

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

Corso di Laurea triennale in Informatica per il Management

Tesi di Laurea in Informatica per il Management

**PROGETTAZIONE E SVILUPPO DI
UN'APPLICAZIONE MOBILE PER IL
MONITORAGGIO DI DISPOSITIVI REMOTI
TRAMITE NMS**

Presentata da:

Letizia Bruno

Relatore:

Dott. Luca Sciullo

Correlatore:

Dott. Leonardo Zambonelli

Sessione I

Anno Accademico 2023/2024

Dedicato a ...

*Tutte quelle persone che hanno sempre
creduto in me e nelle mie capacità.*

Grazie

Introduzione

Nell'era sempre più interconnessa delle reti digitali, il monitoraggio e la gestione efficiente dei dispositivi remoti rappresentano una sfida cruciale per garantire la continuità delle operazioni e la sicurezza delle reti stesse. L'avvento delle tecnologie mobili ha ampliato le prospettive, permettendo un accesso immediato e flessibile alle informazioni critiche, ovunque ci si trovi.

In questo contesto, la presente tesi si focalizza sull'analisi e lo sviluppo di un'applicazione mobile denominata "Ranger", progettata per il monitoraggio avanzato dei dispositivi connessi a reti distanti tramite sistemi NMS (Network Monitor System). Ranger offre agli utenti un'interfaccia intuitiva e potente per monitorare in tempo reale lo stato, la salute e le prestazioni dei dispositivi remoti.

Una parte significativa del lavoro svolto riguarda l'esperienza utente (UX) e l'interfaccia utente (UI) di Ranger. Il prototipo iniziale dell'applicazione ha sì dimostrato la sua funzionalità, ma ha altresì evidenziato alcune carenze nell'esperienza d'uso e nella presentazione grafica.

Attraverso un'approfondita indagine condotta mediante questionari e interviste rivolte ai tecnici, utenti chiave dell'applicazione, sono state identificate criticità e suggerimenti per il miglioramento delle UX e delle UI. Questo prezioso feedback ha costituito il fondamento per un processo iterativo di sviluppo, che ha portato alla realizzazione di una versione aggiornata di Ranger caratterizzata da notevoli miglioramenti in termini di usabilità e di impatto visivo.

Il cuore di Ranger è il suo obiettivo principale: fornire agli utenti un controllo totale sui loro dispositivi remoti, indipendentemente dalla loro posizione geografica. Grazie a una combinazione di funzionalità avanzate e un design intuitivo, l'applicazione mira a semplificare il processo di monitoraggio NMS, permettendo agli utenti di individuare e risolvere

tempestivamente potenziali problematiche.

Indirizzata a un pubblico di tecnici altamente specializzati, Ranger è stata progettata per gestire un vasto numero di dispositivi, inclusi oltre 400 antenne radio sparse sull'intero territorio italiano.

Attraverso un'analisi dettagliata delle caratteristiche e delle funzionalità di Ranger, questa tesi si propone di valutare l'impatto dell'applicazione sul miglioramento dell'efficienza e dell'affidabilità delle reti remote, oltre che a esplorare le potenziali prospettive future per lo sviluppo e l'implementazione di soluzioni simili.

Nel corso di questa tesi, verranno esaminate in dettaglio le metodologie utilizzate per il miglioramento delle UX e delle UI di Ranger, i risultati ottenuti e le possibili implicazioni di tali miglioramenti sulle prestazioni complessive dell'applicazione e sull'esperienza degli utenti finali.

L'obiettivo è quello di fornire una panoramica completa e approfondita dell'importanza dell'usabilità e del design nell'ambito del monitoraggio remoto dei dispositivi NMS, con particolare attenzione al caso di studio di Ranger.

Indice

Introduzione	i
1 Stato dell'arte	1
1.1 Introduzione	1
1.2 Evoluzione Storica della Network Management	2
1.2.1 Anni '60-'70: Le Prime Reti e la Nascita delle LAN	2
1.2.2 Anni '80: L'Ascesa delle Reti Aziendali e l'Introduzione di SNMP	2
1.2.3 Anni '90: L'Era di Internet e l'Integrazione dei Sistemi di Gestione	3
1.2.4 Anni 2000: L'Aumento delle Minacce Informatiche e la Gestione Avanzata della Sicurezza	3
1.2.5 Anni 2010 e Oltre: L'Avvento del Cloud, dell'IoT e dell'AI	4
1.3 Importanza della Network Management	4
1.4 Ruolo e Funzioni di un NMS	6
1.5 Implementazione e Sfide	7
1.6 Esempi di NMS Popolari	8
1.6.1 Cisco Prime Infrastructure	8
1.6.2 SolarWinds Network Performance Monitor (NPM)	9
1.6.3 Paessler PRTG Network Monitor	9
1.6.4 Nagios XI	9
1.7 Conclusione	10
2 Architetture utilizzate	13
2.1 React Native e Expo Go	13
2.1.1 React Native	14

2.1.2	Expo Go	15
2.1.3	Differenze tra React Native e lo Sviluppo Nativo	16
2.1.4	Il Model View Controller (MVC)	17
2.1.5	Il Model View ViewModel (MVVM)	18
2.1.6	Approccio di React Native	19
2.1.7	Il ruolo del JS Bridge e l'integrazione di Expo Go	21
3	Implementazione dell'applicazione	25
3.1	Struttura Generale	25
3.1.1	Gestione dello Stato con React Hooks	25
3.2	Sviluppo Front-End	26
3.2.1	Richiesta di Permessi GPS	26
3.2.2	Gestione delle richieste dei Permessi e delle Posizione	28
3.2.3	Schermata di LogIn	29
3.2.4	Autenticazione Utente	30
3.2.5	Gestione del Login	31
3.3	Barra di Navigazione Principale	32
3.4	Funzionalità Principali	33
3.4.1	Schermata "Account"	33
3.4.2	Schermata "MapPage"	34
3.4.3	Schermata "MapStatus"	40
3.5	Classe "APIBackend"	42
4	Valutazione dell'Applicazione	45
4.1	Introduzione alla Valutazione	45
4.2	Procedura di Valutazione	46
4.3	Struttura del Questionario	47
4.4	Risultati della Valutazione	50
4.4.1	Punti di Forza dell'Applicazione	50
4.4.2	Miglioramento sulla Mappa	50
4.4.3	Miglioramento della Map Status	51
4.4.4	Problematica dell'Area Specifica	51

4.4.5	Aggiornamenti e Miglioramenti dell'Applicazione	52
4.5	Miglioramenti	52
4.6	Rivalutazione e conclusioni	55
Conclusioni		61
4.7	Prospettive di Miglioramento	63
4.8	Conclusioni finali	65
Bibliografia		67

Elenco delle figure

2.1	Architetture dell'applicazione	22
3.1	Confronto tra richiesta di posizione su iOS e Android	27
3.2	Schermata di LogIn	30
3.3	Confronto tra area utenti su iOS e Android	33
3.4	Confronto tra visualizzazione MapPage su iOS e Android	35
3.5	Confronto tra visualizzazione MapStatus	41
4.1	Grafico riassuntivo del primo sondaggio	51
4.2	Grafico delle modifiche richieste dai tecnici	52
4.3	Risultati del secondo sondaggio	58

Capitolo 1

Stato dell'arte

1.1 Introduzione

In questo capitolo esploreremo il campo avanzato dei sistemi di gestione delle reti (Network Management, NM), concentrando l'attenzione sul ruolo fondamentale svolto dal Network Management System (NMS)[1, 2, 3, 4]. Questi strumenti rivestono un'importanza cruciale nel monitorare, gestire e ottimizzare le infrastrutture di telecomunicazioni e informatiche, assicurando un funzionamento efficiente e sicuro.

La Network Management è essenziale per ogni organizzazione che dipende da reti complesse per svolgere le proprie attività quotidiane. Attraverso pratiche e strumenti avanzati, la gestione delle reti non solo garantisce prestazioni ottimali, ma anche una sicurezza robusta contro le minacce emergenti nel panorama digitale[5, 6, 7, 8].

L'obiettivo principale di questo capitolo è offrire una panoramica dettagliata delle migliori pratiche nel settore della Network Management e dell'NMS. Esamineremo come queste tecnologie possano migliorare significativamente le operazioni aziendali, aumentando l'efficienza operativa e riducendo i rischi di downtime e di compromissione della sicurezza[9]. Inoltre, metteremo in luce come l'implementazione di soluzioni avanzate non sia solo un requisito tecnologico, ma rappresenti anche un investimento strategico per migliorare la competitività e garantire la resilienza nel contesto aziendale globale.

Durante il nostro percorso, esploreremo le innovazioni emergenti nel settore della Network Management, come l'integrazione di intelligenza artificiale e di analisi predittiva per

anticipare i problemi di rete e migliorare la pianificazione delle risorse. Affronteremo anche le sfide attuali nel campo, come l'interoperabilità tra dispositivi di diversi fornitori e la necessità di conformarsi a regolamenti sempre più stringenti in materia di sicurezza e privacy[1, 3, 4].

1.2 Evoluzione Storica della Network Management

La gestione delle reti ha subito un'evoluzione significativa dalla nascita delle prime reti di telecomunicazioni fino alle complesse infrastrutture odierne. Questa sezione ripercorre le tappe principali di questa evoluzione, evidenziando i cambiamenti chiave e le innovazioni tecnologiche che hanno modellato il campo della network management.

1.2.1 Anni '60-'70: Le Prime Reti e la Nascita delle LAN

Negli anni '60 e '70, le reti di telecomunicazioni erano relativamente semplici, composte principalmente da linee telefoniche e sistemi di commutazione a circuito. La gestione di queste reti non richiedeva strumenti particolarmente avanzati, in quanto il traffico era principalmente voce e i dati venivano trasmessi attraverso linee dedicate [4].

Con l'avvento dei computer e delle reti locali (LAN) negli anni '70, emerse la necessità di gestire non solo la connettività tra computer, ma anche le risorse condivise come stampanti e file server [4]. Le LAN permisero alle organizzazioni di creare reti interne, migliorando l'efficienza operativa e facilitando la condivisione delle risorse.

1.2.2 Anni '80: L'Ascesa delle Reti Aziendali e l'Introduzione di SNMP

Gli anni '80 videro una crescita significativa delle reti aziendali, con l'introduzione di sistemi di gestione delle reti più sofisticati [4]. La diffusione dei personal computer (PC) e delle LAN portò a una maggiore complessità nella gestione delle reti, poiché le organizzazioni dovevano monitorare e gestire una crescente quantità di dispositivi e traffico di dati [1].

Un passo fondamentale in questo periodo fu l'introduzione del Simple Network Management Protocol (SNMP). SNMP standardizzò il modo in cui i dati di rete potevano essere raccolti e monitorati, permettendo agli amministratori di rete di ottenere una visione centralizzata

delle prestazioni e dello stato dei dispositivi di rete [2]. SNMP facilitò anche l'automazione delle attività di gestione, migliorando l'efficienza operativa [3].

1.2.3 Anni '90: L'Era di Internet e l'Integrazione dei Sistemi di Gestione

Negli anni '90, l'espansione di Internet e delle reti aziendali globali aumentò la complessità della gestione delle reti [3]. Le reti si espansero oltre i confini fisici delle organizzazioni, collegando filiali remote, partner commerciali e clienti in tutto il mondo [1].

Questo periodo vide l'emergere della necessità di sistemi integrati per la gestione delle configurazioni, delle prestazioni e della sicurezza delle reti. Le soluzioni di network management iniziarono a offrire funzionalità avanzate come il monitoraggio delle prestazioni in tempo reale, la gestione centralizzata delle configurazioni e il rilevamento e la risposta agli incidenti di sicurezza [5]. L'integrazione di queste funzionalità in un'unica piattaforma migliorò significativamente la capacità delle organizzazioni di gestire reti sempre più complesse.

1.2.4 Anni 2000: L'Aumento delle Minacce Informatiche e la Gestione Avanzata della Sicurezza

Con l'inizio del nuovo millennio, l'aumento delle minacce informatiche e la crescente complessità delle reti portarono allo sviluppo di strumenti di network management con funzionalità avanzate di sicurezza e monitoraggio in tempo reale [6, 7]. Le reti diventavano sempre più eterogenee, comprendendo una varietà di dispositivi e tecnologie, inclusi dispositivi mobili e wireless [1].

Le soluzioni di network management dovettero adattarsi per affrontare queste nuove sfide, integrando capacità di rilevamento delle intrusioni (IDS/IPS), gestione delle vulnerabilità e controllo degli accessi. Inoltre, l'analisi dei log di rete e il monitoraggio dei comportamenti anomali divennero cruciali per identificare e mitigare le minacce in modo proattivo [8].

1.2.5 Anni 2010 e Oltre: L'Avvento del Cloud, dell'IoT e dell'AI

Gli anni 2010 segnarono l'adozione diffusa di tecnologie rivoluzionarie come il cloud computing, l'Internet of Things (IoT) e l'intelligenza artificiale (AI), che trasformarono ulteriormente la gestione delle reti [1]. Le organizzazioni iniziarono a migrare verso soluzioni cloud, sfruttando la scalabilità e la flessibilità offerte dai servizi cloud per migliorare la gestione delle risorse di rete.

L'IoT introdusse una nuova dimensione di complessità, con miliardi di dispositivi connessi che generavano enormi quantità di dati [9]. La gestione di queste reti richiedeva strumenti avanzati in grado di gestire e analizzare grandi volumi di dati in tempo reale, garantendo al contempo la sicurezza e l'affidabilità delle connessioni [1].

L'intelligenza artificiale e il machine learning iniziarono a essere integrati nei sistemi di network management, consentendo l'automazione delle attività di monitoraggio e gestione. Gli algoritmi di AI potevano analizzare i dati di rete per identificare pattern, prevedere potenziali problemi e ottimizzare le prestazioni in modo proattivo.

In conclusione, l'evoluzione della gestione delle reti riflette il progresso tecnologico e le crescenti esigenze delle organizzazioni nel gestire infrastrutture sempre più complesse e interconnesse. Dai primi giorni delle reti semplici agli avanzati sistemi di network management di oggi, il campo ha fatto passi da gigante, rendendo possibile il funzionamento efficiente e sicuro delle reti moderne. Questo sviluppo è stato guidato dall'introduzione di protocolli standard come SNMP, dall'adozione di tecnologie emergenti come l'intelligenza artificiale e dall'evoluzione delle best practices nel settore. Oggi, la gestione delle reti non solo garantisce prestazioni ottimali, ma anche una sicurezza robusta contro le minacce digitali, rappresentando un pilastro fondamentale per il successo operativo delle organizzazioni in un ambiente sempre più interconnesso e dinamico.

1.3 Importanza della Network Management

La Network Management, o gestione di rete, rappresenta un pilastro fondamentale nell'architettura moderna delle tecnologie dell'informazione e delle comunicazioni (ICT). È l'insieme di pratiche e strumenti progettati per monitorare, gestire e ottimizzare le reti di telecomunicazioni e informatiche, garantendo il loro funzionamento efficiente, sicuro e affidabile

[1, 3]. Questa disciplina è essenziale per qualsiasi organizzazione, indipendentemente dalla sua dimensione, che faccia affidamento su reti complesse per svolgere le proprie operazioni quotidiane.

Le reti, che siano reti locali (LAN) all'interno di un singolo edificio, reti geograficamente distribuite a livello regionale o globale, o reti wireless che connettono dispositivi mobili, costituiscono il tessuto connettivo su cui si basano le moderne attività aziendali [4]. L'efficacia e l'affidabilità di queste reti dipendono direttamente dalla capacità di gestirne in modo proattivo e competente lo stato operativo [9].

Il processo di Network Management si articola attraverso diverse fasi interconnesse:

- **Monitoraggio delle prestazioni:** Questa fase implica il costante monitoraggio delle prestazioni della rete per rilevare eventuali anomalie o inefficienze. Utilizzando strumenti avanzati, i gestori di rete possono monitorare il traffico di rete, l'utilizzo delle risorse, i tempi di risposta e altri parametri critici [6, 7]. Il monitoraggio proattivo consente di identificare e risolvere tempestivamente eventuali problemi prima che possano compromettere le operazioni aziendali [8].
- **Gestione della configurazione:** Gestire accuratamente la configurazione dei dispositivi di rete è cruciale per garantire la coerenza e la sicurezza del sistema [5]. Ciò include l'applicazione di politiche di configurazione standardizzate, l'implementazione di aggiornamenti del firmware e la gestione delle modifiche per adattarsi alle esigenze operative in evoluzione dell'organizzazione [1]. Una buona gestione della configurazione non solo migliora la stabilità della rete, ma facilita anche l'implementazione di nuove tecnologie e l'aderenza alle normative di sicurezza [3].
- **Gestione della sicurezza:** Con la crescente minaccia delle violazioni di sicurezza informatica e la complessità delle reti moderne, la sicurezza della rete è diventata una priorità assoluta [1]. I sistemi di Network Management integrano funzionalità avanzate di sicurezza, come il rilevamento delle intrusioni (IDS), la prevenzione delle intrusioni (IPS) e la gestione delle politiche di sicurezza [9]. Monitorare costantemente le vulnerabilità e rispondere prontamente alle minacce rappresenta un elemento chiave per proteggere i dati aziendali e la reputazione dell'organizzazione [8].

