent e processamento dei webhook) e con Resend per l'invio di email transazionali. In entrambi i casi il client non interagisce direttamente con i servizi esterni: ogni comunicazione è mediata dal server, che valida le richieste e mantiene aggiornato lo stato nel database.

3.4 Deploy, sicurezza e manutenzione

3.4.1 Pipeline di deploy con EAS Build

EAS (Expo Application Services) ha semplificato radicalmente il processo di deploy. La compilazione locale, che richiedeva ore e spesso falliva per errori di configurazione, è stata sostituita da build cloud automatiche attivate dal push su GitHub.

Sono stati configurati tre profili di build: **development** per i test locali, **preview** per il beta testing e **production** per la pubblicazione sugli store. Ogni profilo dispone di variabili d'ambiente, certificati e configurazioni specifiche. Il seguente estratto del file di configurazione illustra la struttura:

```
1 "build": {  
2   "development": {
```

```
3     "developmentClient": true ,
4     "distribution": "internal",
5     "channel": "development",
6     "env": { "APP_VARIANT": "development" }
7 },
8 "preview": {
9     "distribution": "internal",
10    "channel": "preview",
11    "env": { "APP_VARIANT": "preview" }
12 },
13 "production": {
14     "autoIncrement": true ,
15     "channel": "production"
16 }
17 }
```

Listing 3.3: Configurazione dei profili di build EAS

3.4.2 Over-the-air updates

Gli aggiornamenti OTA hanno permesso di iterare rapidamente senza attendere i tempi di revisione degli store. L'utilizzo in produzione ha tuttavia evidenziato la necessità di adottarli con criterio: aggiornamenti troppo pesanti rallentano l'avvio dell'applicazione.

La strategia adottata riserva gli OTA a bug fix critici, aggiustamenti dell'interfaccia, aggiornamenti testuali e piccole funzionalità che non coinvolgono codice nativo. Per modifiche strutturali o l'aggiunta di nuove dipendenze native, si procede con build complete e submission agli store.

3.4.3 Row Level Security e politiche di accesso

La sicurezza dei dati in Supabase si fonda su Row Level Security (RLS) di PostgreSQL. Ogni tabella è protetta da policy che definiscono con precisione

chi può leggere, scrivere, modificare o eliminare ciascuna riga.

Un utente accede esclusivamente ai propri ordini, veicoli e fatture. Un meccanico visualizza gli ordini assegnati a sé e quelli in stato “pending” nella propria zona operativa. Gli amministratori hanno visibilità completa. Queste regole sono applicate a livello di database: anche in caso di compromissione del client, i dati restano protetti.

Capitolo 4

Management e strategia

4.1 Modello di business

Ti-Riparo nasce come iniziativa imprenditoriale volta a introdurre nel mercato locale un servizio di officina mobile specializzato in interventi meccanici a domicilio. L'azienda è costituita da tre soci con ruoli complementari: lo sviluppo dell'intera piattaforma tecnologica (app mobile, dashboard e sito web) è stato curato dall'autore di questa tesi, mentre i soci si occupano della parte operativa e commerciale.

Il progetto ha richiesto un investimento iniziale significativo, principalmente per l'acquisto e l'allestimento di un veicolo attrezzato come officina mobile. Questa infrastruttura costituisce il cuore dell'offerta: un servizio comodo, flessibile e orientato al cliente, proposto a un prezzo leggermente superiore rispetto alle officine tradizionali, coerente con il valore aggiunto della prestazione a domicilio.

Il modello di ricavo è diretto: i servizi meccanici a domicilio vengono offerti con un sovrapprezzo del 15-20% rispetto alle tariffe di officina tradizionale, giustificato dalla comodità e dal risparmio di tempo per il cliente.

Sul piano operativo, tuttavia, la struttura dei costi presenta criticità rilevanti. La manodopera specializzata, i servizi amministrativi e le spese ricorrenti di gestione del veicolo incidono sensibilmente sui margini. Un elemento

strategico positivo è rappresentato dall'infrastruttura software, sviluppata e mantenuta senza costi ricorrenti grazie all'uso di strumenti open source e piani gratuiti di piattaforme cloud.

La difficoltà principale riguarda la conversione degli utenti registrati in ordini effettivi. Nonostante un buon numero di iscrizioni all'applicazione (294 utenti registrati a marzo 2026), la percentuale di utenti che procede con un servizio resta bassa, segnalando criticità nel funnel di vendita o nella percezione del valore del servizio.

4.2 Metriche di utilizzo

L'analisi delle metriche evidenzia un disallineamento tra interesse iniziale e adozione reale del servizio. I 294 utenti registrati rappresentano un dato significativo per una startup di recente avvio nell'area bolognese, ma il tasso di conversione rimane inferiore al 10%.

Le richieste di preventivo non si trasformano in ordini in modo proporzionale: molti utenti abbandonano il processo prima della conferma, suggerendo problemi di percezione del prezzo, aspettativa del servizio o comunicazione del valore.

Dal punto di vista dei ricavi, la maggior parte del fatturato proviene da clienti B2B consolidati, acquisiti tramite attività precedenti dei soci. La Tabella 4.1 illustra la distribuzione dei ricavi tra i due canali.

	Canale B2B	Canale B2C (app)
Quota ricavi stimata	~75%	~25%
Ricavo medio per ordine	Variabile (contratti)	~169€
Canale di acquisizione	Rete personale soci	App + passaparola
Margine per ordine	Più stabile	Eroso dal CAC
Scalabilità	Limitata	Potenzialmente alta

Tabella 4.1: Distribuzione dei ricavi tra canale B2B e canale B2C

La quota di ricavi generata da clienti privati tramite app è ancora minoritaria, indicando che l’infrastruttura digitale, pur essendo funzionale, non rappresenta ancora il driver principale del business.

4.3 Analisi dei costi e ROI potenziale

Per comprendere le criticità del modello, è utile analizzare un mese rappresentativo. L’ottobre 2024, in cui è stato effettuato un investimento significativo in pubblicità digitale, offre un caso di studio emblematico.

Voce	Importo
<i>Ricavi</i>	
Ricavi da 5 interventi	845€
<i>Costi diretti</i>	
Ricambi	240€
Carburante	~80€
Manodopera (5h × 45€/h)	225€
Margine lordo	300€
<i>Costi di acquisizione</i>	
Pubblicità Facebook	470€
Risultato netto	-170€

Tabella 4.2: Analisi economica del mese di ottobre 2024

Il risultato è una perdita di 170€ ancora prima di considerare i costi fissi mensili: quota commercialista (500€/mese), assicurazioni (circa 170€/mese) e manutenzione del veicolo (importo variabile ma significativo).

Su base annuale, il quadro conferma le criticità. A fronte di 36.642€ fatturati, la struttura dei costi si presenta come segue:

Il margine lordo di circa 9.100€ non tiene conto degli investimenti in pubblicità, della manutenzione straordinaria del veicolo e dell’ammortamento dell’investimento iniziale. Il ROI risulta pertanto negativo.

