

UNIVERSITÀ DEGLI STUDI DI SALERNO



DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE ED ELETTRICA E
MATEMATICA APPLICATA

Corso di laurea
in
INGEGNERIA INFORMATICA

**Gymzone: progetto e sviluppo di
un'applicazione iOS per il fitness.
Implementazione delle GUI e tracciamento e
previsione delle calorie bruciate dall'utente
mediante l'utilizzo dei framework HealthKit,
Charts e CoreML.**

Relatore:
Ch. Prof. Luca Greco

Candidato:
Marco Cerino
Mat. 0612704997

ANNO ACCADEMICO 2022/2023

SOMMARIO

Introduzione	3
1 Panoramica delle tecnologie utilizzate	6
1.1 Tecnologie Hardware	7
1.1.1 Dispositivi	7
1.1.2 Componenti	8
1.2 Tecnologie Software	8
1.2.1 Sistemi operativi e sviluppo	8
1.2.2 Framework	9
2 Progettazione della soluzione	12
2.1 CBL	13
2.1.1 Engage	14
2.1.2 Investigate	15
2.1.3 Act	16
2.2 Design	16
2.2.1 Principi di design	17
2.2.2 Design Visivo	18
2.2.3 Modalità	18
2.2.4 Branding	18
2.2.5 Colore	18
2.2.6 Terminologia e Tipografia	18
2.2.7 Navigazione	19
2.2.8 Barre	19
2.2.9 Viste	20
2.3 Mockup	21
2.4 Strutture Dati	21
2.4.1 Proprietà	21
2.4.2 Tipologia	23
2.4.3 Differenza tra Struct e Classi	23
2.4.4 Modello dei dati nell'applicazione	24

3 Implementazione e descrizione delle funzionalità	25
3.1 Funzionamento View	26
3.1.1 Content View	26
3.1.2 AddWorkout View	27
3.1.3 ExercisePropertyView	27
3.1.4 AddExerciseView	27
3.1.5 Workout View	28
3.1.6 Play View	29
3.1.7 WorkoutAward View	31
3.1.8 Execution View	32
3.1.9 SngleExecution View	32
3.1.10 Profile View	33
3.1.11 Analytics View	34
3.1.12 Prediction View	41
4 Conclusioni e prospettive future	43

INTRODUZIONE

Un aspetto fondamentale per la vita di qualsiasi individuo è l'attività fisica. L'Organizzazione Mondiale della Sanità, riferendosi allo sport, spiega come la sua mancanza nella vita delle persone rappresenti una delle possibili cause di mortalità in tutto il mondo. Nella società odierna, dove ormai il lavoro richiede sempre meno movimento (un esempio è proprio lo smart working), l'attività fisica ottiene ancora più importanza, in quanto può ridurre il rischio di gravi malattie (ipertensione, obesità, cancro, cardiopatia, diabete e depressione), migliorare la salute delle ossa ed aiutare ad avere un peso sano. I vantaggi dello sport però non riguardano soltanto la salute del corpo, ma anche quella della mente. In tale ambito l'attività fisica ha una notevole rilevanza in quanto, grazie al rilascio delle endorfine, diminuisce lo stato di ansia e stress. L'International Society of Sport Psychology(ISSP) ha confermato come l' attività fisica sia in grado di apportare miglioramenti psicologici a breve e lungo periodo, come ad esempio un positivo cambiamento nella percezione di sé stessi, un incremento di energia ed entusiasmo, un aumento del piacere per l'esercizio fisico, maggiore fiducia e un maggior stato di prontezza mentale. Fare attività fisica, dunque, è di grande importanza per la propria salute e non solo, in quanto permette di ampliare la sfera sociale, entrando in contatto con altre persone che condividono interessi simili e creando nuove relazioni.

La palestra è un'attività fisica estremamente amata, soprattutto tra i giovani. Molti di loro, soprattutto coloro che si avventurano per la prima volta in questo ambiente, desiderano massimizzare l'efficacia del loro allenamento, cercando di organizzarsi al meglio e di beneficiare del supporto di un allenatore, o persino di raggiungere i loro obiettivi senza l'ausilio di un professionista. È proprio partendo da questa esigenza che nasce il progetto "GymZone".

GymZone è un'innovativa applicazione progettata per offrire una vasta gamma di funzionalità e migliorare l'esperienza dell'allenamento in palestra. L'obiettivo principale è quello di fornire un supporto personalizzato a ciascun individuo durante la fase di allenamento, garantendo un contributo significativo al raggiungimento dei propri obiettivi di fitness. Una delle caratteristiche chiave di GymZone è la sua capacità di fornire istruzioni dettagliate sulla corretta esecuzione degli esercizi. Ciò consente agli utenti di evitare infortuni e massimizzare i risultati del proprio allenamento. Inoltre, l'app offre la flessibilità di creare programmi di allenamento personalizzati in base alle esigenze e alle caratteristiche di ciascun individuo.