- **Analisi e ottimizzazione:** La raccolta e l'analisi dei dati di rete sono cruciali per comprendere le tendenze di utilizzo, identificare i picchi di traffico e pianificare le capacità future [6, 7]. Attraverso strumenti di analisi avanzati, i gestori di rete possono identificare aree di inefficienza, prevedere le esigenze di scalabilità e ottimizzare le risorse di rete per massimizzare le prestazioni e ridurre i costi operativi [1].

La complessità della Network Management è ulteriormente amplificata dalla necessità di gestire reti eterogenee, che possono comprendere dispositivi e tecnologie di diversi fornitori [3]. L'interoperabilità e l'integrazione dei vari componenti di rete rappresentano una sfida continua, richiedendo l'adozione di standard aperti e l'implementazione di soluzioni di gestione di rete scalabili e flessibili [5].

1.4 Ruolo e Funzioni di un NMS

Oltre alle pratiche di base della Network Management, un elemento fondamentale che riveste un ruolo cruciale è il Network Management System (NMS). Un NMS è un sistema software progettato specificamente per facilitare e migliorare le attività di gestione delle reti, offrendo agli amministratori di rete gli strumenti necessari per monitorare, controllare e ottimizzare le infrastrutture di rete in modo centralizzato ed efficace [1, 3].

- **Monitoraggio proattivo delle prestazioni:** Uno dei pilastri principali di un NMS è il suo ruolo nel monitoraggio proattivo delle prestazioni di rete. Utilizzando protocolli standard come SNMP (Simple Network Management Protocol) [2], un NMS è in grado di raccogliere dati dettagliati sullo stato e sull'utilizzo dei dispositivi di rete, consentendo agli amministratori di identificare tempestivamente eventuali problemi di congestione, perdita di pacchetti o altre anomalie che potrebbero compromettere le prestazioni complessive della rete [6, 7].
- **Gestione della configurazione e automatizzazione:** Un altro aspetto chiave di un NMS è la capacità di gestire la configurazione dei dispositivi di rete in modo centralizzato [5]. Questo include la distribuzione e l'applicazione di aggiornamenti del firmware, la configurazione di nuove funzionalità e la gestione delle modifiche. L'automatizza-

zione di queste operazioni non solo riduce il rischio di errori umani, ma migliora anche la coerenza e la conformità con le politiche aziendali e di sicurezza [1].

- **Sicurezza e conformità:** I moderni NMS integrano funzionalità avanzate di sicurezza per proteggere le reti da minacce esterne e interne [3]. Monitorando costantemente i log degli eventi di rete e utilizzando strumenti di analisi comportamentale, un NMS può rilevare e rispondere prontamente a potenziali intrusioni o attività sospette. Inoltre, supporta l'implementazione e l'applicazione delle politiche di sicurezza aziendali, garantendo la conformità con regolamenti e normative settoriali [8].
- **Reportistica e analisi:** Un aspetto critico di un NMS è la capacità di generare report dettagliati sulle prestazioni di rete, sull'utilizzo delle risorse e altre metriche chiave [6, 7]. Questi report non solo forniscono una panoramica chiara dello stato attuale della rete, ma possono anche essere utilizzati per l'analisi delle tendenze, la pianificazione delle capacità e il supporto alle decisioni strategiche IT [9].

1.5 Implementazione e Sfide

Implementare un Network Management System (NMS) è cruciale per migliorare l'efficienza e la sicurezza delle reti aziendali, ma comporta anche diverse sfide che devono essere affrontate con attenzione e competenza.

Nel contesto dell'implementazione di un NMS, una delle sfide principali è rappresentata dall'eterogeneità degli ambienti di rete. Le organizzazioni spesso gestiscono una vasta gamma di dispositivi e tecnologie provenienti da diversi fornitori. È fondamentale che un NMS sia in grado di integrarsi senza problemi con tutti questi dispositivi e supporti standard aperti come SNMP, NetFlow e altri protocolli di gestione [1, 3]. Garantire l'interoperabilità e la compatibilità è essenziale per assicurare un monitoraggio completo e accurato delle infrastrutture di rete.

Un'altra sfida significativa è la gestione dei dati generati da un NMS. Questi sistemi raccolgono continuamente una vasta quantità di dati critici sullo stato delle reti, sulle prestazioni dei dispositivi e sul traffico di rete. È cruciale implementare soluzioni di archiviazione scalabili e robuste per gestire efficacemente questi dati. Inoltre, la sicurezza dei dati è di primaria

importanza: è necessario proteggere i dati sensibili da accessi non autorizzati e da vulnerabilità informatiche, adottando politiche di sicurezza rigorose e utilizzando tecniche avanzate di crittografia e monitoraggio degli accessi [4].

Oltre alle considerazioni tecnologiche, l'implementazione di un NMS richiede anche un adeguato investimento in formazione e risorse umane. Gli amministratori di rete devono essere adeguatamente preparati per utilizzare pienamente le funzionalità dell'NMS e per rispondere efficacemente alle sfide operative quotidiane. Un programma di formazione continua è essenziale per mantenere aggiornate le competenze e affrontare rapidamente i cambiamenti tecnologici e le nuove minacce alla sicurezza [9].

In conclusione, nonostante le sfide associate all'implementazione di un NMS, i benefici derivanti da un monitoraggio e una gestione avanzata delle reti sono significativi. Un NMS ben implementato non solo migliora le prestazioni e la sicurezza delle reti aziendali, ma supporta anche la crescita organizzativa fornendo dati cruciali per prendere decisioni strategiche informate e garantire la conformità con le normative di settore.

1.6 Esempi di NMS Popolari

Ora ci addentreremo nell'esplorazione dei principali Network Management System (NMS) attualmente disponibili sul mercato. Analizzeremo le caratteristiche distintive di alcuni dei NMS più popolari, esaminando come queste piattaforme supportino le esigenze di modernizzazione e sicurezza delle aziende nel panorama ICT contemporaneo.

1.6.1 Cisco Prime Infrastructure

Cisco Prime Infrastructure rappresenta una soluzione di gestione di reti e dispositivi NMS estremamente robusta e ben consolidata nel settore delle telecomunicazioni. Questo software offre una vasta gamma di funzionalità progettate per ottimizzare l'efficienza operativa e garantire la stabilità delle reti.

Tra le principali caratteristiche di Cisco Prime Infrastructure vi è il monitoraggio delle prestazioni, che consente agli amministratori di rete di tracciare in tempo reale le metriche critiche e di individuare eventuali degradi delle prestazioni o anomalie di traffico. Inoltre, il sistema di gestione delle configurazioni di Cisco Prime facilita la distribuzione e l'aggiorna-

mento uniforme delle impostazioni di rete su un vasto numero di dispositivi, migliorando la coerenza e riducendo il rischio di errori umani.

1.6.2 SolarWinds Network Performance Monitor (NPM)

SolarWinds NPM si distingue nel panorama dei software di gestione delle reti per la sua interfaccia utente intuitiva e le potenti capacità di monitoraggio in tempo reale. Questo strumento è particolarmente apprezzato per la sua capacità di fornire una panoramica dettagliata delle prestazioni della rete, consentendo agli amministratori di individuare e risolvere rapidamente le problematiche di connettività, di carico e di disponibilità.

SolarWinds NPM utilizza sensori avanzati per raccogliere dati da una vasta gamma di dispositivi di rete, offrendo una visibilità senza pari sullo stato operativo complessivo della rete di un'azienda. La capacità di generare report personalizzati e di analizzare le tendenze storiche consente agli utenti di prevedere e mitigare i potenziali problemi prima che abbiano un impatto significativo sulle operazioni aziendali.

1.6.3 Paessler PRTG Network Monitor

Paessler PRTG Network Monitor è un software di monitoraggio della rete noto per la sua versatilità e la sua capacità di adattarsi a una vasta gamma di esigenze di monitoraggio. Questo strumento offre una libreria estesa di sensori che consentono di monitorare aspetti critici delle prestazioni della rete, dai parametri di larghezza di banda ai contatori di utilizzo dei dispositivi.

L'approccio altamente personalizzabile di PRTG permette agli amministratori di configurare facilmente sensori specifici per adattarsi alle esigenze uniche del loro ambiente di rete. La capacità di PRTG di integrarsi con una vasta gamma di dispositivi e tecnologie di rete lo rende una scelta popolare tra le aziende che cercano un monitoraggio dettagliato e affidabile delle loro infrastrutture IT.

1.6.4 Nagios XI

Nagios XI è una piattaforma di monitoraggio IT completa e altamente rispettata nel settore, nota per la sua flessibilità e capacità di monitorare non solo dispositivi di rete, ma anche

risorse del server, applicazioni critiche e servizi aziendali essenziali. Nonostante la sua curva di apprendimento e i requisiti di configurazione più elevati rispetto ad altri software di monitoraggio, Nagios XI offre un controllo senza pari sulle operazioni IT di un'azienda.

Gli amministratori di sistema possono beneficiare della capacità di Nagios XI di rilevare e segnalare immediatamente eventuali anomalie o problemi critici, consentendo interventi tempestivi per minimizzare l'impatto sui servizi aziendali.

1.7 Conclusione

Le tecnologie di Network Management e i Network Management System (NMS) rappresentano elementi fondamentali per le organizzazioni moderne, fornendo strumenti essenziali per il monitoraggio, la gestione e l'ottimizzazione delle reti ICT complesse. Questi sistemi sono cruciali per migliorare l'efficienza operativa e garantire la sicurezza delle infrastrutture digitali, supportando al contempo l'innovazione tecnologica e l'adattabilità alle dinamiche esigenze del mercato globale.

L'evoluzione delle reti digitali ha reso necessaria l'implementazione di soluzioni avanzate di Network Management, in grado di affrontare le sfide sempre più complesse legate alla gestione di reti eterogenee e distribuite su vasta scala geografica. Le tecnologie mobili hanno ulteriormente ampliato le possibilità, consentendo un monitoraggio remoto e immediato dei dispositivi da qualsiasi parte del mondo, migliorando significativamente la reattività e la capacità di risposta alle emergenze.

Nel contesto specifico di questa tesi, l'applicazione Ranger rappresenta un esempio tangibile di come le moderne tecnologie di Network Management possano essere implementate per migliorare il monitoraggio e la gestione dei dispositivi remoti. Ranger non solo mira a semplificare il processo di monitoraggio NMS, ma ha anche posto una forte enfasi sull'esperienza utente e sull'interfaccia intuitiva, aspetti cruciali per garantire l'adozione e l'efficacia dell'applicazione da parte degli utenti tecnici.

In conclusione, l'analisi condotta in questa tesi ha evidenziato l'importanza cruciale delle tecnologie di Network Management e dei NMS per le moderne infrastrutture ICT. L'implementazione di soluzioni avanzate come Ranger non solo ottimizza le operazioni aziendali,

ma prepara anche le organizzazioni ad affrontare le sfide future, contribuendo al successo a lungo termine e alla competitività nel panorama globale delle telecomunicazioni e dell'IT.

Capitolo 2

Architetture utilizzate

Nel contesto dello sviluppo dell'applicazione "Ranger", volta a fornire un monitoraggio avanzato dei dispositivi remoti tramite sistemi NMS (Network Monitor System), è essenziale comprendere le architetture tecnologiche impiegate.

Questo capitolo si propone di esaminare dettagliatamente le tecnologie utilizzate nel processo di sviluppo dell'applicazione, ponendo particolare enfasi su due piattaforme fondamentali: React Native e Expo Go. Attraverso un'analisi approfondita, verranno esplorate le caratteristiche principali di entrambe le tecnologie, inclusi i loro ruoli e le loro funzionalità nel contesto dello sviluppo mobile. Inoltre, saranno esaminate le differenze riscontrate tra l'utilizzo di React Native e lo sviluppo nativo, al fine di evidenziare le peculiarità di ciascun approccio.

Infine, verranno esaminati i vantaggi derivanti dall'adozione di React Native e Expo Go per lo sviluppo di applicazioni mobili, fornendo una panoramica chiara delle motivazioni alla base della scelta di tali framework per la realizzazione dell'applicazione.

2.1 React Native e Expo Go

React Native e Expo Go rappresentano le fondamenta su cui è stato costruito il progetto "Ranger". Questa sezione esplorerà dettagliatamente le caratteristiche distintive di entrambe le piattaforme, sottolineando la loro importanza cruciale nel contesto dello sviluppo dell'applicazione.

Iniziamo analizzando React Native, un framework versatile che ha rivoluzionato l'approccio allo sviluppo di app native per dispositivi mobili [10]. Grazie alla sua natura cross-platform e alla stretta integrazione con la libreria JavaScript React, React Native offre una solida base per la creazione di interfacce utente dinamiche e intuitive.

La sua architettura modulare consente agli sviluppatori di condividere il codice tra le diverse piattaforme, riducendo i tempi di sviluppo e garantendo una coerenza nell'esperienza utente su Android e iOS [16].

Passando a Expo Go, ci troviamo di fronte a uno strumento potente e flessibile che semplifica ulteriormente il processo di sviluppo e testing delle applicazioni mobili [11].

Expo Go fornisce un ambiente di sviluppo completo e una vasta gamma di strumenti e librerie predefinite, consentendo agli sviluppatori di concentrarsi sulla creazione di funzionalità innovative senza dover affrontare la complessità della configurazione e del setup iniziale. Inoltre, Expo Go facilita il testing su dispositivi fisici e virtuali, garantendo una rapida iterazione e un feedback immediato durante lo sviluppo dell'applicazione.

Entrambe le piattaforme offrono un'elevata interoperabilità su Android e iOS, garantendo che l'applicazione "Ranger" possa essere distribuita in modo efficiente su entrambi i sistemi operativi mobili senza compromettere l'esperienza utente. Inoltre, la comunità attiva di sviluppatori e la vasta documentazione disponibile per entrambe le piattaforme forniscono un supporto prezioso durante tutto il ciclo di vita dello sviluppo dell'applicazione [14, 16].

In conclusione, React Native e Expo Go rappresentano non solo gli strumenti tecnologici utilizzati per la realizzazione di "Ranger", ma anche i pilastri su cui si basa il successo e la qualità dell'applicazione. La loro combinazione di potenza, flessibilità e facilità d'uso ha consentito di superare le sfide dello sviluppo mobile e di offrire un'esperienza utente eccezionale su Android e iOS.

2.1.1 React Native

React Native è un framework open-source sviluppato da Facebook che consente agli sviluppatori di creare applicazioni mobili native per diverse piattaforme, inclusi Android e iOS, utilizzando JavaScript e React, una libreria JavaScript per la creazione di interfacce utente (UI) dinamiche [10].

Una delle caratteristiche distintive di React Native è la sua architettura cross-platform, che consente agli sviluppatori di scrivere una singola base di codice e condividerla tra le diverse piattaforme, riducendo così il lavoro di sviluppo duplicato e garantendo una maggiore coerenza nell'esperienza utente su dispositivi Android e iOS [12].

React Native sfrutta il concetto di "native components", cioè componenti di interfaccia utente predefiniti specifici della piattaforma, che vengono resi direttamente come elementi nativi del sistema operativo, garantendo un'esperienza utente fluida e performante [16].

Un aspetto chiave di React Native è il suo approccio dichiarativo alla creazione delle interfacce utente, dove gli sviluppatori definiscono la struttura dell'UI in base allo stato dell'applicazione, e React Native si occupa automaticamente dell'aggiornamento dell'UI quando lo stato dell'applicazione cambia [10].

Inoltre, React Native offre una vasta gamma di librerie e moduli aggiuntivi che consentono agli sviluppatori di integrare funzionalità avanzate nelle loro applicazioni, come la gestione dello stato dell'applicazione, la navigazione tra le schermate, l'accesso ai sensori del dispositivo, l'interazione con i servizi web e molto altro ancora [12].

Grazie alla sua flessibilità, scalabilità e alla grande community di sviluppatori attivi, React Native è diventato uno dei framework più popolari per lo sviluppo di applicazioni mobili, consentendo agli sviluppatori di creare app performanti, reattive e ricche di funzionalità in modo efficiente e veloce [16].

2.1.2 Expo Go

Expo Go è una piattaforma e un'applicazione di sviluppo mobile progettata per semplificare il processo di creazione, test e distribuzione di applicazioni per dispositivi mobili su piattaforme iOS e Android . Questa piattaforma offre agli sviluppatori un ambiente di sviluppo completo, eliminando la necessità di configurare manualmente strumenti e ambienti di sviluppo complessi [11].

Con Expo Go, gli sviluppatori possono scrivere codice utilizzando JavaScript e React Native, sfruttando una vasta gamma di librerie e componenti predefiniti per aggiungere funzionalità avanzate alle loro applicazioni [14].