Voce annuale	Importo stimato
Fatturato totale	36.642€
Costo ricambi (~45%)	~16.500€
Commercialista	6.000€
Carburante	~3.000€
Assicurazioni	~2.000€
Totale costi stimati	~27.500€
Margine lordo annuale	~9.100€

Tabella 4.3: Struttura dei costi e margine lordo su base annuale

Il problema principale risiede nel costo di acquisizione cliente (CAC) tramite pubblicità digitale. Con un CAC di circa 94€ per cliente e un ricavo medio di 169€ per ordine, il margine residuo dopo i costi diretti è insufficiente a coprire i costi fissi. Questo dato evidenzia la necessità di strategie di acquisizione alternative, a costo inferiore, come il passaparola incentivato, l'ottimizzazione SEO e le partnership locali.

4.3.1 Analisi di break-even

Per raggiungere il pareggio operativo è necessario stimare il numero minimo di ordini mensili che coprono i costi fissi. Considerando i costi fissi mensili complessivi (commercialista, assicurazioni, carburante base, ammortamento veicolo) pari a circa 900€/mese, e un margine lordo medio per ordine di circa 60€ (dopo ricambi e manodopera), il punto di break-even si colloca intorno a 15 ordini mensili, escludendo i costi di acquisizione. Includendo un CAC sostenibile (stimato a 20€ per cliente tramite canali organici), il break-even sale a circa 20 ordini mensili. Attualmente la media è di circa 5 ordini B2C al mese, evidenziando la necessità di quadruplicare il volume per raggiungere la sostenibilità.

4.4 Rischi critici

L'analisi dei rischi evidenzia quattro elementi principali. Il burn rate attuale rende l'azienda non sostenibile nel medio periodo. La competizione indiretta è in crescita: le officine tradizionali stanno iniziando a offrire servizi di cortesia come il pick-up gratuito, erodendo il vantaggio competitivo di Ti-Riparo. La stagionalità incide negativamente, con i mesi invernali che registrano meno richieste per interventi non urgenti. Infine, la forte dipendenza da pochi clienti B2B rappresenta un rischio di concentrazione.

4.5 Esito operativo e lezioni apprese

Nel febbraio 2026 la società WLF, che gestiva operativamente Ti-Riparo, è stata posta in liquidazione per esaurimento delle risorse finanziarie. Le criticità evidenziate nelle sezioni precedenti, in particolare il costo di acquisizione cliente insostenibile, il basso tasso di conversione e la dipendenza strutturale dal canale B2B, non sono state risolte nei tempi necessari a garantire la continuità operativa. L'applicazione è stata conseguentemente rimossa dagli store.

Questo esito, pur negativo sul piano imprenditoriale, offre indicazioni concrete. Il modello di officina mobile a domicilio genera interesse reale (294 utenti registrati, NPS di +64) ma richiede una struttura dei costi radicalmente diversa per risultare sostenibile. Il sovrapprezzo del 15-20% non compensa i costi operativi quando il volume di ordini resta sotto la soglia di break-even, e la pubblicità digitale si è rivelata un canale di acquisizione troppo oneroso per i margini disponibili.

Le opzioni strategiche che restano percorribili per una eventuale seconda iterazione del progetto includono il pivot verso un modello software-only (vendita della piattaforma ad officine esistenti come SaaS), la focalizzazione esclusiva sul segmento B2B e la partnership con officine tradizionali di cui Ti-Riparo diventerebbe il braccio digitale. L'esperienza accumulata, il codice e l'architettura non vanno perduti: rappresentano il punto di partenza

per ripensare il progetto su basi economiche più solide, come discusso nelle Conclusioni.

Capitolo 5

Valutazione dell'esperienza utente

5.1 Metodologia di valutazione

Per misurare l'usabilità e la qualità percepita dell'applicazione Ti-Riparo, è stato somministrato un questionario online tramite Google Forms a un campione di utenti registrati sulla piattaforma. L'obiettivo era raccogliere dati quantitativi e qualitativi sull'esperienza d'uso, individuare eventuali criticità nell'interfaccia e nelle funzionalità, e derivare indicazioni concrete per un eventuale sviluppo futuro.

Il questionario è stato strutturato in quattro sezioni. La prima, dedicata alle informazioni generali, indaga la frequenza di utilizzo dell'app e il canale di acquisizione. La seconda propone sei affermazioni sull'usabilità generale, valutate su scala Likert a 5 punti (1 = fortemente in disaccordo, 5 = fortemente d'accordo), ispirate al modello System Usability Scale (SUS) [10]. Le affermazioni includono sia item positivi sia item negativi per ridurre il bias di acquiescenza. La terza sezione valuta nove funzionalità specifiche su una scala a 5 punti (da "Molto difficile" a "Molto facile"), con l'opzione aggiuntiva "Non l'ho mai usata" per evitare risposte forzate. La quarta sezione raccoglie la soddisfazione complessiva (scala 1-5), la probabilità di raccomandazione

(scala 1-10, in linea con il Net Promoter Score [12]), il confronto con metodi tradizionali e domande aperte su punti di forza, criticità, bug e funzionalità desiderate.

Il questionario ha raccolto 25 risposte nel periodo compreso tra il 3 febbraio e il 10 marzo 2026. Il campione, seppur limitato, rappresenta una varietà di profili d'uso: utenti che hanno solo completato la registrazione, utenti occasionali e utenti più assidui. Questa eterogeneità permette di analizzare l'esperienza a diversi livelli di profondità d'uso.

È importante segnalare i limiti dello studio. La dimensione ridotta del campione non consente generalizzazioni statisticamente robuste, tuttavia i risultati evidenziano tendenze coerenti e indicazioni qualitative significative. La distribuzione del questionario tramite canali diretti potrebbe inoltre introdurre un bias di autoselezione, con una possibile sovrarappresentazione di utenti più coinvolti o soddisfatti.

Nota metodologica sul calcolo del punteggio di usabilità. Il questionario utilizza 6 item anziché i 10 previsti dal SUS standard [10]. Il punteggio composito calcolato nella Sezione 5.3.3 è pertanto un *adattamento* della scala SUS e non un punteggio SUS in senso stretto. I confronti con i benchmark della letteratura (ad esempio la soglia di 80,3 per il grado A secondo Sauro e Lewis [11]) sono da intendersi come indicativi e non come equivalenze dirette.

5.2 Profilo dei rispondenti

5.2.1 Frequenza di utilizzo

La Figura 5.1 mostra la distribuzione della frequenza d'uso. Il dato più rilevante è la composizione del campione: 9 rispondenti (36%) hanno completato solo la registrazione, 10 (40%) hanno utilizzato l'app 1-2 volte, 5 (20%) tra 3 e 5 volte, e un solo rispondente ha dichiarato un utilizzo superiore alle 5 volte.