Oltre a ciò, GymZone offre agli utenti un modo intuitivo per monitorare e analizzare il loro consumo calorico nel tempo. Mediante un grafico interattivo, è possibile visualizzare le calorie bruciate nei giorni precedenti al giorno corrente. Questa funzione fornisce una panoramica immediata dell'andamento del consumo calorico nel breve termine, consentendo agli utenti di valutare i risultati dei loro sforzi e apportare eventuali modifiche all'allenamento o all'alimentazione. Grazie ad un modello di predizione, GymZone è in grado di effettuare previsioni sul consumo calorico futuro. Questa funzione fornisce una prospettiva utile per la pianificazione dell'allenamento e dell'alimentazione, consentendo agli utenti di adattare le proprie strategie per raggiungere i propri obiettivi di fitness.

Un'altra funzionalità di grande interesse è la possibilità per gli utenti di tenere traccia dei propri progressi attraverso GymZone. Grazie a questa app, è possibile inserire e monitorare il proprio peso corporeo e visualizzarlo tramite un grafico intuitivo.

La funzione di monitoraggio del peso corporeo consente agli utenti di registrare regolarmente il proprio peso e visualizzare i dati in modo chiaro e intuitivo attraverso un grafico. È importante sottolineare che, per quanto riguarda le misurazioni delle parti del corpo, GymZone permette di inserire tali dati, ma non offre una rappresentazione grafica specifica per le misurazioni.

In definitiva, GymZone si propone di soddisfare le esigenze di tutti coloro che desiderano migliorare il proprio allenamento in palestra. Offrendo un'ampia gamma di funzioni, dall'istruzione degli esercizi alla tracciabilità dei progressi personali, l'app mira a rendere l'esperienza in palestra più gratificante ed efficace per tutti gli utenti.

Durante l'esperienza presso la "Apple Foundation", il mio gruppo e io abbiamo creato il corpo dell'applicazione. Successivamente, abbiamo sfruttato le ore di tirocinio rimanenti per concentrarci su aree specifiche. In particolare, mi sono dedicato alla risoluzione di vari bug presenti nel codice preesistente, alla ristrutturazione del codice stesso dell'applicazione e al completamento del corretto funzionamento del timer. Inoltre, ho aggiunto la vista "Analytics". Questa funzionalità permette di tenere traccia delle calorie che l'utente potrebbe bruciare in futuro, consentendo una pianificazione più precisa e mirata degli obiettivi di fitness. Inoltre, la vista "Analytics" fornisce anche informazioni sulle calorie bruciate nel passato (descritto precedentemente).

Al fine di arricchire ulteriormente l'esperienza dell'utente, ho introdotto la "Review Award", che consente all'utente di visualizzare un elenco dettagliato degli esercizi eseguiti durante l'allenamento, compresi i relativi pesi utilizzati. Questo aggiornamento è stato particolarmente significativo poiché offre all'utente un modo intuitivo per rivedere i propri progressi nel tempo.

Nel capitolo 1 della tesi, si analizzeranno in dettaglio le diverse tecnologie hardware e software che sono state impiegate nel contesto dello sviluppo dell'applicazione. Saranno esaminate le caratteristiche e le specifiche di tali tecnologie, evidenziando la loro rilevanza nell'ambito del progetto. Inoltre, verranno presentate le motivazioni alla base della scelta di tali tecnologie, considerando fattori come l'efficienza, la scalabilità e l'integrazione con l'ambiente Apple.

Nel capitolo 2, sarà illustrata la progettazione della soluzione proposta, ponendo particolare enfasi sul modello adottato durante il corso "Apple Foundation". Si forniranno dettagli sulla struttura e l'architettura dell'applicazione, esplorando le scelte progettuali effettuate per soddisfare i requisiti funzionali e non funzionali. Saranno presentate anche eventuali pattern di design utilizzati e le motivazioni che hanno guidato tali de-

cisioni.

Successivamente, nel capitolo 3, si affronterà la fase di implementazione, descrivendo nel dettaglio le funzionalità principali dell'applicazione. Saranno inclusi mockup che illustrano l'aspetto visivo dell'interfaccia utente, basati su screenshot dell'applicazione reale. Questo permetterà ai lettori di comprendere meglio l'esperienza utente e l'organizzazione delle informazioni all'interno dell'applicazione stessa.