Una delle caratteristiche chiave di Expo Go è la possibilità di testare le applicazioni su dispositivi fisici e virtuali in modo rapido e semplice . Gli sviluppatori possono utilizzare

il proprio dispositivo mobile per testare le applicazioni in tempo reale tramite Expo Go o utilizzare emulatori e simulatori per testare su una vasta gamma di dispositivi virtuali. Questa agilità nello sviluppo consente agli sviluppatori di iterare rapidamente durante il processo di sviluppo dell'applicazione, testando e validando le modifiche in tempo reale.

Una volta completata l'applicazione, Expo Go offre strumenti per distribuire facilmente l'applicazione su App Store di Apple e Google Play Store. Gli sviluppatori possono generare file binari per la distribuzione utilizzando Expo Go, semplificando il processo di distribuzione dell'applicazione [14].

Inoltre, Expo Go gode di una vasta e attiva community di sviluppatori che fornisce supporto, risorse e soluzioni per le sfide comuni nello sviluppo di applicazioni mobili. La documentazione dettagliata e le risorse educative aiutano gli sviluppatori a familiarizzare rapidamente con la piattaforma e a sfruttarne appieno le potenzialità.

In conclusione, Expo Go è uno strumento completo e versatile per lo sviluppo di applicazioni mobili, che offre agli sviluppatori un ambiente di sviluppo completo, strumenti di test e distribuzione semplificati e una vasta community di supporto [11].

2.1.3 Differenze tra React Native e lo Sviluppo Nativo

La scelta tra React Native e lo sviluppo nativo rappresenta una delle decisioni più cruciali nell'ambito dello sviluppo di applicazioni mobili, poiché influisce direttamente sull'intera architettura e sulla tecnologia utilizzata nel progetto

Questa decisione impatta su numerosi aspetti del processo di sviluppo, tra cui la velocità di sviluppo, la manutenibilità del codice, le prestazioni dell'applicazione e la disponibilità di funzionalità avanzate [13].

Innanzitutto, è importante comprendere che lo sviluppo nativo coinvolge la scrittura di codice specifico per ciascuna piattaforma, utilizzando gli strumenti e i linguaggi di programmazione ufficiali forniti da Android e iOS, come Java o Kotlin per Android e Swift o Objective-C per iOS [12].

Questo approccio offre un alto grado di controllo e personalizzazione sulle funzionalità dell'applicazione, ma richiede un maggior investimento di tempo e risorse per sviluppare e mantenere due codebase separate per Android e iOS [12].

D'altra parte, React Native offre un approccio cross-platform, consentendo agli sviluppatori di utilizzare un unico set di codice scritto in JavaScript per creare applicazioni native per entrambe le piattaforme. Questo framework si basa sul concetto di "write once, run anywhere", dove gran parte della logica di business e dell'interfaccia utente può essere condivisa tra le versioni Android e iOS dell'applicazione [13].

Tuttavia, è importante notare che React Native non elimina completamente la necessità di scrivere codice specifico per ogni piattaforma, poiché alcune funzionalità avanzate possono richiedere ancora l'implementazione di componenti nativi per ottenere le massime prestazioni e l'esperienza utente desiderata

Concentrandosi sui modelli architetturali comuni come MVC e MVVM, è possibile comprendere meglio le differenze tra React Native e lo sviluppo nativo. Mentre il paradigma MVC è ampiamente utilizzato nello sviluppo nativo per separare la logica di business (Model), l'interfaccia utente (View) e il controllo del flusso (Controller), l'approccio di React Native adotta una filosofia simile ma con alcuni adattamenti per soddisfare le esigenze dello sviluppo mobile cross-platform

Inoltre, l'introduzione del pattern MVVM, che aggiunge uno strato intermedio chiamato ViewModel per gestire lo stato dell'applicazione e le interazioni utente, offre un'ulteriore opzione architetturale per gli sviluppatori React Native [13].

In conclusione, l'analisi delle differenze architetturali tra React Native e lo sviluppo nativo fornisce una panoramica approfondita delle opzioni disponibili per gli sviluppatori di applicazioni mobili. La scelta tra i due approcci dipende da una serie di fattori, tra cui i requisiti del progetto, le capacità del team di sviluppo, le prestazioni desiderate e le preferenze personali. La comprensione di queste differenze è essenziale per prendere decisioni informate e garantire il successo del progetto nell'ecosistema mobile in continua evoluzione [13].

2.1.4 Il Model View Controller (MVC)

Il Model View Controller rappresenta uno dei paradigmi architetturali più consolidati e utilizzati nel contesto dello sviluppo software, compreso quello delle applicazioni native per dispositivi mobili [18]. Esso si distingue per la sua capacità di suddividere un'applicazione in tre componenti distinti, ognuno con un compito specifico, promuovendo così una struttura modulare e una separazione chiara delle responsabilità.

Il **Model** rappresenta il nucleo dell'applicazione ed è responsabile della gestione dei dati e delle logiche di business. Questo componente incapsula la struttura dati dell'applicazione e definisce le operazioni che possono essere eseguite su di essa [18].

La **View** si occupa della presentazione dell'interfaccia utente e dell'interazione con l'utente finale. Essa rappresenta la visualizzazione dei dati presenti nel Model, trasformandoli in elementi visivi comprensibili e interattivi [18].

Infine, il **Controller** funge da intermediario tra il Model e la View, gestendo il flusso di dati tra di essi e coordinando le azioni dell'utente. Esso riceve gli input dall'utente attraverso la View, li elabora in base alle logiche di business definite nel Model e aggiorna di conseguenza lo stato dell'applicazione e la visualizzazione dell'interfaccia utente [18].

Nel contesto delle applicazioni native per dispositivi mobili, l'adozione del Model View Controller offre numerosi vantaggi. Innanzitutto, la separazione delle responsabilità tra Model, View e Controller facilita la manutenzione del codice, consentendo agli sviluppatori di apportare modifiche a una parte dell'applicazione senza dover necessariamente modificare le altre. Inoltre, questo approccio favorisce la scalabilità del progetto, consentendo una gestione più efficiente della complessità e delle dimensioni dell'applicazione [18].

Per quanto riguarda gli esempi pratici di implementazione del Model View Controller, si possono trovare numerosi casi di utilizzo in varie applicazioni native disponibili sul mercato. Ad esempio, molte app di social media utilizzano questo modello per gestire la visualizzazione dei contenuti (View), la gestione dei dati degli utenti (Model) e l'interazione con gli elementi dell'interfaccia utente (Controller) [18].

2.1.5 Il Model View ViewModel (MVVM)

L'evoluzione del Model View Controller (MVC) ha portato all'adozione di nuove architetture, tra cui il Model View ViewModel (MVVM), che introduce una componente aggiuntiva chiamata ViewModel [19]. Il ViewModel è una parte fondamentale dell'architettura MVVM, che si posiziona tra il Model e la View.

A differenza del Controller nel classico MVC, il ViewModel non si occupa direttamente della gestione del flusso di dati o delle azioni dell'utente. Piuttosto, funge da intermediario tra il Model e la View, fornendo una rappresentazione dei dati nel formato più adatto per la visualizzazione e la manipolazione nell'interfaccia utente [19].

Una delle principali differenze tra il ViewModel e il Controller è la sua responsabilità nel fornire dati già formattati e pronti per essere visualizzati. Inoltre, il ViewModel è spesso progettato in modo da essere specifico per una particolare View, il che significa che ogni View ha associato un ViewModel dedicato che fornisce i dati e la logica necessari per quella specifica interfaccia utente [19].

Questa separazione delle responsabilità offre numerosi vantaggi, tra cui una maggiore modularità e riusabilità del codice. Inoltre, facilita il testing, in quanto il ViewModel può essere testato in modo isolato rispetto alla View e al Model, consentendo una maggiore copertura dei test e una maggiore sicurezza nel rilascio delle applicazioni [19].

In sintesi, mentre il Model View Controller ha rappresentato a lungo un solido paradigma per lo sviluppo di software, il Model View ViewModel rappresenta un'evoluzione che si adatta meglio alle esigenze delle moderne applicazioni, offrendo una maggiore separazione delle responsabilità e una migliore gestione dei dati e della logica di presentazione [19].

2.1.6 Approccio di React Native

React Native si distingue all'interno del panorama dello sviluppo mobile per la sua capacità di utilizzare controlli dell'interfaccia utente specifici della piattaforma per garantire un rendering nativo. Tale approccio consente agli sviluppatori di creare applicazioni mobili che offrono un'esperienza utente autentica e integrata, mantenendo al contempo la flessibilità e la portabilità del linguaggio JavaScript [13].

In questa sezione, esamineremo attentamente il funzionamento di React Native, con particolare attenzione alla sua architettura a thread singolo e al ruolo cruciale del JavaScript Bridge (JS Bridge) nel facilitare la comunicazione e l'integrazione tra il codice JavaScript e le componenti native dell'applicazione [15].

Architettura a thread singolo

L'architettura a thread singolo in React Native rappresenta un pilastro essenziale della sua struttura, conferendole efficacia e efficienza nello sviluppo di applicazioni mobili. Tale approccio si discosta dalle metodologie comuni delle piattaforme di sviluppo mobile, in cui spesso si fa ricorso a thread multipli per gestire diverse operazioni interne all'applicazione.

In React Native, tutto il codice JavaScript è eseguito su un singolo thread, noto come "thread JS" [17].

L'importanza di questa scelta architeturale risiede principalmente nella semplificazione della gestione dello stato dell'applicazione. L'impiego di un unico thread per l'esecuzione del codice JavaScript elimina potenziali conflitti o complessità legate alla sincronizzazione tra thread multipli. Di conseguenza, gli sviluppatori trovano agevole mantenere e gestire lo stato dell'applicazione in modo uniforme e affidabile [15].

Inoltre, l'architettura a thread singolo si traduce in una maggiore coerenza e prevedibilità nell'interazione con l'interfaccia utente. Poichè tutte le operazioni, incluse la logica applicativa e le interazioni con l'interfaccia utente, avvengono sullo stesso thread, gli aggiornamenti grafici avvengono in modo fluido e reattivo, senza ritardi o interruzioni nell'esperienza utente [16].

Da un punto di vista delle prestazioni, l'architettura a thread singolo favorisce l'ottimizzazione delle risorse dell'applicazione. Grazie alla semplificazione dell'esecuzione su un singolo thread, il sistema può gestire in modo più efficiente le risorse disponibili, garantendo una fruizione fluida e reattiva dell'applicazione, anche su dispositivi con limitate capacità hardware [17].

Ruolo del JS Bridge

Il ruolo del bridge JavaScript (JS Bridge) rappresenta un aspetto cruciale nell'ecosistema di React Native, svolgendo un'importante funzione di mediatore nella comunicazione tra il codice JavaScript e le componenti native dell'applicazione. Questo componente fondamentale si distingue per la sua capacità di garantire un'integrazione senza soluzione di continuità tra l'ambiente JavaScript e quello nativo, contribuendo così alla coerenza e all'efficacia dell'applicazione finale [16, 17].

In particolare, il bridge JS è responsabile della trasmissione bidirezionale delle richieste e delle risposte tra il codice JavaScript e le parti native dell'app. Quando una parte dell'app richiede l'esecuzione di un'azione che coinvolge il codice nativo del dispositivo, come ad esempio l'accesso ai sensori o il rendering di componenti specifici della piattaforma, il bridge JS entra in azione. Esso traduce la richiesta dal linguaggio di programmazione JavaScript al lin-

guaggio nativo della piattaforma di destinazione e la inoltra al thread nativo per l'esecuzione effettiva dell'operazione [10].

Un esempio tangibile di questa operazione è il rendering di un componente React Native. Quando viene richiesto il rendering di un componente, il bridge JS si occupa di trasmettere questa richiesta al thread nativo, che si occupa poi di gestire il rendering utilizzando i controlli nativi specifici della piattaforma. Questo processo assicura un'esperienza utente coerente e integrata, poichè il rendering avviene utilizzando elementi nativi, garantendo al contempo la flessibilità e la potenza dello sviluppo basato su JavaScript [17].

Inoltre, il bridge JS gestisce anche la comunicazione asincrona tra il codice JavaScript e le componenti native dell'applicazione. Ciò significa che è in grado di gestire richieste e risposte che avvengono in modo non sequenziale o concorrente, garantendo una fluidità e una reattività ottimali nell'interazione dell'utente con l'applicazione [10].

In conclusione, il ruolo cruciale del bridge JavaScript in React Native risiede nella sua capacità di facilitare una comunicazione efficace e integrata tra il codice JavaScript e le componenti native dell'applicazione. Grazie a questa componente fondamentale, gli sviluppatori possono creare applicazioni mobili di alta qualità che offrono un'esperienza utente superiore su una vasta gamma di dispositivi e piattaforme [15].

2.1.7 Il ruolo del JS Bridge e l'integrazione di Expo Go

Ora avremo l'opportunità di esaminare più approfonditamente il ruolo imprescindibile svolto dal JS Bridge e l'importanza strategica dell'integrazione di Expo Go nel processo di sviluppo dell'applicazione "Ranger".

Il ruolo del JS Bridge

Il JS Bridge rappresenta un elemento chiave all'interno dell'ecosistema di React Native, fungendo da ponte vitale tra il codice JavaScript e le componenti native dell'applicazione. Questo componente fondamentale è responsabile di garantire una comunicazione bidirezionale efficiente e senza soluzione di continuità tra il mondo dello sviluppo JavaScript e quello nativo delle piattaforme mobile [10].

Il JS Bridge si distingue per la sua capacità di tradurre le richieste provenienti dal codice JavaScript in azioni eseguibili sulle componenti native del dispositivo, e viceversa. Quando

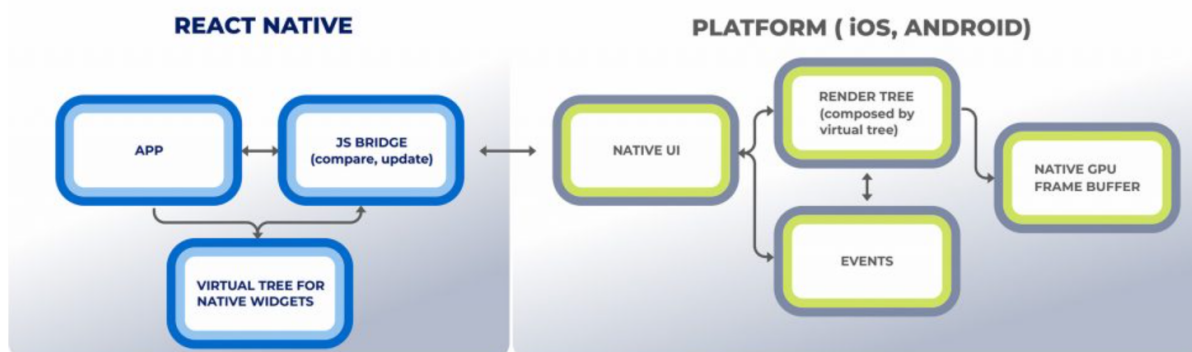


Figura 2.1: Architetture dell'applicazione

il codice JavaScript richiede l'esecuzione di operazioni che coinvolgono funzionalità native, come ad esempio l'accesso al GPS o la gestione di notifiche push, il JS Bridge si attiva per inoltrare la richiesta al codice nativo del sistema operativo [16].

Questo processo di comunicazione bidirezionale garantisce una perfetta integrazione tra il codice JavaScript e le funzionalità native, consentendo agli sviluppatori di sfruttare appieno le potenzialità del dispositivo senza compromettere la coerenza dell'esperienza utente [17].

L'integrazione di Expo Go

Parallelamente al ruolo centrale svolto dal JS Bridge, l'integrazione di Expo Go assume un'importanza strategica nel processo di sviluppo dell'applicazione "Ranger". Expo Go rappresenta una piattaforma completa e unificata per lo sviluppo di applicazioni React Native, offrendo agli sviluppatori una serie di strumenti e funzionalità che semplificano significativamente il ciclo di sviluppo e distribuzione dell'applicazione [11].

Uno dei principali vantaggi offerti da Expo Go è la sua facilità d'uso [11]. Grazie a un'ampia gamma di funzionalità predefinite e componenti prontamente disponibili, Expo Go permette agli sviluppatori di concentrarsi esclusivamente sulla logica applicativa e sulle funzionalità distintive dell'app, senza doversi preoccupare degli intricati dettagli tecnici legati alla configurazione e distribuzione dell'applicazione su diverse piattaforme [11].

In aggiunta, la realizzazione individuale dell'applicazione mi ha portato a sfruttare Expo Go, un robusto framework che mette a disposizione un insieme sofisticato di strumenti per la gestione delle risorse e delle dipendenze [11]. Tale piattaforma ha notevolmente semplificato

il processo di sviluppo, contribuendo a ottimizzare l'efficienza del mio lavoro. L'approccio integrato di Expo Go ha consentito di canalizzare le mie energie verso la progettazione di un'esperienza utente coinvolgente e di alto livello, riducendo simultaneamente i tempi di sviluppo e accelerando il lancio dell'applicazione sul mercato [11].