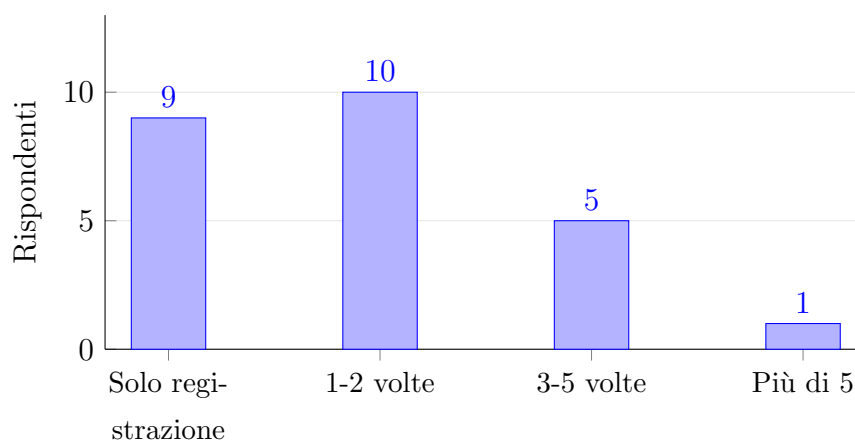


Figura 5.1: Distribuzione della frequenza di utilizzo dell'app tra i rispondenti (n=25)

Questo dato rispecchia la tendenza già osservata nell'analisi delle metriche di business (Capitolo 4): il tasso di conversione tra registrazione e primo utilizzo effettivo rappresentava la criticità principale del funnel. La maggioranza degli utenti si registrava ma non procedeva con un servizio, suggerendo che le barriere all'azione non risiedevano nell'usabilità dell'app in sé, ma nella percezione del servizio, nella fiducia verso un modello nuovo, o nella mancanza di un'esigenza immediata al momento della registrazione.

In un'ottica di ripresa del progetto, l'introduzione di meccanismi di onboarding più incisivi rappresenterebbe una priorità: notifiche di promemoria per chi non ha completato il primo ordine, offerte di benvenuto con sconto sul primo intervento, oppure un sistema di diagnosi gratuita che abbassi la soglia psicologica del primo contatto con il servizio.

5.2.2 Canale di acquisizione

Come illustrato nella Figura 5.2, il passaparola domina nettamente con il 76% dei rispondenti (19 su 25), confermando la centralità delle relazioni personali nell'acquisizione utenti. I social media e la pubblicità online rappresentano ciascuno l'8% del campione.

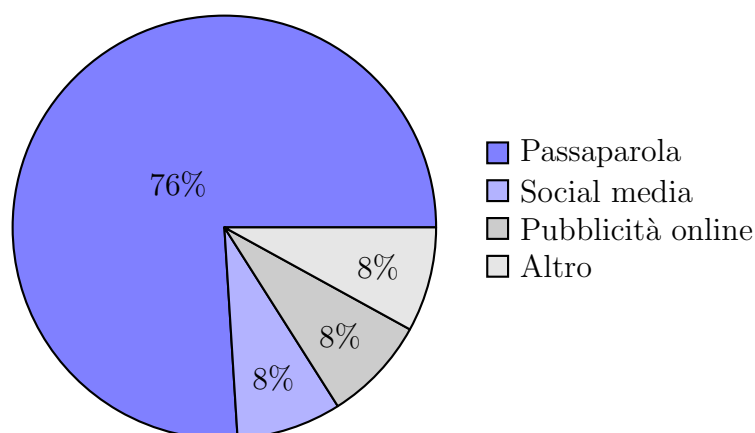


Figura 5.2: Distribuzione dei canali di acquisizione tra i rispondenti (n=25)

La predominanza del passaparola è un dato a doppio taglio. Da un lato, indica che chi utilizza il servizio ne parla positivamente, generando acquisizione organica a costo zero. Dall'altro, rivela una dipendenza da un canale non scalabile: il passaparola cresce linearmente e non è controllabile.

La quasi totale assenza di utenti provenienti da ricerca su Google, App Store o Play Store suggerisce che la visibilità organica dell'app era insufficiente. Per un'eventuale ripresa, una strategia di miglioramento dovrebbe includere l'ottimizzazione ASO (App Store Optimization) con keyword mirate, l'investimento in contenuti SEO e campagne di recensioni sugli store.

5.3 Usabilità generale

La sezione di usabilità generale ha incluso sei affermazioni valutate su scala Likert a 5 punti, da 1 (*fortemente in disaccordo*) a 5 (*fortemente d'accordo*). Alcuni item sono formulati in positivo, mentre altri sono formulati in negativo, in modo da ridurre il rischio di risposte automatiche e ottenere una valutazione più affidabile della percezione d'uso dell'applicazione.

Poiché il campione è contenuto ($n = 25$), si è scelto di rappresentare i risultati mostrando, per ciascuna domanda, la distribuzione completa del-

le risposte invece di un unico grafico riassuntivo. Questa scelta consente di osservare con maggiore chiarezza sia la tendenza centrale sia l'eventuale presenza di risposte polarizzate.

5.3.1 Userei Ti-Riparo frequentemente per la manutenzione della mia auto

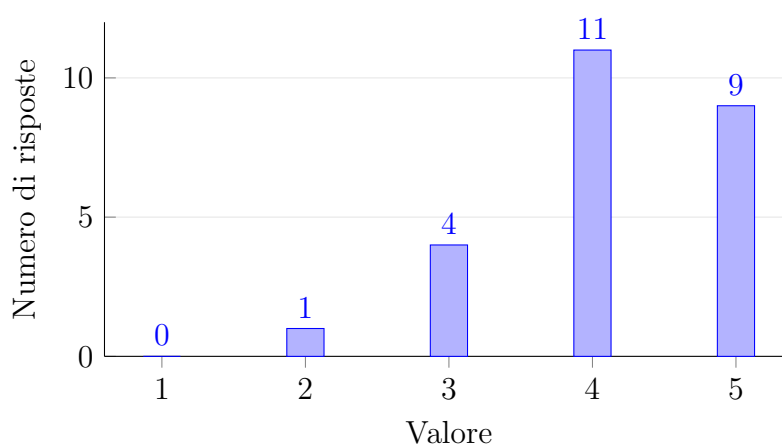


Figura 5.3: Distribuzione delle risposte: Userei Ti-Riparo frequentemente per la manutenzione della mia auto

La distribuzione delle risposte mostra una prevalenza netta di valutazioni alte: 11 utenti hanno selezionato il valore 4 e 9 utenti il valore 5. Solo un rispondente ha scelto 2, mentre nessuno ha selezionato 1. Nel complesso, il risultato evidenzia una forte propensione dichiarata all'utilizzo futuro dell'app per la manutenzione dell'auto.

Questo dato suggerisce un atteggiamento favorevole verso il servizio, anche se va interpretato insieme ai dati d'uso effettivo: una parte degli utenti, pur esprimendo un'intenzione positiva, non ha poi utilizzato frequentemente l'app oltre le fasi iniziali.