Infine, nel capitolo 4, si fornirà una sintesi completa del progetto sviluppato durante il corso "Apple Foundation". Saranno evidenziate le principali sfide affrontate durante l'intero processo di sviluppo e si discuteranno i risultati ottenuti in termini di funzionalità implementate e soddisfazione degli obiettivi di progetto. Inoltre, si forniranno spunti per future implementazioni e possibili miglioramenti dell'applicazione, prendendo in considerazione aspetti come l'usabilità e l'espandibilità. Attraverso questa struttura di capitoli, la tesi fornirà una panoramica completa del progetto, dalla selezione delle tecnologie all'implementazione finale, offrendo un quadro esaustivo del percorso di sviluppo seguito e delle prospettive future dell'applicazione.

CAPITOLO 1

PANORAMICA DELLE TECNOLOGIE UTILIZZATE

«Il computer è la bicicletta della nostra mente.»
- Steve Jobs

All'interno del progetto, si è fatto ampio uso di una vasta gamma di tecnologie hardware e software, offrendo così un approccio multidisciplinare e integrato. L'approccio multidisciplinare del progetto ha permesso di ottenere risultati significativi, combinando conoscenze di ingegneria del software, informatica e intelligenza artificiale. Grazie alla formazione ricevuta durante il corso "Apple Foundation", è stato possibile acquisire competenze tecniche avanzate. Infine, va sottolineato che la scelta di adottare una vasta gamma di tecnologie hardware e software ha ampliato le potenzialità del progetto, consentendo l'implementazione di soluzioni innovative e all'avanguardia nel campo delle applicazioni digitali. L'integrazione sinergica di queste tecnologie ha permesso di sviluppare un sistema complesso ma ben strutturato, in grado di soddisfare le esigenze e le aspettative dei potenziali utenti.

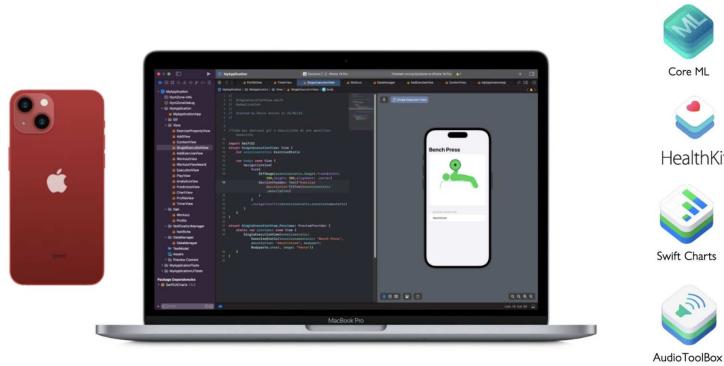


Figura 1.1: Tecnologie e framework utilizzati per il progetto

1.1 Tecnologie Hardware

Nel contesto dello sviluppo del progetto, sono stati impiegati dispositivi all'avanguardia, messi a disposizione durante il corso "Apple Foundation". Questa selezione di dispositivi ha consentito di lavorare con tecnologie di punta e massimizzare le potenzialità del progetto. Tra i dispositivi utilizzati, spiccano quelli rappresentativi di un'evoluzione significativa nel panorama dell'hardware Apple:

1.1.1 Dispositivi

- Il **MacBook Pro M1** [4] è stato presentato nel novembre 2020 ed è la prima generazione di MacBook Pro a essere alimentata dal chip M1. La configurazione di base, con un chip M1, un'unità SSD da 256 GB e 8 GB di RAM, offre una potenza di elaborazione notevole, velocità di archiviazione rapida e una quantità di memoria adeguata per molte attività quotidiane. Una delle caratteristiche più importanti del MacBook Pro M1 è la sua potenza di elaborazione. Il pc presenta un display Retina con una risoluzione nitida e colori vivaci. Presenta un design elegante ed una tastiera retroilluminata. Presenta un touchpad Force Touch, il quale offre un feedback tattile preciso con gesture avanzate. Grazie alla presenza del chip M1, il laptop ha un'incredibile autonomia. Inoltre, riguardo la connettività, il MacBook Pro M1 offre diverse opzioni, quali porte Thunderbolt/USB 4, che consentono di collegare display esterni ad alta risoluzione, unità di archiviazione e periferiche. Supporta anche il Wi-Fi 6 e Bluetooth 5.0 per una connessione senza fili veloce e affidabile. In conclusione, il MacBook è equipaggiato con il sistema operativo macOS Big Sur che offre un'esperienza utente intuitiva ed una vasta gamma di funzionalità. Grazie alla tecnologia di traduzione dei chip M1 è compatibile con molti software e applicazioni, anche quelle sviluppate per i processori Intel.
- L'**iPhone 13** [3] rappresenta una delle ultime generazioni di smartphone di Apple. Dotato di componenti hardware all'avanguardia, l'iPhone 13 offre capacità di elaborazione veloci e un'efficienza energetica ottimizzata. Questo consente di sfruttare al massimo le sue prestazioni durante lo sviluppo del progetto e di creare applicazioni innovative e performanti. Inoltre, l'iPhone 13 è integrato con il sistema operativo iOS, che fornisce un'interfaccia intuitiva e una vasta gamma di strumenti di sviluppo, facilitando il processo di creazione di applicazioni. Infine, la configurazione utilizzata è stata quella con 4 GB di RAM, 128 GB di memoria interna (ROM), GPU 5-core, Wi-Fi 6 e Bluetooth 5.0.