In conclusione, sia l'impiego del JS Bridge che l'integrazione di Expo Go hanno rivestito un ruolo imprescindibile nell'iter di sviluppo dell'applicazione "Ranger". Questi elementi cruciali hanno consentito la creazione di un prodotto mobile di eccellente qualità, in grado di garantire un'esperienza utente superiore su una vasta gamma di piattaforme e dispositivi. Parimenti, hanno facilitato una gestione efficiente e ottimizzata delle risorse di sviluppo, concorrendo al successo complessivo del progetto [10, 16, 11].

Capitolo 3

Implementazione dell'applicazione

3.1 Struttura Generale

La logica primaria che guida il funzionamento dell'applicazione é incapsulata nella classe `App.js`. Questo componente svolge un ruolo cruciale nel coordinare il flusso complessivo dell'applicazione, gestendo una serie di operazioni cruciali, che spaziano dalla gestione della posizione iniziale all'autenticazione dell'utente, passando attraverso il controllo del login e la navigazione tra le diverse schermate.

3.1.1 Gestione dello Stato con React Hooks

Per gestire in modo efficace lo stato dell'applicazione e gli effetti collaterali, `App.js` sfrutta ampiamente gli Hook di React, una funzionalità chiave di questo framework. Gli hook, come `useState` e `useEffect`, consentono di integrare lo stato e i comportamenti reattivi all'interno dei componenti funzionali in modo intuitivo.

Per esempio, `useState` é un hook che permette di dichiarare variabili di stato all'interno di componenti funzionali. Con questa dichiarazione, é possibile accedere sia al valore corrente dello stato che a una funzione associata per modificarlo. Ciò é particolarmente utile per mantenere lo stato dell'applicazione coerente e aggiornato, come nel caso delle credenziali dell'utente durante il processo di login.

Dall'altra parte, `useEffect` é un altro hook fondamentale che viene utilizzato per eseguire operazioni di cambiamento all'interno dei componenti. La sua esecuzione avviene dopo

che il processo di rendering del componente é stato completato, consentendo la gestione di effetti collaterali in risposta a cambiamenti di stato, props o altri eventi nell'applicazione. Questo hook é particolarmente utile quando si devono gestire operazioni asincrone. Quando un componente viene renderizzato per la prima volta, `useEffect` viene attivato. All'interno di questo effetto, é possibile eseguire operazioni asincrone senza bloccare il thread principale, garantendo che l'interfaccia utente rimanga reattiva.

Funzione Asincrona

Una funzione asincrona é una funzione JavaScript che consente l'utilizzo della keyword `'await'` per attendere il completamento di operazioni asincrone, come chiamate API, richieste di rete o altre operazioni che richiedono tempo. Questo consente di eseguire operazioni che potrebbero richiedere del tempo, come ad esempio richieste di permessi o ottenimento di dati dalla rete, senza bloccare il thread principale e garantendo che l'interfaccia utente rimanga reattiva.

3.2 Sviluppo Front-End

3.2.1 Richiesta di Permessi GPS

Prima ancora di procedere con l'inserimento delle credenziali, gli utenti si trovano di fronte a una richiesta cruciale: una notifica che chiede il consenso per accedere alla posizione del GPS. Questo approccio proattivo consente all'applicazione di fornire servizi basati sulla posizione, se necessario, migliorando cosí l'esperienza utente e offrendo funzionalità aggiuntive.

La richiesta di permessi GPS rappresenta un elemento chiave nell'architettura dell'applicazione, con un impatto significativo sulla funzionalità e sull'esperienza utente. Attraverso un processo ben progettato e una gestione attenta della privacy, l'applicazione riesce a sfruttare i dati di localizzazione per offrire servizi avanzati e personalizzati, migliorando l'efficacia operativa dei tecnici sul campo.

Personalizzazione dei Servizi

Grazie all'accesso alla posizione GPS, Ranger non solo monitora in tempo reale lo stato dei dispositivi remoti, ma utilizza anche questa informazione per offrire agli utenti esperienze personalizzate e altamente efficienti. Attraverso l'integrazione della localizzazione dell'utente, l'applicazione adatta dinamicamente la visualizzazione delle mappe e dei punti di interesse alla posizione corrente dell'utente. Questo significa che gli utenti possono vedere esattamente dove si trovano rispetto ai dispositivi monitorati, migliorando non solo la precisione delle informazioni ricevute ma anche la loro reattività nel gestire eventuali problematiche.

Immagina di essere un tecnico che gestisce una vasta rete di antenne radio sparse in tutto il territorio italiano: grazie alla funzionalità di localizzazione di Ranger, puoi vedere istantaneamente sulla mappa quali dispositivi sono più vicini a te e quali richiedono la tua attenzione immediata. Questo non solo ottimizza il tempo di risposta alle emergenze, ma riduce anche il rischio di interruzioni nei servizi, migliorando complessivamente l'affidabilità delle reti gestite.

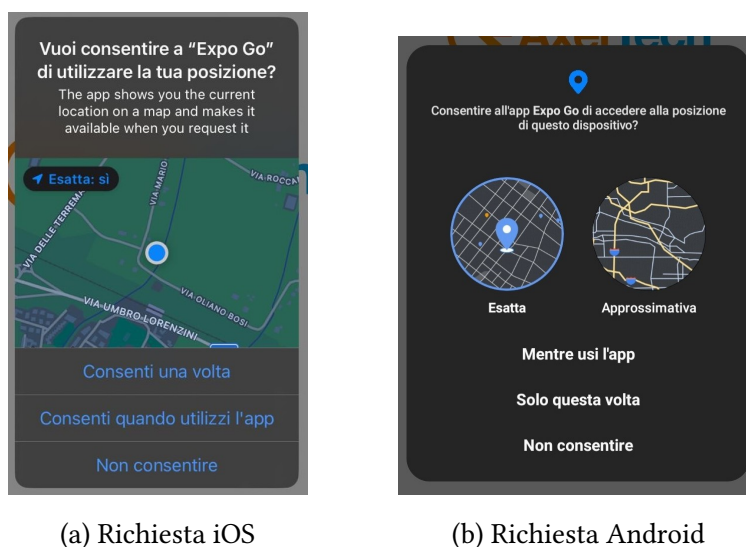


Figura 3.1: Confronto tra richiesta di posizione su iOS e Android

Inoltre, l'adattamento delle informazioni in base alla posizione dell'utente rende l'interazione con Ranger più intuitiva e rilevante. Gli utenti possono navigare attraverso le funzionalità dell'applicazione con maggiore facilità e fiducia, sapendo di avere sempre a disposizione

le informazioni più pertinenti per il loro contesto operativo. Questo tipo di personalizzazione non solo ottimizza l'efficacia dell'uso dell'applicazione, ma contribuisce anche a rendere l'esperienza utente complessivamente più soddisfacente e gratificante.

3.2.2 Gestione delle richieste dei Permessi e delle Posizione

Il primo metodo nel componente `App.js` è un `useEffect` che gestisce la richiesta di permessi e l'ottenimento della posizione iniziale al momento dell'avvio dell'applicazione. Questo `useEffect` viene eseguito non appena il componente viene montato, garantendo così che l'applicazione inizi a funzionare con le informazioni di localizzazione necessarie fin dal primo momento.

L'uso di `useEffect` in questo contesto consente di incorporare una funzione asincrona che richiede i permessi di accesso alla posizione GPS e, se concessi, ottiene la posizione corrente dell'utente. Questo processo è cruciale perché permette all'applicazione di prepararsi a fornire contenuti e funzionalità basati sulla posizione geografica, migliorando significativamente l'esperienza utente.

Il codice all'interno di `useEffect` è strutturato per gestire vari scenari. In primo luogo, viene inviata una richiesta di permessi tramite la

funzione `Location.requestForegroundPermissionsAsync()`. Se l'utente concede i permessi, l'applicazione procede a ottenere la posizione corrente

con `Location.getCurrentPositionAsync()`. Questi dati vengono poi memorizzati nello stato dell'applicazione, permettendo di utilizzarli in seguito per funzionalità che dipendono dalla localizzazione.

```
1 // Effetto per richiedere i permessi e ottenere la posizione all'
  avvio dell'app
2 useEffect(() => {
3   (async () => {
4     let { status } = await Location.
      requestForegroundPermissionsAsync();
5     if (status !== 'granted') {
6       setErrorMsg('Permission to access location was denied');
7     }
    return;
  })();
}
```

```
8     }  
9     let location = await Location.getCurrentPositionAsync({});  
10    setLocation(location);  
11  })();  
12 }, []);
```

3.2.3 Schermata di Login

All'apertura dell'applicazione, gli utenti sono accolti da una schermata di Login che rappresenta il punto di ingresso essenziale per accedere alle funzionalità dell'applicazione.

Questa schermata, progettata con cura per garantire un'esperienza utente fluida ed efficiente, è caratterizzata da elementi visivi chiaramente distinti che facilitano il riconoscimento e l'utilizzo.

In primo piano, il logo distintivo dell'azienda, insieme al nome dell'applicazione, si stagliano con evidenza, offrendo un'immediata identificazione del contesto di utilizzo. Questo design strategico non solo conferisce un senso di appartenenza e affidabilità agli utenti, ma li orienta anche verso l'obiettivo principale: l'accesso alle risorse e alle funzionalità offerte dall'applicazione.

Sotto il logo e il nome dell'applicazione, sono posizionati due campi di inserimento, chiaramente etichettati per l'username e la password. Questi campi forniscono agli utenti un mezzo diretto e sicuro per immettere le proprie credenziali di accesso, consentendo loro di accedere rapidamente all'ambiente protetto dell'applicazione.

Contrariamente a molte altre piattaforme, non è presente un pulsante di registrazione ('Sing up') sulla schermata di Login. Questa scelta deliberata è il riflesso della natura specifica dell'applicazione, che è riservata esclusivamente a un team selezionato di tecnici. Questi professionisti vengono registrati in modo centralizzato in un database dedicato e ricevono le credenziali di accesso necessarie per l'utilizzo dell'applicazione da parte dell'amministratore del sistema. Di conseguenza, l'accesso è limitato a coloro che sono stati autorizzati, eliminando la necessità di un processo di registrazione aperto.



Figura 3.2: Schermata di LogIn

3.2.4 Autenticazione Utente

Una volta presa la decisione di fornire o meno l'accesso al GPS, gli utenti procedono con l'inserimento delle proprie credenziali di accesso. Il processo di autenticazione è progettato per essere rapido ed efficiente: in caso di inserimento corretto di entrambe le credenziali, l'accesso all'interno dell'applicazione è consentito e gli utenti sono immediatamente indirizzati alle funzionalità principali. Questo flusso semplificato garantisce un'esperienza utente senza interruzioni, permettendo ai tecnici di iniziare subito a utilizzare le risorse dell'applicazione senza ritardi.

Tuttavia, nel caso in cui le credenziali inserite non siano corrette, compare un messaggio d'allarme che avvisa l'utente dell'errore e lo invita a riprovare. Questo sistema di avviso è fondamentale per garantire la sicurezza e l'integrità dell'accesso all'applicazione. L'uso di messaggi di errore chiari e immediati non solo aiuta a prevenire accessi non autorizzati, ma riduce anche la frustrazione degli utenti, guidandoli efficacemente verso la corretta immissione delle informazioni di accesso.

Questo approccio rigoroso alla gestione delle credenziali e all'autenticazione degli utenti è una componente chiave della sicurezza complessiva dell'applicazione, assicurando che solo i tecnici autorizzati possano accedere e utilizzare le funzionalità offerte.

3.2.5 Gestione del Login

Alla pressione del pulsante “Login”, viene invocato il metodo `handleLogin`, il quale é responsabile della verifica delle credenziali fornite dall’utente utilizzando l’ `APIBackend`. Questo metodo é progettato per gestire in modo asincrono la richiesta di autenticazione, garantendo una risposta rapida e adeguata a seconda della validità delle credenziali inserite.

```
1 // Chiamata all'API per ottenere i dettagli dell'utente
2 const handleLogin = async () => {
3   const LoggedAccount = api.getAccountUser(username,password);
4   if(LoggedAccount == null) {
5     setIsLoggedIn(false);
6     setLoginError('Credenziali non valide. Per favore, riprova.')
```

In dettaglio, `handleLogin` funziona come segue:

Chiamata all'API

Viene eseguita una chiamata all'API `api.getAccountUser` con i parametri `username` e `password` forniti dall'utente. Questa chiamata verifica se le credenziali corrispondono a un account valido nel sistema.

Verifica della Risposta

Se la risposta dell'API é null, significa che le credenziali non sono valide. In tal caso, il metodo aggiorna lo stato dell'applicazione impostando `setIsLoggedIn(false)` e mostrando un messaggio di errore tramite `setLoginError('Credenziali non valide. Per favore, riprova.')`. Questo feedback immediato é fondamentale per informare l'utente dell'errore e invitarlo a riprovare.

Accesso Consentito

Se invece le credenziali sono corrette, l'applicazione aggiorna lo stato impostando `setIsLoggedIn(true)` e memorizza le informazioni dell'utente autenticato con `setLoggedInUser(LoggedAccount)`. Questo permette all'utente di accedere immediatamente alle funzionalità principali dell'applicazione.

Il metodo `handleLogin` è una componente chiave del processo di autenticazione, garantendo che solo gli utenti con credenziali valide possano accedere all'applicazione. La gestione efficiente dello stato e la comunicazione chiara con l'utente sono essenziali per una buona esperienza d'uso, migliorando la sicurezza e l'affidabilità del sistema di login.

3.3 Barra di Navigazione Principale

Dopo aver completato il processo di accesso, l'intera esperienza utente si trasferisce verso una barra di navigazione principale, strategicamente posizionata nella parte inferiore dello schermo. Questa barra di navigazione, gestita dalla classe `navBarBottom`, assume un ruolo fondamentale nell'orientare gli utenti attraverso le diverse funzionalità dell'applicazione. Utilizzando React Navigation, una libreria di navigazione per app React Native, la `navBarBottom` definisce un navigatore a schede che semplifica la navigazione e l'interazione dell'utente con l'applicazione.

All'interno di questa barra di navigazione inferiore, sono presenti tre schede distinte: "Account", "MapPage" e "MapStatus", ciascuna associata a un'icona e un'etichetta specifica. Questo design intuitivo offre agli utenti un accesso rapido alle diverse sezioni dell'applicazione, consentendo loro di esplorare le funzionalità con facilità.

Inizialmente, viene creato un oggetto `Tab` che funge da navigatore a schede, ospitando le diverse schermate dell'applicazione. Il componente `Tab.Navigator` contiene l'insieme completo di schede configurate, con la schermata iniziale visualizzata impostata su "MapPage," definita mediante l'opzione `initialRouteName`. Per ogni scheda, viene utilizzato il componente `Tab.Screen` per definire le relative opzioni. L'opzione `component` specifica il componente associato a quella particolare scheda, mentre l'opzione `options` consente la personalizzazione di aspetti come etichette e icone, contribuendo così a una navigazione chiara e intuitiva.

3.4 Funzionalità Principali

3.4.1 Schermata "Account"

Cliccando sul primo pulsante, denominato "Account", della `navBar`, gli utenti vengono indirizzati all'area riservata all'utente. Qui, all'interno della sezione "Account", vengono esclusivamente visualizzati i dati pertinenti all'utente che ha recentemente eseguito l'accesso, come ad esempio il nome, il cognome e l'indirizzo email. Questo livello di personalizzazione è reso possibile grazie al componente funzionale `AccountUser`, il quale accetta un parametro denominato `user`, rappresentante l'oggetto contenente le informazioni dell'utente che ha effettuato il login. Di conseguenza, la sezione "Account" offre un ambiente dedicato e mirato alla visualizzazione delle informazioni specifiche dell'utente attualmente autenticato.