5.3.2 Ho trovato l'app semplice da usare

L'item relativo alla semplicità d'uso presenta il risultato più netto dell'intera sezione. La grande maggioranza dei rispondenti ha espresso un giudizio molto positivo: 20 utenti hanno selezionato il valore 5 e 3 il valore 4, mentre solo 2 hanno indicato 3. Nessuno ha assegnato valori pari a 1 o 2.

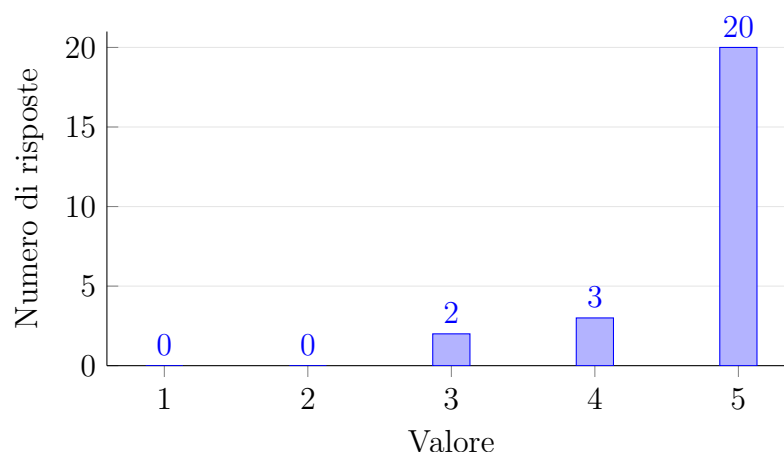


Figura 5.4: Distribuzione delle risposte: Ho trovato l'app semplice da usare

La distribuzione conferma quindi che l'interfaccia è stata percepita come immediata e comprensibile. Questo risultato rappresenta una validazione importante delle scelte progettuali adottate, in particolare per quanto riguarda la chiarezza del flusso di prenotazione e la riduzione degli elementi superflui.

5.3.3 Le funzioni dell'app sono ben organizzate e facili da trovare

La distribuzione delle risposte coincide esattamente con quella dell'item precedente. Anche in questo caso, 20 utenti hanno selezionato 5, 3 hanno selezionato 4 e 2 hanno indicato 3. Non compaiono valutazioni negative.

Il risultato suggerisce che non solo l'app è stata percepita come semplice da usare, ma anche come ben strutturata dal punto di vista dell'architettura.

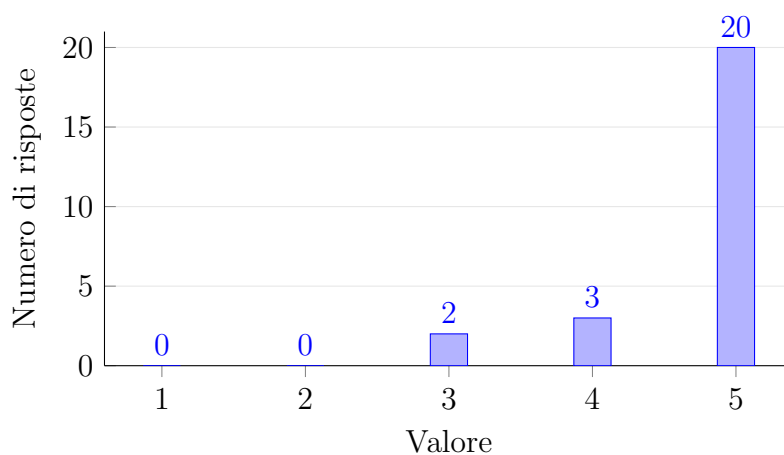


Figura 5.5: Distribuzione delle risposte: Le funzioni dell'app sono ben organizzate e facili da trovare

ra dell'informazione. Le principali funzionalità risultano quindi facilmente individuabili, senza generare confusione o disorientamento nei partecipanti.

5.3.4 Ho avuto bisogno di aiuto per capire come usare l'app

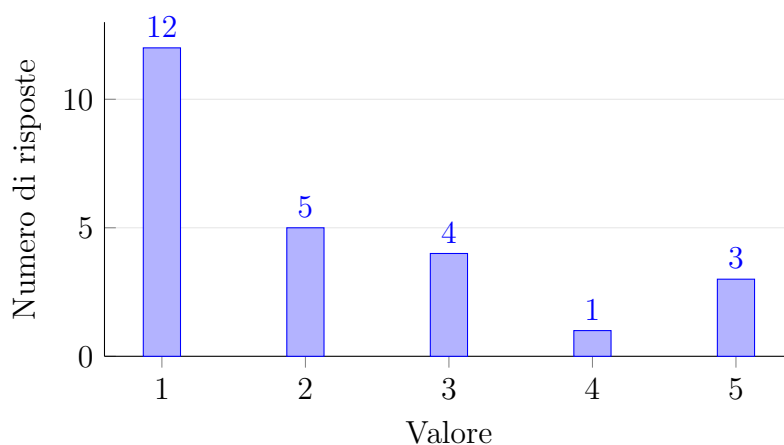


Figura 5.6: Distribuzione delle risposte: Ho avuto bisogno di aiuto per capire come usare l'app

In questo caso l'affermazione è formulata in negativo, quindi un punteggio basso corrisponde a un esito migliore. La maggior parte dei rispondenti ha infatti selezionato valori bassi: 12 utenti hanno scelto 1 e 5 hanno scelto 2. Sono però presenti anche alcune risposte più alte, con 4 utenti che hanno indicato 3, 1 utente che ha indicato 4 e 3 utenti che hanno selezionato 5.

La distribuzione risulta quindi meno compatta rispetto agli item precedenti. Nel complesso emerge che molti utenti non hanno avuto bisogno di supporto, ma una minoranza ha incontrato maggiori difficoltà iniziali. Questo rende visibile una certa eterogeneità nell'esperienza d'uso, probabilmente legata al diverso livello di familiarità con applicazioni digitali di questo tipo.

5.3.5 Mi sono sentito sicuro nell'utilizzare le varie funzioni

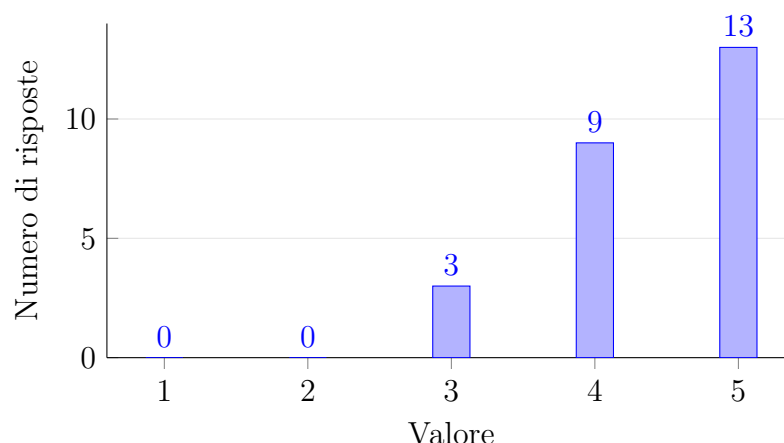


Figura 5.7: Distribuzione delle risposte: Mi sono sentito sicuro nell'utilizzare le varie funzioni

Anche l'item relativo alla percezione di sicurezza restituisce un quadro molto positivo. Nessun utente ha selezionato 1 o 2; 3 utenti hanno indicato 3, 9 hanno selezionato 4 e 13 hanno espresso il massimo punteggio, pari a 5.

Questo dato è particolarmente rilevante perché la percezione di sicurezza rappresenta una componente centrale dell'esperienza d'uso, soprattutto in

un'applicazione che gestisce prenotazioni, richieste di assistenza e interazione con un servizio tecnico. Il risultato suggerisce che l'interfaccia è riuscita a trasmettere affidabilità, chiarezza e controllo.

5.3.6 Ho dovuto imparare molte cose prima di riuscire ad usare l'app

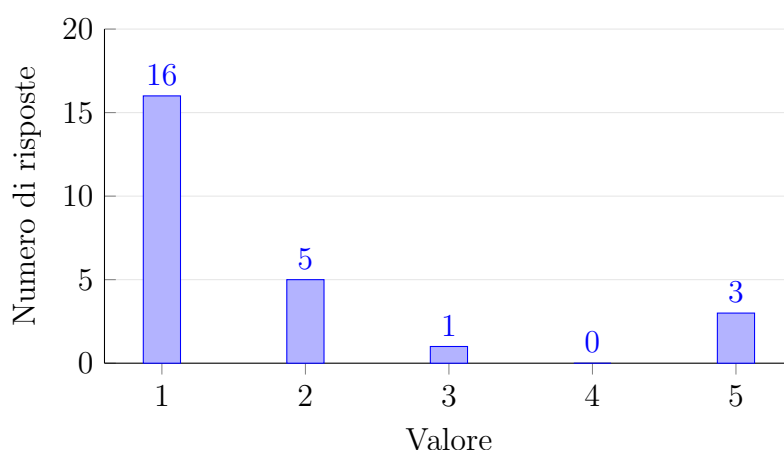


Figura 5.8: Distribuzione delle risposte: Ho dovuto imparare molte cose prima di riuscire ad usare l'app

Anche questo item è formulato in negativo, per cui valori bassi indicano un risultato migliore. La distribuzione delle risposte mostra che la grande maggioranza degli utenti non ha percepito una curva di apprendimento significativa: 16 rispondenti hanno selezionato il valore 1 e 5 il valore 2. Solo un utente ha indicato il valore 3 e nessuno il valore 4, mentre 3 rispondenti hanno selezionato il valore 5, segnalando una minoranza che ha percepito una curva di apprendimento più impegnativa.

Nel complesso, il dato suggerisce che la curva di apprendimento dell'app risulta contenuta e che la maggior parte degli utenti è riuscita a orientarsi senza dover acquisire conoscenze preliminari particolarmente complesse.

5.3.7 Sintesi dell'usabilità generale

Nel complesso, i risultati della sezione mostrano una percezione dell'usabilità molto positiva. Gli item formulati in positivo presentano una forte concentrazione di risposte nei valori 4 e 5, in particolare per quanto riguarda la semplicità d'uso, l'organizzazione delle funzioni e la percezione di sicurezza. Gli item formulati in negativo confermano il quadro generale: nella maggior parte dei casi gli utenti non hanno segnalato un forte bisogno di aiuto né una curva di apprendimento elevata, pur con una certa variabilità tra i partecipanti.

La rappresentazione per distribuzione delle risposte consente inoltre di cogliere meglio le differenze tra gli item: accanto a dimensioni quasi unanimemente valutate in modo positivo, emergono infatti alcuni segnali di difficoltà per una minoranza di utenti. Questo suggerisce che l'app risulta generalmente semplice e chiara, ma che esistono ancora margini di miglioramento nell'accompagnamento iniziale e nell'onboarding.

5.4 Valutazione delle singole funzionalità

Il questionario ha chiesto di valutare la facilità d'uso di nove funzionalità specifiche, su una scala da 1 (Molto difficile) a 5 (Molto facile). La Figura 5.9 presenta i risultati, riportando per ciascuna funzionalità il punteggio medio ottenuto.

I risultati evidenziano una valutazione complessivamente molto positiva dell'applicazione. Tutte le funzionalità analizzate ottengono infatti un punteggio medio superiore a 4,4 su 5, con una media generale pari a 4,52/5. Questo dato suggerisce che le principali interazioni offerte dall'app siano state percepite come semplici, chiare e facilmente utilizzabili.

La funzionalità che registra il punteggio più alto è la scelta di data e orario, con una media di 4,65/5, seguita dall'aggiunta del veicolo tramite targa (4,61/5) e dalla visualizzazione delle fatture (4,58/5). Anche la selezione del servizio (4,57/5) e la consultazione dei preventivi (4,52/5) mostrano va-

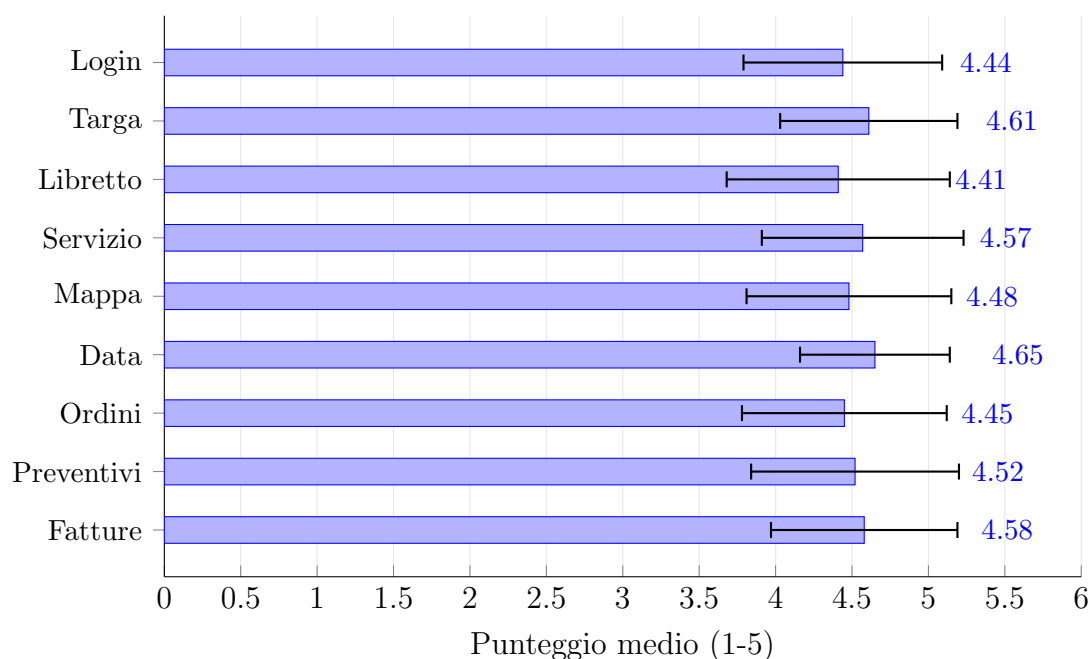


Figura 5.9: Punteggi medi e deviazione standard (σ) per le nove funzionalità valutate. Le barre di errore indicano $\pm 1\sigma$.

lutazioni elevate, confermando la buona qualità percepita delle funzionalità centrali del processo di prenotazione e gestione dell'intervento.