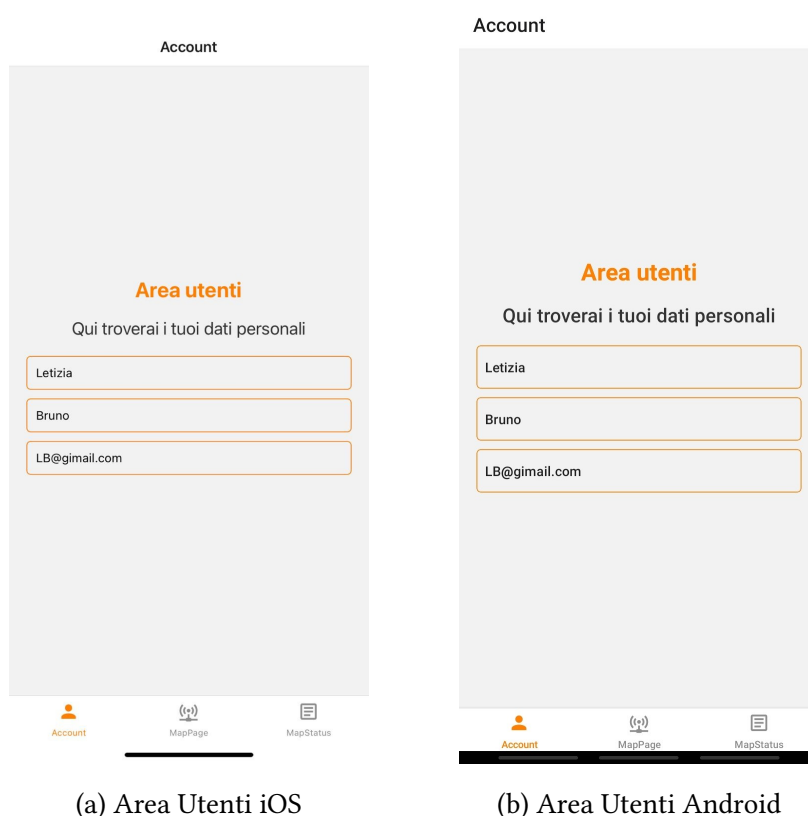


Figura 3.3: Confronto tra area utenti su iOS e Android

Questo approccio mirato non solo fornisce una chiara rappresentazione visiva delle informazioni personali dell'utente, ma offre anche un accesso diretto e agevole alle stesse, contribuendo così a migliorare complessivamente l'esperienza utente all'interno dell'applicazione. La combinazione di una barra di navigazione ben strutturata e una sezione "Account" personalizzata crea un ambiente intuitivo e user-friendly che facilita l'utilizzo e la comprensione dell'applicazione da parte degli utenti.

3.4.2 Schermata "MapPage"

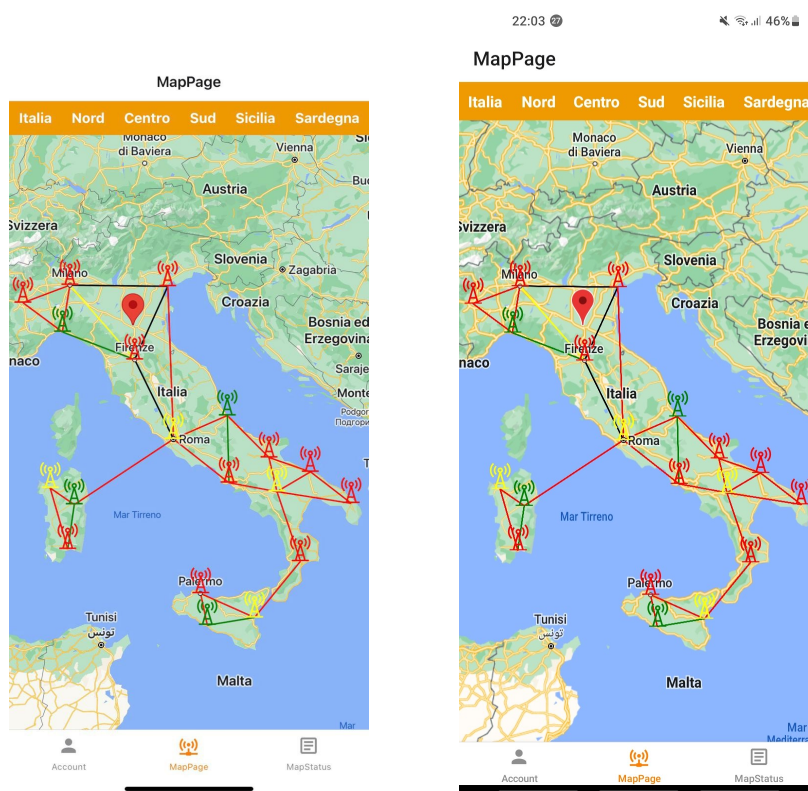
Cliccando sul pulsante centrale "MapPage", l'utente viene trasportato in un'esperienza visiva che si materializza sotto forma di una mappa dell'Italia. Questa mappa è molto più di una semplice rappresentazione geografica, è un'interfaccia interattiva che offre una panoramica dettagliata dei siti sparsi sul territorio nazionale. Ogni sito è simboleggiato da un'icona, rappresentata come un'antenna, e caratterizzato da un colore che indica il suo stato attuale: Rosso - In errore, Giallo - Problemi di scarsa rilevanza, Verde - Non ci sono problemi. Le connessioni tra i vari siti sono altrettanto evidenti, tracciate con gli stessi colori delle antenne che evidenziano lo stato delle connessioni stesse. Inoltre, la mappa mostra anche la posizione del dispositivo dell'utente che ha effettuato l'accesso, evidenziata da un marker distintivo sulla mappa che rappresenta la posizione attuale dell'utente, sempre che abbia accettato il permesso alla localizzazione del GPS.

Ma la mappa non è solo una visione statica. È un ambiente dinamico che offre agli utenti la possibilità di esplorare e interagire con le informazioni geografiche in modi diversi. Al suo apice, nella parte superiore della schermata, si trovano diversi tasti che consentono di navigare tra le varie dimensioni della mappa in modo dinamico. Ogni tasto rappresenta un'area geografica specifica, e premendolo, la mappa si adatta istantaneamente alla dimensione selezionata. Questa funzionalità offre agli utenti un controllo completo sulla visualizzazione della mappa, consentendo loro di concentrarsi su aree specifiche o di avere una visione più ampia del territorio italiano.

Il cuore pulsante di questa esperienza è la classe "MapPage.js", che funge da direttrice per tutte le operazioni legate alla mappa. Incorporando una serie di componenti specializzati e librerie di supporto, come React Native Maps, questa classe è il motore che alimenta la visualizzazione fluida e interattiva della mappa. Attraverso React Native Maps, integrato

con il provider di Google, è possibile incorporare direttamente la mappa di Google Maps nell'applicazione, garantendo una visualizzazione accurata e aggiornata del territorio italiano.

Ma la classe “MapPage.js” non si ferma qui. Essa si interfaccia anche con le API di backend, utilizzate per recuperare i dati dal database, e coordina l'interazione con altri componenti cruciali come “MyPosition”, “Sito” e “Connessione”. Questi componenti si occupano rispettivamente della visualizzazione della posizione in tempo reale dell'utente, della gestione dei siti sulla mappa e delle connessioni tra di essi.



(a) MapPage iOS

(b) MapPage Android

Figura 3.4: Confronto tra visualizzazione MapPage su iOS e Android

All'interno, la classe utilizza gli Hooks `useState` e `useEffect` per gestire gli stati relativi alla lista dei siti e delle connessioni, garantendo un aggiornamento costante e dinamico delle informazioni visualizzate sulla mappa. Utilizzando un ref denominato `mapViewRef`, si accede al componente `MapView` per gestire le animazioni e garantire una visualizzazione fluida dei cambiamenti.

Nel blocco di ritorno, la classe renderizza una serie di pulsanti posizionati strategicamente nella parte superiore della schermata. Ogni pulsante è associato a una specifica regione geografica italiana e, quando viene premuto, attiva la funzione `handleButtonPress`, che muove automaticamente la mappa sulla regione selezionata. Infine, il componente `MapView`, utilizzando la libreria `react-native-maps`, visualizza la mappa inizializzandola con una regione centrata sull'Italia, mostrando la posizione dell'utente e i marker dei siti insieme alle linee di connessione tra di essi sulla mappa.

Questa implementazione meticolosa garantisce una chiara e intuitiva visualizzazione delle informazioni geografiche e di connessione all'interno dell'applicazione, offrendo agli utenti un'esperienza coinvolgente e informativa.

Esploreremo ora nello specifico le tre classi che compongono "MapPage.js", cioè "MyPosition", "Sito" e "Connessione".

MyPosition

La classe "MyPosition" svolge un ruolo fondamentale nell'identificare e rappresentare la posizione dell'utente all'interno dell'applicazione. Grazie all'utilizzo del modulo

`expo-location`, è in grado di acquisire i permessi di localizzazione e di individuare la posizione geografica corrente dell'utente. Questa informazione viene poi visualizzata sotto forma di marker sulla mappa, fornendo un chiaro punto di riferimento per la posizione attuale dell'utente. La gestione della posizione avviene attraverso lo stato `origin`, che traccia la posizione attuale dell'utente e si aggiorna costantemente durante l'utilizzo dell'applicazione. Al momento del montaggio del componente, viene avviato un effetto che richiede i permessi di localizzazione e recupera la posizione corrente dell'utente mediante la funzione asincrona `getLocationPermission`. Successivamente, questa posizione viene aggiornata nello stato `origin`, garantendo un'indicazione sempre precisa e aggiornata della posizione dell'utente.

Oltre alla rappresentazione visuale sulla mappa, il componente "MyPosition" offre la possibilità di interagire con il marker. Il marker può essere trascinato per aggiornare manualmente la posizione e fornisce un `Callout` che segnala la presenza dell'utente con il testo "I'm here". Questo callout fornisce un'informazione immediata e visiva della posizione corrente, migliorando l'interazione dell'utente con la mappa.

La possibilità di trascinare il marker e il callout informativo migliorano l'esperienza utente, rendendo l'interazione con l'applicazione più intuitiva e immediata.

```
1
2 // Funzione asincrona per ottenere i permessi di localizzazione e la
   posizione corrente
3
4 async function getLocationPermission() {
5   let status = await Location.requestForegroundPermissionsAsync();
6   console.log('Status permission: ', status);
7   if (status.status !== 'granted') {
8     alert('Permission denied');
9     return;
10  }
11  let location = await Location.getCurrentPositionAsync({});
12  const current = {
13    latitude: location.coords.latitude,
14    longitude: location.coords.longitude,
15  };
16  setOrigin(current);
17 }
```

Sito

La classe “Sito” rappresenta e gestisce i siti sulla mappa, utilizzando la libreria `react-native-maps` per visualizzare ciascun sito come un marker con un'icona specifica (un'antenna). Ogni marker è dotato di un `Callout` che fornisce ulteriori dettagli e consente di condividere lo stato del sito. Le principali funzionalità e logiche di questa classe includono:

Gestione dello Stato Locale “currentSite”: Lo stato locale `currentSite` tiene traccia del sito corrente. Ogni volta che le proprietà del componente cambiano, il `currentSite` viene aggiornato di conseguenza.

Configurazione degli Handler per le Notifiche: Utilizzando Expo Notifications, la classe configura gli handler per gestire le notifiche, assicurandosi che vengano mostrate, suonate e contate correttamente.

Invio delle Notifiche: La funzione `sendNotification` invia una notifica quando lo stato del sito cambia, specificamente se lo stato è critico.

```
1 // Funzione per inviare la notifica
2 const sendNotification = async (site) => {
3   try {
4     const notificationContent = {
5       title: 'Lo stato del sito di ${site?.description} e
6         cambiato',
7       body: 'Errore',
8     };
9     console.log('Notification content:', notificationContent)
10    await Notifications.scheduleNotificationAsync({
11      content: notificationContent,
12      trigger: null,
13    });
14  } catch (error) {
15    console.error('Errore invio della notifica:', error);
16  }
17 }
```

Definizione dei Colori del Marker: La funzione `getSiteColor` definisce il colore del marker in base allo stato del sito, rendendo immediatamente visibile se un sito è operativo, ha problemi minori o è in errore.

Condivisione dello Stato del Sito: La funzione `shareStato` permette di condividere lo stato del sito tramite la funzionalità di condivisione nativa del dispositivo. Il messaggio condiviso include una descrizione dello stato attuale del sito.

Il rendering del marker e del *Callout* offre un'interfaccia intuitiva per gli utenti, consentendo loro di accedere facilmente alle informazioni sui siti e di condividere lo sta-

to con altri utenti. Questa implementazione migliora notevolmente l'usabilità dell'applicazione, fornendo un feedback visivo immediato e funzionalità di condivisione integrate.

Connection

La classe "Connection" è fondamentale per rappresentare visivamente le connessioni tra i siti sulla mappa all'interno dell'applicazione, utilizzando il componente `Polyline` fornito da `react-native-maps`. Ogni connessione viene visualizzata come una linea sulla mappa, il cui colore varia in base allo stato della connessione, offrendo così una rappresentazione immediata e intuitiva delle condizioni delle varie connessioni.

Le principali funzionalità e logiche di questa classe includono la gestione del colore della connessione tramite lo stato locale `connectionColor`, che viene aggiornato ogni volta che cambia lo stato della connessione. Questo garantisce che la visualizzazione sia sempre coerente con le condizioni attuali della rete. La funzione `getConnectionColor` determina il colore della linea in base allo stato della connessione, utilizzando il verde per le connessioni senza problemi, il giallo per le connessioni con problemi minori e il rosso per le connessioni con errori gravi, con il nero riservato per stati non definiti.

Il componente `Polyline` è utilizzato per disegnare la linea che rappresenta la connessione sulla mappa. Le coordinate di origine e destinazione della connessione sono passate come proprietà al componente e vengono convertite in valori numerici per il rendering. Utilizzando l'hook `useEffect`, il colore della connessione viene aggiornato ogni volta che cambia lo stato della connessione, assicurando che l'utente abbia sempre una visualizzazione aggiornata e precisa.

La rappresentazione visiva delle connessioni tramite linee colorate facilita la comprensione e l'analisi delle reti all'interno dell'applicazione, permettendo agli utenti di identificare rapidamente eventuali problemi e di agire di conseguenza. In sintesi, la classe "Connection" offre una rappresentazione visiva chiara e intuitiva delle connessioni tra i siti sulla mappa, evidenziando visivamente lo stato delle connessioni e facilitando la comprensione e l'analisi delle reti all'interno dell'applicazione.

3.4.3 Schermata "MapStatus"

Esaminando attentamente l'ultima voce della nostra barra di navigazione, "MapStatus", ci troviamo di fronte a una pagina dedicata che offre una panoramica completa dello stato dei siti presenti nell'applicazione. Questa sezione assume un ruolo fondamentale nel fornire agli utenti un resoconto dettagliato e aggiornato delle condizioni operative dei siti monitorati.

All'interno di questa pagina, una lista dinamica di siti si adatta costantemente allo stato attuale di ciascun sito, offrendo un'immediata visualizzazione delle eventuali problematiche. Questo é possibile grazie a un sistema di colori distintivi, dove ogni colore corrisponde a uno specifico stato operativo. Ad esempio, lo stato di errore viene evidenziato in rosso, mentre altri stati sono contrassegnati con una gamma diversificata di colori, precedentemente definiti per indicare varie condizioni.

Una caratteristica particolarmente utile di questa sezione é il pulsante di filtro, posizionato strategicamente nella parte inferiore della pagina. Questo pulsante consente agli utenti di concentrarsi esclusivamente sui siti che presentano problemi, semplificando ulteriormente il processo di identificazione delle criticità e rendendo più efficiente la gestione delle risorse e delle attività di manutenzione.

Tra le funzionalità principali offerte da questa classe, vi é l'inizializzazione e l'aggiornamento dinamico dei dati relativi ai siti e alle loro condizioni operative. Durante il caricamento della pagina, i dati vengono recuperati tramite chiamate API al backend, garantendo così che la lista dei siti rifletta sempre lo stato più aggiornato e accurato.

Inoltre, il sistema di filtro dei siti offre agli utenti la flessibilità di visualizzare solo i siti in errore o tutti i siti, in base alle proprie esigenze e priorità operative. Questo meccanismo consente una rapida focalizzazione sulle criticità più urgenti, ottimizzando il tempo e le risorse impiegate per la risoluzione dei problemi.

Un elemento chiave per garantire un'esperienza utente chiara e intuitiva é la rappresentazione visiva dei dati, resa possibile attraverso l'uso di colori distintivi e la chiara indicazione dello stato operativo di ciascun sito. Questo approccio consente agli utenti di identificare rapidamente le criticità e di prendere decisioni informate in merito alle azioni da intraprendere.

In conclusione, "MapStatus" svolge un ruolo cruciale nell'offrire agli utenti un'interfaccia chiara e completa per monitorare lo stato operativo dei siti. Grazie alle sue funzionalità di



(a) MapStatus generica

(b) Errori filtrati

Figura 3.5: Confronto tra visualizzazione MapStatus

visualizzazione dinamica, filtraggio dei dati e rappresentazione visiva dei risultati, questa sezione contribuisce significativamente a migliorare l'efficienza e l'efficacia delle operazioni di monitoraggio e manutenzione.

3.5 Classe "APIBackend"

La classe "APIBackend" assume un ruolo cruciale all'interno dell'applicazione, fungendo da simulatore di un backend per la gestione dei dati relativi ai siti, alle connessioni e agli utenti registrati. Questa componente é fondamentale per consentire il funzionamento dell'applicazione in assenza di un vero server aziendale, fornendo dati dinamici e simulando interazioni con un sistema remoto.