I punteggi leggermente inferiori, pur restando ampiamente positivi, riguardano il caricamento del libretto di circolazione (4,41/5), il login e la registrazione (4,44/5), e la gestione degli ordini (4,45/5). Questi valori non indicano criticità vere e proprie, ma suggeriscono possibili margini di miglioramento in alcune interazioni specifiche, ad esempio nell'acquisizione documentale o nella chiarezza di alcune fasi iniziali del flusso d'uso.

Nel complesso, i dati confermano che tutte le funzionalità considerate risultano facili da usare secondo gli utenti, senza differenze marcate tra aree di base e funzionalità più avanzate. L'esperienza appare quindi coerente e ben progettata lungo l'intero percorso di utilizzo dell'applicazione.

5.5 Soddisfazione complessiva e confronto con metodi tradizionali

5.5.1 Soddisfazione complessiva

La domanda “Quanto sei soddisfatto complessivamente di Ti-Riparo?” su scala 1-5 ha prodotto una media di 4,56/5 ($\sigma=0,71$). La grande maggioranza ha indicato punteggi di 4 o 5. I rispondenti con punteggio 3 sono prevalentemente utenti che hanno completato solo la registrazione: la soddisfazione moderata riflette l'impossibilità di valutare appieno un servizio non ancora sperimentato.

5.5.2 Net Promoter Score (NPS)

La domanda “Quanto è probabile che consiglieresti Ti-Riparo a un amico o collega?” su scala 1-10 ha prodotto una media di 8,92. La Figura 5.10 mostra la distribuzione delle risposte.

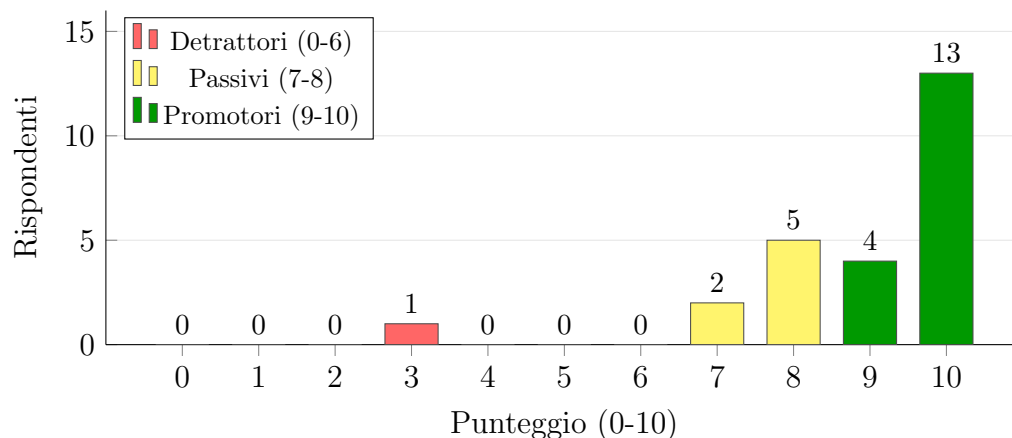


Figura 5.10: Distribuzione delle risposte alla domanda di raccomandazione (n=25)

Classificando le risposte secondo la metodologia NPS [12]:

5.5 Soddisfazione complessiva e confronto con metodi tradizionali 65

Categoria	Percentuale	Rispondenti
Promotori (9-10)	68,0%	17
Passivi (7-8)	28,0%	7
Detrattori (0-6)	4,0%	1

Tabella 5.1: Classificazione delle risposte secondo la metodologia Net Promoter Score

Il risultato è un NPS di +64, un valore considerato eccellente nei benchmark di settore. L'unico detrattore corrisponde a un utente che ha solo effettuato la registrazione e ha conosciuto l'app tramite social media, suggerendo aspettative non allineate con l'offerta.

Il dato è particolarmente significativo se letto insieme alla liquidazione della società (Sezione 4.5): gli utenti raccomanderebbero con convinzione un servizio il cui modello economico non ha retto. Questo conferma che il problema non era il prodotto, ma la sostenibilità del business che lo sosteneva.

5.5.3 Confronto con metodi tradizionali

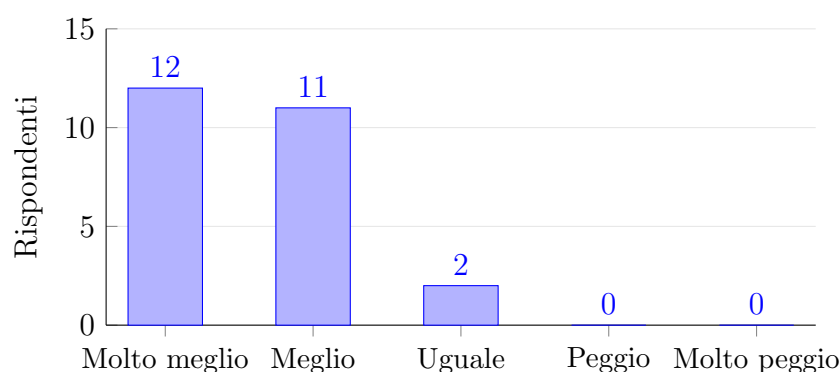


Figura 5.11: Valutazione di Ti-Riparo rispetto ai metodi tradizionali di prenotazione (n=25)

Come mostrato nella Figura 5.11, 12 rispondenti hanno indicato “Molto meglio” e 11 “Meglio”, per un totale del 92%. Due rispondenti hanno indicato “Uguale”, nessuno “Peggio” o “Molto peggio”.

Il dato quasi unanimemente positivo conferma che il valore proposto era chiaro e riconosciuto. Il problema di conversione non era legato alla percezione del servizio ma ad altri fattori: urgenza del bisogno, prezzo e abitudine.

5.6 Analisi qualitativa e suggerimenti degli utenti

Le domande aperte del questionario hanno raccolto risposte particolarmente ricche, offrendo indicazioni concrete sia sui punti di forza sia sulle aree di miglioramento.

5.6.1 Cosa piace di più

Le risposte convergono su tre temi dominanti. La *comodità* è il tema più ricorrente: i rispondenti menzionano il risparmio di tempo e la possibilità di evitare lo spostamento in officina. Un rispondente sintetizza efficacemente: “È un nuovo modo per poter avere assistenza, invece del classico metodo in cui sei costretto tu ad andare dal meccanico”. La *semplicità d'uso* emerge come secondo tema: “la mappa molto semplice da utilizzare per localizzare l'auto” e “è facile da usare, spesso la gente poco pratica ha poca confidenza con i vari meccanici quindi un'applicazione che fa tutto per me è molto comoda”. Infine, la *rapidità*: “il fatto che ottimizza i tempi per il preventivo” e “la rapidità dell'assistenza soprattutto per chi è neopatentato”.

Questo feedback conferma che la value proposition era percepita correttamente dagli utenti.

5.6.2 Cosa migliorare

I suggerimenti degli utenti si articolano su quattro direttrici. La più ricorrente riguarda l'*ampliamento dell'offerta*: più servizi disponibili, una sezione dedicata ai veicoli a due ruote e maggiore capillarità nelle zone coperte, inclusa la richiesta di “una maggiore disponibilità di officine”. La seconda riguarda i *prezzi*: la richiesta di tariffe più competitive evidenzia la tensione strutturale già discussa nel Capitolo 4 tra il sovrapprezzo per il domicilio e la sensibilità al prezzo del mercato target. Un suggerimento particolarmente costruttivo propone un “range di prezzi indicativi per il servizio richiesto basato sugli ordini simili passati”, una funzionalità che avrebbe potuto ridurre l'incertezza senza richiedere un preventivo formale. Un rispondente ha inoltre segnalato un problema di *navigazione*: “alcuni passaggi ti fanno avanzare automaticamente e per modificare qualche dettaglio devi ripartire da capo”, indicando un'area specifica di miglioramento nel flusso di prenotazione.

5.6.3 Bug segnalati

Un solo rispondente ha segnalato problemi tecnici: la necessità di registrarsi nuovamente ad ogni accesso (problema di persistenza della sessione, successivamente risolto) e un episodio isolato di mancato riconoscimento della targa. I restanti rispondenti hanno esplicitamente confermato l'assenza di bug. Il dato indica una stabilità tecnica adeguata per lo stadio di sviluppo.

5.6.4 Funzionalità desiderate

La Figura 5.12 mostra le funzionalità desiderate. La chat con il meccanico emerge come la più richiesta (11 rispondenti, 44%), coerentemente con il contesto analizzato nel Capitolo 1: il rapporto personale con il meccanico è un elemento culturalmente radicato nel mercato italiano. Il pagamento in-app e i promemoria automatici per tagliando e revisione si collocano a pari merito (10 rispondenti ciascuno, 40%), seguiti dal confronto preventivi



Figura 5.12: Funzionalità desiderate dagli utenti, ordinate per numero di richieste (n=25, scelta multipla)

da più meccanici (8, 32%), dal programma fedeltà (7, 28%) e dallo storico manutenzioni (5, 20%).

Queste indicazioni, in un'ottica di ripresa del progetto, delineerebbero una roadmap chiara: la chat, implementabile tramite Supabase Realtime, e i promemoria automatici offrirebbero il miglior rapporto tra impatto percepito e complessità tecnica. Il confronto preventivi sarebbe invece più rilevante in caso di evoluzione verso un modello marketplace.

5.7 Sintesi dei risultati e migliorie derivate

L'analisi del questionario, pur con i limiti di un campione ridotto, restituisce un quadro coerente sintetizzabile in tre evidenze principali.

L'usabilità dell'app non era un problema. Con un punteggio composito di 4,35/5, valutazioni medie superiori a 4,4/5 su tutte le funzionalità testate e un NPS di +64, l'applicazione raggiungeva gli obiettivi di semplicità e accessibilità. La bottom sheet dinamica, il flusso lineare di prenotazione

e l'integrazione automatica della targa sono stati riconosciuti come efficaci dagli utenti.

Il problema principale era l'attivazione. La discrepanza tra il 36% di utenti che si fermava alla registrazione e il 92% che giudicava Ti-Riparo migliore dei metodi tradizionali è il dato più significativo. Gli utenti comprendevano il valore, trovavano l'app facile da usare, dichiaravano l'intenzione di utilizzarla, ma non completavano il primo ordine. Questo gap tra intenzione e azione rappresenta il nodo cruciale emerso dall'analisi.

Le priorità di sviluppo sono chiare. La Tabella 5.2 riassume le miglie derivate dall'analisi, ordinate per impatto atteso e complessità tecnica.

Miglioria	Fonte	Impatto	Complessità
Chat con il meccanico	44% richieste	Alto	Media
Promemoria tagliando	40% richieste	Alto	Bassa
Onboarding guidato	Gap attivazione	Alto	Bassa
Pagamento in-app	40% richieste	Medio	Media
Confronto preventivi	32% richieste	Alto	Alta
Programma referral	NPS +64	Medio	Bassa
Ottimizzazione ASO/SEO	Assenza da ricerca	Medio	Bassa

Tabella 5.2: Miglie derivate dall'analisi, ordinate per impatto e complessità

Queste indicazioni confermano che, per un'eventuale ripresa del progetto, il percorso verso la sostenibilità passerebbe dal miglioramento della capacità di attivare e fidelizzare gli utenti già acquisiti, più che dall'aggiunta di nuove funzionalità tecniche.

Conclusioni

Questa tesi ha documentato l'intero ciclo di vita di Ti-Riparo: dall'analisi del contesto di mercato alla progettazione dell'architettura, dallo sviluppo dell'implementazione alla valutazione dell'esperienza utente, fino all'analisi critica del modello di business e al suo esito operativo.

Il primo contributo del lavoro riguarda la dimostrazione di fattibilità tecnica. Un singolo sviluppatore, utilizzando React Native con Expo, Supabase e servizi cloud a costo zero, può realizzare un ecosistema completo composto da app mobile cross-platform, dashboard amministrativa e sito web, con funzionalità di livello professionale: autenticazione multicanale, notifiche push, ricerca veicolo per targa, gestione preventivi e fatturazione digitale. L'infrastruttura software non ha generato costi ricorrenti nella fase iniziale, un risultato che ha implicazioni concrete per la digitalizzazione delle piccole imprese di servizi, dove il costo della tecnologia rappresenta spesso una barriera insormontabile.

Il secondo contributo riguarda la validazione dell'usabilità. Il questionario ispirato al SUS, somministrato a 25 utenti reali, ha prodotto un punteggio composito di 4,35/5 con un NPS di +64. Tutte le funzionalità testate hanno ricevuto valutazioni superiori a 4/5, e il 92% dei rispondenti ha giudicato Ti-Riparo migliore dei metodi tradizionali per prenotare un meccanico. Questi risultati confermano che le scelte architetturali e di design, in particolare la bottom sheet dinamica come paradigma di navigazione e il flusso lineare di prenotazione, rispondono efficacemente alle esigenze degli utenti.

Il terzo contributo, forse il più significativo in prospettiva, è l'analisi one-

sta delle criticità del modello di business. Con 294 utenti registrati ma un tasso di conversione inferiore al 10%, un CAC di 94€ tramite pubblicità digitale e un ROI annuale negativo, i dati mostrano che la qualità del prodotto tecnologico, pur necessaria, non è sufficiente a garantire la sostenibilità economica. Il gap tra intenzione d'uso dichiarata e comportamento effettivo rappresenta il nodo critico: gli utenti comprendono e apprezzano il valore del servizio, ma non completano il primo ordine.