Una delle prime funzionalità di rilievo di questa classe é la gestione dei dati iniziali, rappresentati in formato JSON. Questi dati includono informazioni dettagliate sui siti e sulle connessioni, quali ID univoci, coordinate geografiche, descrizioni e stato operativo corrente. Inoltre, la presenza di un contatore periodico all'interno della classe consente di simulare il cambiamento dinamico dello stato dei siti e delle connessioni nel tempo, fornendo così una rappresentazione realistica delle condizioni operative.

I metodi principali offerti da questa classe sono essenziali per il recupero dei dati fondamentali. Il metodo `getSites` restituisce la lista aggiornata dei siti, mentre il metodo `getConnection` fornisce informazioni sulle connessioni tra i siti. Ogni volta che viene richiamato il metodo `getSites`, il contatore interno si incrementa, influenzando lo stato dei siti e garantendo un flusso dinamico di informazioni all'interno dell'applicazione.

Inoltre, la classe "APIBackend" gestisce anche le informazioni sugli utenti registrati, fornendo un meccanismo per l'accesso e la verifica delle credenziali degli utenti. Il metodo `getAccountUser` consente di verificare l'autenticità di un account utente in base alle credenziali fornite, garantendo così un livello di sicurezza e autenticazione all'interno dell'applicazione.

```
1  getAccountUser(userName, password) {  
2      return this.users.find(user => user.userName === userName &&  
        user.Password === password);  
3  }
```

Riassumendo, la classe "APIBackend" svolge un ruolo cruciale nell'offrire dati dinamici e simulare interazioni con un backend remoto all'interno dell'applicazione. Grazie alle sue funzionalità di gestione dei dati, simulazione dello stato operativo dei siti e autenticazione degli utenti, questa componente contribuisce significativamente al funzionamento e alla completezza dell'applicazione, garantendo un'esperienza utente fluida e realistica.

La decisione di utilizzare dati fittizi o "mock" all'interno della prima versione di Ranger è stata una scelta temporanea, mirata a facilitare lo sviluppo iniziale dell'applicazione. Questo approccio ha permesso di concentrarsi sulla creazione e sulla progettazione dell'interfaccia utente e della logica dell'applicazione senza dipendere immediatamente dall'accesso ai veri dati del backend dell'azienda.

Tuttavia, dopo aver verificato il funzionamento e la stabilità dei APIBackend nel simulare i dati e le interazioni necessarie, si è optato per una transizione verso l'utilizzo delle vere API per collegarsi direttamente al backend dell'azienda. Questo passaggio è avvenuto nella seconda versione dell'applicazione, consentendo un'esperienza più autentica e un accesso in tempo reale ai dati aziendali.

L'integrazione delle vere API ha comportato una maggiore precisione e affidabilità nei dati visualizzati all'interno dell'applicazione, consentendo ai tecnici di accedere a informazioni aggiornate e pertinenti direttamente dal sistema aziendale. Questa transizione ha reso l'applicazione Ranger ancora più potente e funzionale, fornendo un ambiente di lavoro più efficace e produttivo per il team di tecnici.

Capitolo 4

Valutazione dell'Applicazione

4.1 Introduzione alla Valutazione

In questo capitolo verrà esaminato il processo di valutazione dell'applicazione Ranger tramite un sondaggio condotto tra i tecnici che l'hanno utilizzata. Questa valutazione è cruciale per comprendere l'efficacia di Ranger nell'ambiente operativo reale e per identificare punti di forza e aree di miglioramento.

L'applicazione Ranger è attualmente nella sua prima fase di realizzazione e non è ancora distribuita al pubblico. Questo stato iniziale di sviluppo rende la valutazione ancora più importante, poiché permette di identificare tempestivamente eventuali criticità e di apportare i necessari miglioramenti prima del rilascio definitivo. Ranger è stata realizzata per funzionare sia su dispositivi Android che iOS, pertanto, la valutazione include anche un'analisi comparativa delle prestazioni e dell'usabilità dell'applicazione su entrambe le piattaforme.

Il processo di valutazione è stato articolato in due fasi principali: una fase iniziale di raccolta del feedback e una successiva fase di miglioramento e rivalutazione. Nella prima fase, ai tecnici è stato somministrato un questionario dettagliato progettato per raccogliere feedback sull'esperienza d'uso dell'applicazione, sulle funzionalità utilizzate e su eventuali difficoltà riscontrate. Le domande sono state strutturate per coprire vari aspetti dell'interfaccia utente (UI), dell'esperienza utente (UX) e delle funzionalità tecniche.

Le risposte raccolte dal primo questionario hanno fornito preziose informazioni sui punti di forza e sulle criticità dell'applicazione. Sulla base di questi feedback, sono state apportate

modifiche e miglioramenti all'applicazione Ranger. Questo processo iterativo di sviluppo ha permesso di implementare miglioramenti significativi sia nell'usabilità dell'interfaccia che nelle funzionalità tecniche, rispondendo alle esigenze e alle aspettative degli utenti.

Successivamente, un secondo sondaggio è stato sottoposto agli stessi tecnici. Questo nuovo questionario includeva le stesse domande del primo per valutare se i miglioramenti apportati fossero appropriati e per monitorare l'evoluzione della percezione dell'applicazione da parte degli utenti. Inoltre, sono state aggiunte domande specifiche per raccogliere feedback sui cambiamenti implementati e per identificare ulteriori aree di miglioramento.

In sintesi, il processo di valutazione dell'applicazione Ranger, condotto attraverso un sondaggio dettagliato tra tecnici specializzati, è stato fondamentale per comprendere l'efficacia dell'applicazione in un contesto operativo reale. Questo processo ha permesso di identificare i punti di forza e le criticità dell'app, di apportare miglioramenti significativi e di pianificare ulteriori sviluppi per il futuro.

4.2 Procedura di Valutazione

La valutazione dell'applicazione Ranger è stata effettuata attraverso un sondaggio che ha coinvolto una cinquantina di tecnici specializzati nel monitoraggio dei dispositivi NMS. Il processo di valutazione è stato articolato in due fasi principali: una fase iniziale di raccolta del feedback e una successiva fase di miglioramento e rivalutazione.

Nella prima fase, ai tecnici è stato somministrato un questionario dettagliato progettato per raccogliere feedback sull'esperienza d'uso dell'applicazione, sulle funzionalità utilizzate e su eventuali difficoltà riscontrate. Le domande sono state strutturate per coprire vari aspetti dell'interfaccia utente (UI), dell'esperienza utente (UX) e delle funzionalità tecniche.

Le risposte raccolte dal primo questionario hanno fornito preziose informazioni sui punti di forza e sulle criticità dell'applicazione. Sulla base di questi feedback, sono state apportate modifiche e miglioramenti all'applicazione Ranger. Questo processo iterativo di sviluppo ha permesso di implementare miglioramenti significativi sia nell'usabilità dell'interfaccia che nelle funzionalità tecniche, rispondendo alle esigenze e alle aspettative degli utenti.

Successivamente, un secondo sondaggio è stato sottoposto agli stessi tecnici. Questo nuovo questionario includeva le stesse domande del primo per valutare se i miglioramenti ap-

portati fossero appropriati e per monitorare l'evoluzione della percezione dell'applicazione da parte degli utenti. Inoltre, sono state aggiunte domande specifiche per raccogliere feedback sui cambiamenti implementati e per identificare ulteriori aree di miglioramento.

4.3 Struttura del Questionario

Il questionario utilizzato per la valutazione dell'applicazione Ranger era composto da 17 domande, suddivise in diverse sezioni, ciascuna mirata a indagare specifici aspetti dell'utilizzo dell'applicazione. La struttura del questionario è stata progettata per raccogliere dati dettagliati su vari aspetti dell'esperienza d'uso, al fine di ottenere una comprensione completa dell'efficacia e delle aree di miglioramento dell'applicazione.

Frequenza di Utilizzo

La prima sezione del questionario era dedicata alla frequenza di utilizzo dell'applicazione Ranger. Questa sezione era fondamentale per valutare il grado di integrazione dell'applicazione nella routine lavorativa quotidiana dei tecnici. I partecipanti sono stati invitati a indicare la frequenza con cui utilizzavano l'applicazione tramite la domanda:

- "Quanto spesso utilizzi l'applicazione?"

Usabilità e Interfaccia Utente (UI)

La seconda sezione del questionario si concentrava sull'usabilità e sull'interfaccia utente dell'applicazione. L'obiettivo era comprendere quanto i tecnici trovassero l'interfaccia intuitiva e facile da usare. A tal fine, è stata posta la seguente domanda:

- "Trovi l'interfaccia dell'app intuitiva e facile da utilizzare?"

Funzionalità dell'Applicazione

Un'altra parte importante del questionario riguardava le funzionalità dell'applicazione. L'obiettivo era determinare quali funzionalità venivano utilizzate più frequentemente e quali meno spesso dai tecnici. Per raccogliere queste informazioni, sono state poste le seguenti domande:

- "Quali funzionalità dell'app utilizzi più frequentemente?"
- "Quali funzionalità dell'app utilizzi meno spesso?"

Utilità delle Funzionalità

La sezione successiva del questionario si focalizzava sull'utilità delle funzionalità offerte da Ranger. Ai tecnici è stato chiesto di identificare le principali funzionalità dell'app che trovavano più utili nel monitoraggio dei dispositivi NMS tramite la domanda:

- "Quali sono le principali funzionalità dell'app che trovi più utili nel monitoraggio dei dispositivi NMS?"

Precisione e Dettaglio delle Informazioni

La precisione e il dettaglio delle informazioni fornite dall'applicazione erano altri aspetti cruciali valutati tramite il questionario. I tecnici hanno espresso la loro opinione sulla precisione e sul livello di dettaglio delle informazioni disponibili nell'app tramite la domanda:

- "Trovi che l'app fornisca informazioni dettagliate e precise sullo stato di monitoraggio dei dispositivi NMS? Se no, quali informazioni trovi mancanti o poco chiare?"

Difficoltà Riscontrate

Il questionario includeva anche una sezione dedicata alle difficoltà riscontrate. I tecnici sono stati invitati a segnalare eventuali problemi tecnici o di usabilità incontrati durante l'uso dell'applicazione tramite la domanda:

- "Hai incontrato difficoltà nell'utilizzo dell'app? Se sí, quali?"

Punti di Forza e Aree di Miglioramento

Le domande sul punto di forza principale dell'applicazione e sulle aree di miglioramento hanno permesso ai tecnici di esprimere la loro opinione sugli aspetti più positivi dell'app e su quelli che potrebbero essere migliorati tramite le domande:

- "Cosa pensi sia il punto di forza principale dell'app?"
- "Cosa pensi potrebbe essere migliorato nell'app?"

Soddisfazione Complessiva e Motivazione all'Uso

La soddisfazione complessiva degli utenti e la loro motivazione all'uso dell'app sono state valutate tramite domande specifiche, come:

- "Quanto sei soddisfatto dell'esperienza complessiva con l'app?"
- "Trovi che l'app abbia soddisfatto le tue aspettative?"
- "Ti senti motivato a utilizzare l'app per gestire al meglio il tuo lavoro?"

Problemi Tecnici e Suggerimenti per Nuove Funzionalità

Il questionario includeva anche domande sui problemi tecnici riscontrati e sui suggerimenti per nuove funzionalità, come:

- "Quali altri servizi o funzionalità vorresti vedere aggiunti all'app?"

Preferenze tra Versione Mobile e Web

Infine, i tecnici sono stati invitati a esprimere le loro preferenze tra la versione mobile e quella web dell'applicazione tramite la domanda:

- "Preferisci la versione mobile o quella web di Ranger? Perché?"

Durata dell'Uso per Sessione

Questa sezione del questionario mirava a capire quanto tempo i tecnici trascorrevano mediamente utilizzando l'applicazione in una singola sessione. Queste informazioni erano fondamentali per valutare l'efficienza dell'applicazione e l'impatto sul flusso di lavoro quotidiano. La domanda posta era:

- "Quanto tempo trascorri mediamente utilizzando l'app in una sessione?"

Commenti e Suggerimenti

La sezione finale del questionario offriva ai partecipanti l'opportunità di condividere ulteriori osservazioni, esperienze e suggerimenti riguardo all'uso dell'applicazione. Questo spazio libero era cruciale per raccogliere feedback dettagliati e identificare aree di miglioramento non coperte dalle domande precedenti. La domanda posta era:

- "C'è qualcos'altro che vorresti condividere riguardo alla tua esperienza con l'app?"

Questa struttura ha permesso di ottenere una valutazione dettagliata dell'applicazione Ranger da parte dei tecnici, evidenziando punti di forza e aree di miglioramento fondamentali per l'evoluzione dell'applicazione.

4.4 Risultati della Valutazione

La valutazione condotta tra i tecnici specializzati ha fornito importanti dati quantitativi e qualitativi sull'esperienza con l'applicazione Ranger. I risultati evidenziano sia punti di forza significativi che criticità da affrontare. In generale l'applicazione è stata valutata positivamente da 92% dei tecnici che l'hanno testata.

4.4.1 Punti di Forza dell'Applicazione

Gli utenti hanno apprezzato particolarmente l'interfaccia intuitiva e facile da usare di Ranger, che ha reso l'interazione con l'applicazione efficiente e accessibile anche per utenti meno esperti. Inoltre, è stato molto ben accolto il fatto che l'applicazione abbia implementato una mappa dinamica interattiva, che permette agli utenti di interagire direttamente con la mappa per visualizzare informazioni dettagliate e gestire efficacemente i siti e i dispositivi monitorati.

4.4.2 Miglioramento sulla Mappa

Durante il sondaggio, è emerso che il 70% dei tecnici aveva il desiderio di migliorare l'esperienza sulla mappa dell'applicazione. In particolare, è stata espressa la necessità di poter cliccare direttamente su un sito nella mappa per accedere alla sezione "Maps status". Questa

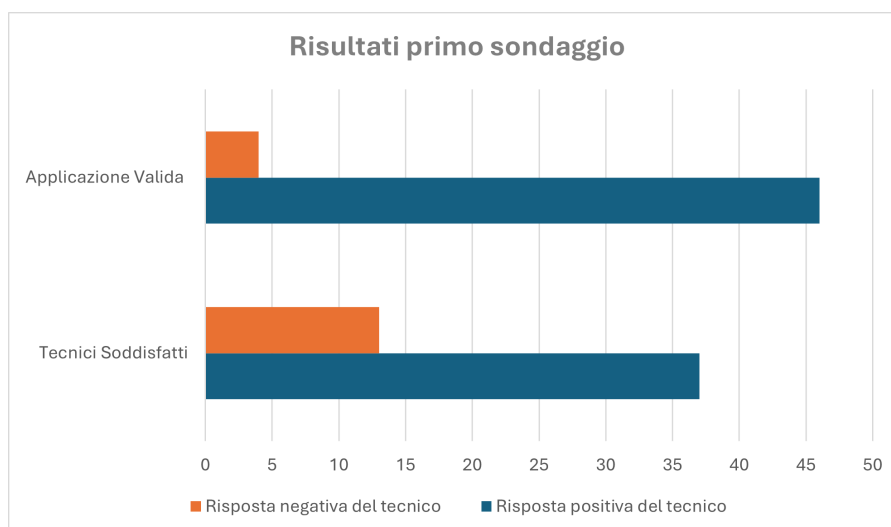


Figura 4.1: Grafico riassuntivo del primo sondaggio

funzionalità consentirebbe ai tecnici di visualizzare in modo chiaro i problemi specifici del sito e lo stato attuale dei vari dispositivi collegati. Inoltre, è stato rilevato un bisogno urgente di poter salvare i siti preferiti per semplificarne il ritrovamento e la gestione futura, migliorando così l'efficienza complessiva del monitoraggio dei siti.

4.4.3 Miglioramento della Map Status

Dal 68% dei tecnici è emersa anche la necessità di migliorare la funzionalità di Map Status dell'applicazione. Attualmente, questa sezione fornisce solo il nome del sito e l'errore correlato, evidenziato da un codice colore. I tecnici hanno espresso il desiderio di poter cliccare su ciascun sito per aprire una tendina espandibile contenente informazioni dettagliate, comprese le specifiche dei dispositivi affiliati e lo stato dettagliato. Questo aggiornamento consentirebbe una visione più completa e dettagliata dei problemi dei siti e dei dispositivi connessi, facilitando notevolmente il processo di monitoraggio e risoluzione dei problemi.