Tra i limiti del lavoro, il più rilevante riguarda la dimensione del campione per la valutazione dell'usabilità. Con 25 rispondenti, i risultati non permettono generalizzazioni statisticamente robuste. Inoltre, il questionario utilizzato è un adattamento del SUS e non la scala originale a 10 item, il che limita la comparabilità diretta con i benchmark della letteratura. Sul piano del modello di business, l'analisi si basa su un solo anno di operatività in un'unica città, rendendo prematura qualsiasi conclusione sulla scalabilità del modello.

In caso di ripresa del progetto, le direzioni di sviluppo derivano direttamente dai dati raccolti. Sul piano del prodotto, la priorità sarebbe l'implementazione della chat con il meccanico (richiesta dal 44% degli utenti), seguita dai promemoria automatici per tagliando e revisione, e dal completamento del pagamento in-app. Sul piano dell'attivazione, un sistema di onboarding guidato e un programma di referral strutturato rappresenterebbero le leve a più alto impatto e bassa complessità. Sul piano strategico, il pivot verso un modello SaaS o la focalizzazione sul segmento B2B, già identificati nella Sezione 4.5, appaiono le vie più percorribili per costruire un modello economicamente sostenibile.

A livello personale, Ti-Riparo ha rappresentato un'esperienza formativa che va ben oltre lo sviluppo software. Ho imparato che costruire un prodotto tecnologico è la parte più controllabile del processo; fare in modo che le persone lo utilizzino e che il modello economico funzioni è la sfida vera. Ho imparato a leggere i dati senza autocompiacimento, a riconoscere che un'app ben fatta non basta se il funnel di conversione non funziona, e che l'onestà

nell'analisi dei propri risultati è la base per qualsiasi miglioramento.

Ti-Riparo, nella sua forma attuale, non è stato un successo commerciale. È però una dimostrazione concreta di come competenze informatiche, visione imprenditoriale e un approccio sistematico alla misurazione possano convergere in un progetto reale, operativo e misurabile. La sfida, da qui in avanti, è ripensare il modello e trasformare un buon prodotto in un business sostenibile.

Bibliografia

- [1] ADIRA (2025). *Relazione sullo stato della filiera aftermarket Italia 2025*. Disponibile su: <https://www.adira.it/>.
- [2] ANFIA (2025). *Quarto anno di crescita per il mercato aftermarket, con un fatturato in rialzo del 2,4% nel 2024*. Comunicato stampa, 27 febbraio 2025. Disponibile su: <https://www.anfia.it/>.
- [3] Osservatorio Autopromotec (2024). *Officine auto in Italia: numeri, distribuzione sul territorio, nuove competenze*. Su dati Osservatorio MPI Confartigianato Lombardia, Unioncamere e Infocamere, II trimestre 2023. Disponibile su: <https://www.autopromotec.com/>.
- [4] Osservatorio MPI Confartigianato Lombardia (2023). *Alcuni numeri chiave sulla filiera auto in Italia nel 2023*. Elaborazione Flash per il 41° Premio Confartigianato Motori. Disponibile su: <https://www.confartigianato.it/>.
- [5] Ipsos; Europ Assistance (2023). *Mobility Barometer 2023*. Europ Assistance Group. Disponibile su: <https://www.europ-assistance.com>.
- [6] Ecoprogram Flotte (2022). *Mobilità e nuovi servizi: il caso di successo di Officlick*. Disponibile su: <https://ecoprogramflotte.com/2022/06/13/mobilita-e-nuovi-servizi-il-caso-di-successo-di-officlick/>.

-
- [7] Motor Finance Online (2024). *From startup to acquisition: Fixter's journey in car maintenance — Q&A with CEO Limvirak Chea*. 22 agosto 2024. Disponibile su: <https://www.motorfinanceonline.com/>.
 - [8] Città Metropolitana di Bologna (2023). *Parco veicolare nella città metropolitana di Bologna, Anno 2023*. I Numeri di Bologna Metropolitana. Disponibile su: <https://inumeridibolognametropolitana.it/>.
 - [9] Virgilio (2025). *Autofficine e centri assistenza a Bologna*. Disponibile su: https://www.virgilio.it/italia/bologna/cat/AUTOFFICINE_E_CENTRI_ASSISTENZA.html.
 - [10] Brooke, J. (1996). SUS — A quick and dirty usability scale. *Usability Evaluation in Industry*, 189(194), 4–7.
 - [11] Sauro, J.; Lewis, J. R. (2016). *Quantifying the User Experience: Practical Statistics for User Research*. 2nd ed. Morgan Kaufmann.
 - [12] Reichheld, F. F. (2003). The one number you need to grow. *Harvard Business Review*, 81(12), 46–54.
 - [13] Meta Platforms (2024). *React Native — Learn once, write anywhere*. Disponibile su: <https://reactnative.dev/>.
 - [14] Expo (2024). *Expo Documentation*. Disponibile su: <https://docs.expo.dev/>.
 - [15] Supabase Inc. (2024). *Supabase Documentation*. Disponibile su: <https://supabase.com/docs>.

Ringraziamenti

Desidero ringraziare innanzitutto il Professor Federico Montori, per la disponibilità e i consigli durante la stesura di questo lavoro.

Un ringraziamento speciale va ai miei soci: Basil, mio fratello, ideatore e creatore di Ti-Riparo, e Nico, il meccanico che ha gestito l'intera parte operativa e burocratica. Senza di loro questo progetto sarebbe rimasto un'idea su carta.

Alla mia famiglia devo tutto. Mamma, Papà, Bashy e Basil: mi avete supportato in ogni fase di questo percorso, anche quando le serate e i weekend sparivano tra codice, studio e qualche svago di troppo. Un grazie particolare ai miei fratelli, il cui aiuto è stato immenso non solo negli studi ma in ogni aspetto della mia vita. Senza di loro non saprei dove sarei oggi.

Grazie a Isa, che in questi tre anni mi è stata accanto con amore e pazienza infinita, dandomi forza nei momenti difficili e non smettendo mai di credere in me.

Un grazie anche a Fiorella e Maria, che ormai sono parte della famiglia e hanno sempre avuto una parola di incoraggiamento per me.

Grazie agli amici universitari che hanno reso questo percorso indimenticabile. A Thomas, Guada e Pietro, con cui ho passato giornate intere a Mineralogia tra sessioni di studio e partite a Cactus. A tutti coloro che hanno condiviso con me esami, ansie e soddisfazioni.

Grazie a Wafo, Gabbo, John, Baka, Jo e Abe, amici di sempre, per le chiacchiere infinite, le serate e i momenti passati insieme che porto con me.

Infine, grazie a tutti gli utenti di Ti-Riparo che hanno scaricato l'app, pro-

vato il servizio e dedicato tempo a compilare il questionario. Ogni feedback ha reso questo progetto un po' migliore.