4.4.4 Problematica dell'Area Specifica

Un problema comune riportato dal 44% dei tecnici riguarda la necessità di dover continuamente ritrovare l'area specifica con i siti sulla mappa. Questa difficoltà comporta un'inevita-

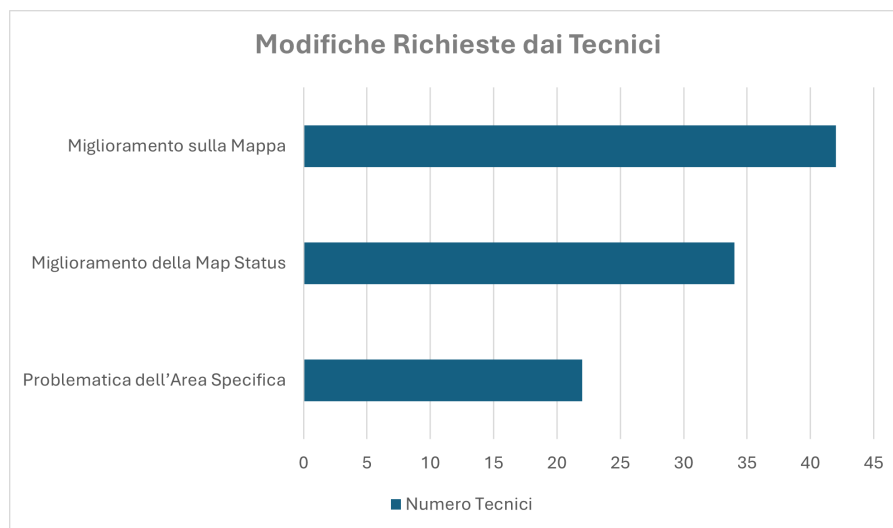


Figura 4.2: Grafico delle modifiche richieste dai tecnici

bile perdita di tempo, poiché i tecnici devono frequentemente ingrandire o selezionare nuovamente l'area desiderata. Questo feedback sottolinea l'importanza di migliorare l'esperienza utente dell'applicazione, rendendo più intuitiva e efficiente la gestione e la visualizzazione degli elementi sulla mappa durante l'uso quotidiano.

4.4.5 Aggiornamenti e Miglioramenti dell'Applicazione

Grazie ai risultati emersi dalla valutazione, l'applicazione Ranger è stata oggetto di significativi aggiornamenti e modifiche. Le richieste e i suggerimenti dei tecnici hanno guidato lo sviluppo di nuove funzionalità e miglioramenti dell'interfaccia utente, mirati a migliorare ulteriormente l'usabilità e l'efficacia dell'applicazione nell'ambiente operativo reale. Nei paragrafi successivi, esploreremo nel dettaglio le modifiche apportate e come queste abbiano risposto alle esigenze degli utenti.

4.5 Miglioramenti

Per migliorare l'applicazione Ranger in risposta ai risultati del sondaggio condotto tra i tecnici specializzati, ho concentrato la mia attenzione su diversi aspetti fondamentali per ottimizzare l'esperienza degli utenti. Uno degli aspetti principali su cui mi sono focalizzata è stata

la visualizzazione dettagliata e accessibile delle informazioni sugli errori dei siti nella sezione MapStatus. Dalle risposte raccolte nel sondaggio é emerso chiaramente che il 70% dei tecnici avvertivano la necessità di una modalità piú intuitiva per individuare e gestire gli errori specifici dei siti. Questo feedback é stato fondamentale nel guidare le modifiche apportate: ora, quando un tecnico seleziona un sito sulla mappa, si apre una tendina espandibile che fornisce dettagli precisi sul sito stesso, includendo il numero e la tipologia degli errori rilevati. Questo aggiornamento mira a fornire agli utenti una panoramica immediata e dettagliata delle problematiche, consentendo loro di intervenire prontamente per risolverle. Inoltre, ho introdotto due nuovi bottoni per arricchire ulteriormente l'esperienza dell'utente durante l'utilizzo di Ranger. Il primo bottone, denominato "Proprietá del sito", offre agli utenti la possibilità di accedere a informazioni cruciali come il nome del sito, l'ID univoco, la posizione geografica e lo stato attuale del sito selezionato. Questo permette ai tecnici di avere una visione completa del contesto operativo del sito, fornendo loro tutte le informazioni necessarie per comprendere il contesto in cui si verifica l'errore. Il secondo bottone, denominato "Elenco device sito", presenta un dettagliato elenco di tutti i dispositivi associati al sito selezionato, evidenziando lo stato di ciascun dispositivo con colori distinti (verde, giallo e rosso) per indicare se sono attivi o se presentano problematiche. Questa funzionalità é stata implementata per aiutare i tecnici a identificare rapidamente quali dispositivi sono coinvolti negli errori segnalati, consentendo loro di concentrare i loro sforzi di risoluzione esattamente dov'è necessario. Un'altra problematica significativa riscontrata durante la valutazione dell'applicazione é stata la necessità di migliorare l'accesso alla MapStatus direttamente dalla mappa. I tecnici hanno espressamente richiesto la possibilità di cliccare su un sito specifico nella mappa e di poter visualizzare istantaneamente lo stato dettagliato del sito, compresi tutti gli errori e le problematiche segnalate. Questo miglioramento é stato implementato per consentire una diagnosi piú rapida e precisa delle situazioni critiche, migliorando cosí la tempestività delle azioni correttive. Inoltre, é emersa l'importanza di poter salvare i siti di interesse come preferiti. Questa funzionalità é stata introdotta per consentire ai tecnici di accedere rapidamente ai siti piú rilevanti per il loro lavoro quotidiano senza dover cercare ogni volta o ripetere le operazioni di selezione. Salvare un sito come preferito non solo semplifica l'accesso immediato, ma ottimizza anche la gestione complessiva delle risorse e delle attività di monitoraggio, migliorando l'efficienza operativa complessiva dell'applicazione Ranger. Per rispondere in modo efficace a queste

esigenze specifiche dei tecnici, ho introdotto una serie di nuove funzionalità direttamente sulla mappa. Quando un tecnico seleziona un sito specifico nella mappa, si apre una tendina che presenta due opzioni fondamentali: il primo pulsante, denominato "Condividi stato", consente di condividere istantaneamente lo stato attuale del sito con altri utenti, facilitando la collaborazione e mantenendo intatta la funzionalità già apprezzata nella versione precedente dell'applicazione. Il secondo pulsante, "Apri MapStatus", reindirizza direttamente alla sezione MapStatus del sito selezionato. Questa sezione fornisce dettagli precisi e accurati sugli errori rilevati sul sito, permettendo ai tecnici di analizzare con precisione e tempestività le problematiche individuate. In aggiunta a questi due pulsanti, è stata introdotta una nuova icona a forma di stella nella tendina espandibile. Questa icona rappresenta una funzionalità fondamentale per migliorare l'efficienza dei tecnici. Cliccando sulla stella, i tecnici hanno la possibilità di salvare il sito specifico nella lista dei loro siti preferiti. Questa funzione offre un notevole vantaggio, poiché consente agli utenti di accedere rapidamente e senza sforzo ai siti di maggiore interesse o a quelli che richiedono un monitoraggio continuo. Di conseguenza, il recupero e la gestione dei siti di interesse diventano notevolmente più semplici e veloci, riducendo il tempo necessario per navigare tra i vari siti sulla mappa. Inoltre, questa funzionalità personalizzata permette ai tecnici di organizzare meglio il loro lavoro, migliorando la produttività e garantendo un intervento più rapido ed efficace sui siti critici. La possibilità di creare una lista di siti preferiti rende l'applicazione Ranger più user-friendly e adattabile alle esigenze specifiche di ciascun utente, ottimizzando l'intero processo di monitoraggio e manutenzione. L'elenco dei siti preferiti è stato integrato in modo completo e intuitivo nell'area account dell'utente. Oltre ai dati anagrafici standard, che rimangono invariati rispetto alla versione precedente, l'area utente è stata arricchita con due nuove e potenti funzionalità. La prima è il pulsante "I miei siti", che, se selezionato, apre una tendina dettagliata che contiene l'elenco completo di tutti i siti che il tecnico ha scelto come preferiti. Questa funzionalità rappresenta un significativo miglioramento rispetto alla versione precedente, poiché consente una visualizzazione immediata e organizzata dei siti di maggiore importanza per l'utente. Grazie a questa tendina, i tecnici possono accedere rapidamente ai siti preferiti, monitorare gli errori in modo più efficiente e intervenire tempestivamente, riducendo il tempo speso nella ricerca dei siti sulla mappa. La seconda nuova funzionalità è il pulsante "La mia Area", che aggiunge un ulteriore livello di personalizzazione e praticità all'esperienza utente. Quan-

do un tecnico seleziona questo pulsante, viene automaticamente reindirizzato alla sua area di lavoro specifica sulla mappa, senza dover navigare manualmente per localizzarla. Questa funzione non solo semplifica l'accesso alle aree di interesse, ma migliora significativamente l'efficienza operativa, permettendo ai tecnici di concentrarsi immediatamente sui siti che necessitano di attenzione. La combinazione di queste due nuove funzionalità offre un'esperienza utente ottimizzata, che facilita la gestione degli errori e l'organizzazione del lavoro quotidiano. Questi miglioramenti sono stati pensati e sviluppati con l'obiettivo di rendere l'applicazione Ranger più intuitiva, funzionale ed efficiente nel supportare le operazioni quotidiane dei tecnici. Oltre a facilitare la gestione delle emergenze e la risoluzione dei problemi, questi aggiornamenti mirano a migliorare l'efficienza complessiva di Ranger nell'ambiente operativo reale, riducendo al contempo il tempo necessario per risolvere le criticità segnalate. In sintesi, l'implementazione di queste caratteristiche risponde direttamente alle esigenze emerse dal sondaggio, migliorando significativamente l'esperienza degli utenti e l'efficacia dell'applicazione nell'affrontare le sfide quotidiane del monitoraggio e della gestione dei siti.

4.6 Rivalutazione e conclusioni

Dopo aver implementato queste significative modifiche all'applicazione Ranger, i tecnici sono stati nuovamente invitati a rivalutare l'applicazione. È stato chiesto loro di fornire un feedback specifico riguardo alle nuove funzionalità e alle modifiche introdotte, per determinare se queste migliorie sono state apprezzate e se hanno effettivamente migliorato l'esperienza utente. Durante la fase di rivalutazione, i tecnici hanno avuto l'opportunità di testare in modo approfondito le nuove caratteristiche e di esprimere le loro opinioni su come queste abbiano influenzato il loro lavoro quotidiano. Questo processo di rivalutazione ha coinvolto una serie di test pratici e simulazioni di scenari reali per garantire che le nuove funzionalità fossero messe alla prova in condizioni operative realistiche. I tecnici hanno potuto verificare se le modifiche apportate rispondevano effettivamente alle loro esigenze e se facilitavano il loro lavoro, migliorando al contempo l'efficienza e la produttività. È emerso chiaramente che le modifiche apportate sono state accolte con entusiasmo. Il 90% dei tecnici ha apprezzato notevolmente le nuove funzionalità, rilevando un significativo miglioramento nella facilità d'uso e nell'efficienza operativa dell'applicazione. Le nuove opzioni di interazione con la mappa, la

possibilità di salvare e accedere rapidamente ai siti preferiti, e la sezione "La mia Area" sono state particolarmente elogiate. Queste funzionalità hanno reso l'applicazione più intuitiva e user-friendly, riducendo il tempo necessario per navigare e gestire i siti, e aumentando la precisione con cui i tecnici possono identificare e risolvere i problemi. Come detto precedentemente, una delle funzionalità che ha ricevuto un consenso particolarmente positivo è stata l'introduzione della possibilità di salvare i siti preferiti e accedervi rapidamente tramite la nuova sezione "I miei siti". In particolare la possibilità di salvare i siti preferiti ha ricevuto il 96% d'apprezzamento da parte dei tecnici, mentre la lista "I miei siti" è stata apprezzata dal 82% dei professionisti. I tecnici hanno sottolineato come questa funzione abbia semplificato notevolmente il processo di monitoraggio e gestione dei siti di maggiore importanza, permettendo loro di risparmiare tempo prezioso e di concentrarsi maggiormente sulla risoluzione dei problemi. La capacità di visualizzare immediatamente un elenco dei siti preferiti ha reso l'applicazione più intuitiva, migliorando l'esperienza utente complessiva. Tuttavia, è stato suggerito che l'usabilità dell'applicazione potrebbe essere ulteriormente migliorata rendendo la funzione "I miei siti" accessibile direttamente dall'area di controllo dei siti, ovvero dalla MapStatus. Attualmente, questa funzionalità è collocata nell'area personale dell'utente, il che richiede un passaggio aggiuntivo per accedere ai siti preferiti. Spostare questa funzione nella MapStatus permetterebbe ai tecnici di avere un accesso immediato ai loro siti preferiti senza dover navigare attraverso il menu dell'account utente. Questo cambiamento ridurrebbe significativamente i tempi di ricerca e selezione dei siti di interesse, rendendo il processo molto più efficiente e intuitivo. I tecnici potrebbero quindi beneficiare di una maggiore comodità, potendo richiamare rapidamente i loro siti preferiti direttamente dalla mappa di controllo, migliorando così la loro capacità di monitorare e gestire i siti in maniera più fluida e dinamica. Inoltre, questa modifica risponderebbe meglio alle esigenze di lavoro sul campo, dove rapidità e accessibilità sono essenziali per affrontare prontamente le emergenze e risolvere i problemi in modo tempestivo.

Un'altra funzionalità che ha ricevuto ampi consensi è stata "La mia Area", con l'86% di gradimento. Questa opzione permette ai tecnici di essere reindirizzati automaticamente alla loro area di lavoro specifica sulla mappa, senza dover navigare manualmente per individuarla ogni volta. Questa innovazione è stata considerata estremamente utile e ha avuto un impatto significativo sul flusso di lavoro quotidiano dei tecnici. Prima dell'introduzione di questa

funzionalità, i tecnici dovevano spendere tempo e sforzi preziosi per localizzare manualmente la loro area di interesse sulla mappa. Questo processo non solo era inefficiente ma aumentava anche la frustrazione, soprattutto in situazioni in cui la rapidità di intervento era essenziale. Con "La mia Area", questo problema è stato efficacemente risolto. Ora, i tecnici possono accedere direttamente alla loro area di lavoro con un solo clic, migliorando notevolmente l'efficienza operativa. Questa funzionalità ha ridotto significativamente il tempo necessario per la navigazione e ha permesso ai tecnici di concentrarsi maggiormente sulle loro attività principali, come il monitoraggio e la risoluzione dei problemi. I feedback ricevuti dai tecnici hanno evidenziato che questa novità ha semplificato notevolmente l'interazione con la mappa e ha migliorato l'esperienza utente complessiva. La possibilità di accedere rapidamente alla propria area di lavoro ha reso l'applicazione più intuitiva e ha incrementato la soddisfazione degli utenti, confermando l'efficacia di questa modifica.

Anche la nuova funzione di diagnostica, che consente l'accesso diretto alla MapStatus, è stata apprezzata dal 76% dei tecnici per la sua utilità nel fornire una visione chiara e immediata delle problematiche. Ora, i tecnici possono semplicemente cliccare sull'icona del sito di interesse sulla mappa e successivamente sul pulsante "Apri MapStatus", il quale apre direttamente la finestra MapStatus con informazioni dettagliate sugli errori del sito selezionato. Questa modifica ha semplificato notevolmente il processo di identificazione dei problemi. La capacità di accedere immediatamente ai dettagli specifici degli errori del sito ha migliorato l'esperienza utente e ha reso l'applicazione ancora più funzionale e pratica per il lavoro quotidiano. Questa funzionalità avanzata non solo ha migliorato l'efficienza operativa, ma ha anche aumentato la soddisfazione degli utenti, poiché la diagnostica più rapida e dettagliata ha reso il loro lavoro meno stressante e più produttivo. La capacità di visualizzare immediatamente le informazioni critiche direttamente dalla mappa ha rappresentato un notevole passo avanti nell'usabilità dell'applicazione, trasformando un processo che prima poteva essere complesso e dispendioso in termini di tempo in una procedura semplice e immediata.

In conclusione, il feedback positivo ricevuto dai tecnici dopo l'implementazione delle nuove funzionalità e modifiche all'applicazione Ranger dimostra chiaramente che le migliorie apportate hanno avuto un impatto significativo e positivo sull'esperienza utente. Questi cambiamenti hanno reso l'applicazione più efficiente, intuitiva e adatta alle esigenze operative dei tecnici, confermando l'importanza di ascoltare attentamente i feedback degli utenti

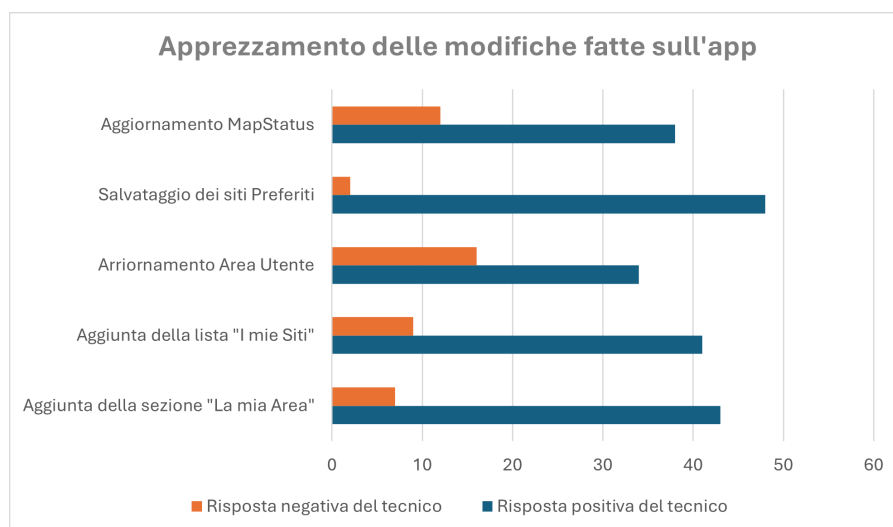


Figura 4.3: Risultati del secondo sondaggio

per il continuo miglioramento e successo dell'applicazione. L'importanza di una buona UI (interfaccia utente) e UX (esperienza utente) nella realizzazione e aggiornamento di un'applicazione mobile non può essere sottolineata abbastanza. Una UI ben progettata non solo rende l'applicazione esteticamente gradevole, ma facilita anche la navigazione e l'utilizzo delle sue funzionalità, riducendo la curva di apprendimento per gli utenti. In questo contesto, l'aggiunta di pulsanti intuitivi e icone chiare ha contribuito notevolmente a semplificare l'interazione con l'applicazione Ranger, rendendo i processi di diagnostica e risoluzione dei problemi molto più fluidi. Parallelamente, una UX eccellente si traduce in un'applicazione che non solo soddisfa le esigenze funzionali degli utenti, ma lo fa in modo che l'interazione risulti piacevole e priva di frustrazioni. Gli aggiornamenti introdotti, come la possibilità di accedere direttamente alla MapStatus e la funzionalità "La mia Area", hanno migliorato la velocità e l'efficacia con cui i tecnici possono eseguire le loro attività quotidiane. La riduzione del tempo necessario per navigare manualmente attraverso l'applicazione ha permesso ai tecnici di concentrarsi maggiormente sulle loro mansioni principali, aumentando la produttività e la soddisfazione nel lavoro. Inoltre, la capacità di salvare e accedere rapidamente ai siti preferiti ha reso l'applicazione ancora più user-friendly. Questa funzionalità ha dimostrato quanto sia cruciale progettare con attenzione ogni aspetto dell'esperienza utente per garantire che le operazioni quotidiane dei tecnici siano il più possibile efficienti e senza intoppi. I risultati della

rivalutazione sottolineano l'efficacia del processo iterativo di sviluppo e miglioramento dell'applicazione. Questo approccio permette di affinare continuamente l'applicazione in base ai feedback reali degli utenti, assicurando che Ranger rimanga uno strumento indispensabile e sempre all'avanguardia per i tecnici nel loro lavoro quotidiano. In sintesi, l'implementazione di una UI e UX eccellenti non solo ha migliorato la funzionalità e l'efficienza dell'applicazione Ranger, ma ha anche aumentato la soddisfazione degli utenti. Questi miglioramenti dimostrano come un'attenzione costante alle esigenze e ai feedback degli utenti possa guidare lo sviluppo di soluzioni tecnologiche efficaci, confermando l'importanza di un design centrato sull'utente per il successo a lungo termine di qualsiasi applicazione mobile.

Conclusioni e Prospettive Future

A conclusione di questa tesi, il cui obiettivo è stato quello di sviluppare un'applicazione Android e iOS per l'azienda Axel Technology, dedicata alla gestione dei dispositivi NMS per il monitoraggio dei siti delle antenne radiofoniche sparse su tutto il territorio italiano, è possibile affermare che, rispetto ai test effettuati sulle funzionalità implementate e alle migliorie apportate in seguito al primo sondaggio condotto tra i tecnici specializzati, l'applicazione rappresenta un notevole supporto tecnico rispetto alle soluzioni attualmente disponibili sul mercato, poiché queste ultime risultano essere molto generiche. Questo progetto ha dimostrato come l'integrazione di nuove tecnologie e un'attenzione particolare alle esigenze specifiche dell'azienda possano portare allo sviluppo di un prodotto altamente efficiente.

I principali punti di forza dell'applicazione includono:

Verticalità e Personalizzazione

L'applicazione è progettata specificamente per le esigenze di Axel Technology, offrendo funzionalità su misura che non si trovano nelle soluzioni generiche. Questo approccio permette di soddisfare requisiti operativi specifici, migliorando l'efficacia e l'efficienza del lavoro dei tecnici. La possibilità di personalizzare l'applicazione in base alle peculiarità operative di Axel Technology consente un adattamento perfetto alle diverse situazioni lavorative. Questo vantaggio si traduce in una maggiore flessibilità nell'affrontare le sfide quotidiane e nell'implementare miglioramenti continui, garantendo un vantaggio competitivo nel settore.

Interfaccia Intuitiva

La user interface è progettata per essere facile da usare, riducendo il tempo necessario per la formazione dei tecnici e minimizzando gli errori operativi. L'accessibilità immediata alle

funzioni chiave e una navigazione fluida aumentano la produttività e la soddisfazione degli utenti. Un'interfaccia intuitiva riduce significativamente la curva di apprendimento, permettendo ai nuovi utenti di diventare operativi in tempi brevi. Inoltre, la semplicità d'uso dell'interfaccia minimizza il rischio di errori umani, aumentando l'accuratezza delle operazioni e garantendo un utilizzo più efficiente delle risorse.

Supporto Multi-Dispositivo

Implementare il supporto per diversi dispositivi mobili assicura che l'applicazione sia accessibile a tutti i membri del team, indipendentemente dal dispositivo utilizzato. Questo non solo aumenta la flessibilità operativa, consentendo ai tecnici di accedere alle informazioni cruciali ovunque si trovino, ma anche garantisce una maggiore adozione e utilizzo dell'applicazione su scala aziendale.

Centralizzazione delle Operazioni

Integrare vari sistemi e funzionalità in un'unica piattaforma consente una gestione più efficiente e coordinata delle attività di manutenzione e monitoraggio. Questo riduce la necessità di utilizzare più applicazioni e migliora la coerenza dei dati. La centralizzazione delle operazioni favorisce una visione d'insieme delle attività e facilita la comunicazione e il coordinamento tra i diversi reparti. Inoltre, permette una gestione centralizzata degli aggiornamenti e delle modifiche al sistema, semplificando la manutenzione e riducendo i tempi di inattività.

Automazione delle Attività

La possibilità di automatizzare compiti ripetitivi e di monitorare continuamente le condizioni della rete libera i tecnici per compiti più complessi e strategici. L'automazione riduce gli errori umani e garantisce una risposta tempestiva alle problematiche. Implementare processi automatizzati consente di ottimizzare l'allocazione delle risorse e di migliorare l'efficienza operativa. Gli algoritmi di automazione possono essere programmati per identificare anomalie, eseguire diagnosi preliminari e persino attuare interventi correttivi senza intervento umano, migliorando significativamente la continuità del servizio.

Notifiche in Tempo Reale

L'invio immediato di avvisi e notifiche ai tecnici in caso di anomalie o guasti permette interventi tempestivi, migliorando la continuità del servizio. Questo sistema di allerta proattiva è cruciale per mantenere alte le prestazioni della rete. Le notifiche in tempo reale assicurano che i tecnici siano sempre informati sulle condizioni della rete e possano intervenire prontamente per risolvere i problemi. Questo non solo riduce il tempo di inattività, ma aumenta anche la fiducia degli utenti finali nella stabilità e nell'affidabilità del servizio offerto.

Diagnostica Remota

Strumenti di diagnosi a distanza aiutano i tecnici a identificare e risolvere problemi senza la necessità di essere fisicamente presenti sul sito. Questo non solo riduce i tempi di intervento, ma anche i costi associati agli spostamenti. La diagnostica remota consente un monitoraggio continuo e dettagliato delle condizioni della rete, permettendo ai tecnici di prendere decisioni informate basate su dati accurati. Questa capacità di diagnosi e intervento a distanza è particolarmente utile in contesti geografici estesi o difficilmente accessibili, migliorando l'efficienza operativa e riducendo i costi operativi complessivi.

4.7 Prospettive di Miglioramento

Nonostante i numerosi punti di forza, emersi dai recenti sondaggi effettuati, è evidente che l'applicazione presenta ancora margini di miglioramento significativi. Questo feedback ha indicato diverse aree in cui l'applicazione potrebbe essere ulteriormente sviluppata per ottimizzare le sue funzionalità e prestazioni. Di seguito sono descritte in dettaglio le principali aree di miglioramento:

1. **Scalabilità:** È essenziale ampliare la capacità dell'applicazione per gestire un numero crescente di dispositivi e reti. Questo garantirà che l'applicazione possa supportare l'espansione futura dell'azienda senza compromettere l'efficienza operativa. Migliorare la scalabilità consentirà di mantenere prestazioni elevate anche con un aumento significativo del carico di lavoro e della complessità dell'infrastruttura.

2. **Integrazione con Tecnologie Avanzate:** Incorporare l'uso di intelligenza artificiale (AI) e machine learning (ML) rappresenta un passo cruciale per migliorare ulteriormente la prevenzione dei guasti e la gestione proattiva della rete. Le tecnologie avanzate possono analizzare grandi quantità di dati in tempo reale, identificando pattern e anomalie che potrebbero indicare potenziali problemi prima che si verifichino. Ciò aumenta la resilienza della rete e riduce i tempi di inattività.
3. **Interoperabilità:** Assicurare che l'applicazione possa integrarsi senza problemi con altri sistemi di gestione e piattaforme utilizzate da diverse aziende del settore delle telecomunicazioni è fondamentale per migliorare l'efficienza complessiva delle operazioni. Una migliore interoperabilità facilita la collaborazione tra team e l'interscambio di dati, permettendo una gestione più fluida e coordinata delle attività di monitoraggio e manutenzione.
4. **Usabilità:** Continuare a migliorare l'interfaccia utente basandosi sul feedback degli utenti è essenziale per mantenere elevati standard di usabilità e facilità d'uso dell'applicazione. Un'interfaccia intuitiva e ben progettata non solo riduce il tempo necessario per la formazione dei tecnici, ma anche il rischio di errori operativi, migliorando complessivamente l'efficacia dell'applicazione nell'ambiente operativo.
5. **Analisi Avanzata dei Dati:** Introdurre funzionalità avanzate di analisi dei dati è fondamentale per ottenere insights dettagliati e utili per migliorare la gestione della rete. L'analisi avanzata dei dati permette di identificare tendenze, pattern e correlazioni nascoste nei dati operativi, supportando una migliore strategia operativa, decisionale e pianificazione futura delle risorse.

Ciascuna di queste aree di miglioramento non solo potenzia le capacità dell'applicazione nell'affrontare sfide future, ma anche contribuisce a consolidare la posizione di Axel Technology come leader innovativo nel settore delle telecomunicazioni, fornendo soluzioni robuste e all'avanguardia per la gestione dei dispositivi NMS e la supervisione delle reti radiofoniche in Italia e oltre.

4.8 Conclusioni finali

In conclusione, l'applicazione sviluppata per Axel Technology rappresenta un importante supporto nel campo della gestione dei dispositivi NMS, offrendo soluzioni su misura che rispondono alle specifiche esigenze dei tecnici. I numerosi miglioramenti apportati, basati sui feedback degli utenti, hanno significativamente migliorato l'efficienza operativa, l'usabilità e la funzionalità dell'applicazione. Tuttavia, per mantenere il vantaggio competitivo e continuare a soddisfare le crescenti esigenze del settore, è fondamentale continuare a investire nello sviluppo e nel miglioramento dell'applicazione. Implementando le aree di miglioramento suggerite, l'applicazione potrà non solo consolidare la sua posizione sul mercato, ma anche contribuire significativamente all'efficienza operativa e alla qualità del servizio offerto da Axel Technology.

Il futuro sviluppo dell'applicazione deve anche considerare l'implementazione di nuove funzionalità avanzate che possano anticipare le esigenze emergenti dei clienti e del mercato. Questo approccio proattivo garantirà che Axel Technology rimanga all'avanguardia nel fornire soluzioni innovative e affidabili, mantenendo la fiducia e la soddisfazione dei suoi utilizzatori.

La costante attenzione alla UI e alla UX è cruciale per garantire che l'applicazione rimanga intuitiva e user-friendly, favorendo un'interazione fluida e produttiva da parte dei tecnici. L'adozione di un approccio iterativo e centrato sull'utente nel processo di sviluppo continuerà a essere la chiave per il successo a lungo termine dell'applicazione, assicurando che essa possa evolversi e adattarsi alle nuove sfide e opportunità del settore delle telecomunicazioni.

Bibliografia

- [1] Stallings, W. (2013). *Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud*. Addison-Wesley Professional.
Questo libro offre una panoramica delle tecnologie di rete moderne, inclusi il cloud computing, l'IoT e le reti definite dal software (SDN).
- [2] Case, J., Fedor, M., Schoffstall, M., & Davin, J. (1990). *A Simple Network Management Protocol (SNMP)*. RFC 1157.
Documento fondamentale che descrive il protocollo SNMP, una pietra miliare nella storia della gestione delle reti.
- [3] Bourne, S., & Rutherglen, A. (2016). *Network Management: Concepts and Practice, A Hands-On Approach*. Addison-Wesley.
Questo testo offre un approccio pratico alla gestione delle reti, con una panoramica storica e approfondimenti sulle tecnologie moderne.
- [4] Tanenbaum, A. S., & Wetherall, D. J. (2010). *Computer Networks*. Pearson.
Un classico nel campo delle reti di computer, con sezioni dedicate alla gestione delle reti e alla loro evoluzione storica.
- [5] Cisco Systems. *Cisco Prime Infrastructure*.
Documentazione ufficiale e white papers di Cisco Systems su Cisco Prime Infrastructure, una delle principali soluzioni di gestione delle reti.
- [6] SolarWinds. *Network Performance Monitor (NPM)*.
Documentazione ufficiale e risorse di SolarWinds su NPM, uno dei principali strumenti di monitoraggio delle prestazioni di rete.

- [7] Paessler AG. *PRTG Network Monitor*.
Risorse e documentazione ufficiale su PRTG Network Monitor, una soluzione di monitoraggio della rete versatile e popolare.
- [8] Nagios Enterprises. *Nagios XI*.
Documentazione ufficiale e risorse su Nagios XI, una piattaforma di monitoraggio IT completa e flessibile.
- [9] Garcia, M. (2018). *Network Management Fundamentals*. Cisco Press.
Una guida completa che copre i fondamenti della gestione delle reti, con un occhio di riguardo alle tecnologie moderne e alle best practice.
- [10] React Native. *React Native Documentation*.
<https://reactnative.dev/docs/getting-started>.
Documentazione ufficiale di React Native che fornisce dettagli completi sulle caratteristiche, l'architettura e l'implementazione di React Native.
- [11] Expo Documentation.
<https://docs.expo.dev/>.
Documentazione ufficiale di Expo che offre una panoramica delle funzionalità e degli strumenti forniti da Expo per lo sviluppo di applicazioni mobile con React Native.
- [12] Tillmann, M. (2019). *React Native for Mobile Development*. Apress.
Un libro che esplora in dettaglio l'uso di React Native per lo sviluppo di applicazioni mobili, coprendo best practice e casi d'uso reali.
- [13] Brown, J. (2020). *Learning React Native: Building Native Mobile Apps with JavaScript*. O'Reilly Media.
Questo libro offre una guida pratica allo sviluppo di applicazioni mobili utilizzando React Native, con esempi dettagliati e spiegazioni passo passo.
- [14] Expo Team. (2020). *Expo: A Complete Guide to Developing with React Native and Expo*. Expo Documentation.
Una guida completa per lo sviluppo con Expo e React Native, che copre configurazione, sviluppo, test e distribuzione delle applicazioni.

[15] Parashar, A. (2024). *Advanced React Native Concepts*. Packt Publishing.

Un libro che esplora concetti avanzati di React Native, inclusa l'architettura a thread singolo e il JS Bridge.

[16] Microsoft. (2024). *React Native Performance Optimization*. Microsoft Docs.

<https://docs.microsoft.com/en-us/react-native/performance>.

Documentazione che offre linee guida e best practice per ottimizzare le prestazioni delle applicazioni React Native.

[17] Osterlund, K. (2024). *Mastering React Native*. Wiley.

Un libro completo che copre vari aspetti dello sviluppo con React Native, inclusi il rendering dei componenti e la gestione del JS Bridge.

[18] Microsoft. (2011). *MVC Design Pattern*.

<https://learn.microsoft.com/it-it/archive/msdn-magazine/2011/december/mvpvm-design-pattern-the-model-view-p>

Un articolo che descrive il pattern di design MVC.

[19] Microsoft. (2009). *Model-View-ViewModel Design Pattern*.

<https://learn.microsoft.com/it-it/archive/msdn-magazine/2009/february/patterns-wpf-apps-with-the-model-view>

Un articolo che descrive il pattern di design MVVM per lo sviluppo di applicazioni WPF.