L'utilizzo di questi dispositivi all'avanguardia ha offerto un solido fondamento tecnologico per il progetto. Il Mac Pro con processore M1 ha fornito una potenza di calcolo superiore e un'efficienza energetica ottimale, consentendo un'elaborazione rapida dei dati. L'iPhone 13, invece, ha offerto una piattaforma mobile potente, permettendo di sfruttare le sue capacità di elaborazione e l'efficienza energetica per lo sviluppo di applicazioni innovative. Insieme, il Mac Pro con processore M1 e l'iPhone 13 hanno garantito una collaborazione fluida tra l'ambiente di sviluppo desktop e il dispositivo mobile. Questa sinergia ha facilitato l'implementazione di soluzioni innovative, garantendo la compatibilità e la fluidità tra i due dispositivi durante tutto il processo di sviluppo del progetto.

1.1.2 Componenti

- Il **Neural Engine** è un'unità di elaborazione specializzata progettata per accelerare le operazioni di machine learning. È una componente chiave del chip M1, sviluppato da Apple, e offre prestazioni eccezionali nell'esecuzione di algoritmi di intelligenza artificiale e apprendimento automatico. Il Neural Engine è composto da diverse unità di elaborazione parallela che lavorano insieme per eseguire operazioni matematiche ad alta velocità su grandi quantità di dati. È ottimizzato per svolgere calcoli vettoriali e matriciali, che sono fondamentali in molte operazioni di machine learning, come l'addestramento e l'inferenza di modelli. La sua architettura avanzata consente al Neural Engine di eseguire simultaneamente diverse operazioni di machine learning in parallelo, sfruttando l'ampia potenza di calcolo del chip. Questo si traduce in prestazioni elevate e tempi di risposta rapidi per le applicazioni che utilizzano il machine learning. Inoltre, è progettato per lavorare in stretta collaborazione con il processore CPU e il processore grafico (GPU) integrati nel chip M1. Questa collaborazione tra le diverse unità di elaborazione consente di ottimizzare le prestazioni complessive del sistema, offrendo un'esperienza di utilizzo fluida e reattiva.
- Il **chip M1** è un processore sviluppato da Apple ed introdotto nel novembre 2020. È progettato ed ottimizzato per i sistemi Mac(MacBook Air, MacBook Pro e Mac mini), nei quali le dimensioni ridotte e l'efficienza in termini energetici rivestono un ruolo cruciale. È basato su un'architettura Arm e presenta un'organizzazione a 8 core, tra cui 4 core, chiamati Firestorm, ad alte prestazioni e 4 core, chiamati Icestorm, ad alta efficienza energetica. La combinazione di questi due livelli di core consente di ottenere un ottimo equilibrio tra prestazioni e durata della batteria. Oltre ai core della CPU, il chip M1 include una GU ad alta efficienza energetica, un motore di elaborazione del segnale(DSP) per applicazioni audio e video ed un Neural Engine. La GPU integrata offre ottime prestazioni grafiche, in quanto consente di gestire compiti complessi di grafica e offre un'esperienza visiva straordinaria. Il chip sfrutta anche la tecnologia di processo a 5 nanometri, che consente di ottenere prestazioni superiori a parità di consumo ed un'alta densità di transistor. Grazie a tali caratteristiche elencate, il chip M1 consente ai dispositivi Mac di offrire un'esperienza ottimale all'utente.

1.2 Tecnologie Software

Per lo sviluppo del progetto è stato fatto largo uso delle tecnologie software fornite da Apple, sfruttando appieno le potenzialità dei dispositivi a disposizione.

1.2.1 Sistemi operativi e sviluppo

Tra le tecnologie software per sistemi operativi e sviluppo utilizzate, spiccano:

- **macOS**: è il sistema operativo principale dei computer Mac, fornisce agli utenti un ambiente di lavoro versatile ed efficiente. La sua interfaccia user-friendly e le sue caratteristiche avanzate consentono di svolgere una vasta gamma di attività, inclusa la creazione di software e contenuti multimediali. Inoltre, macOS offre una ricca selezione di strumenti specifici per lo sviluppo, che facilitano la creazione di applicazioni di alta qualità. Grazie alla sua ampia compatibilità con applicazioni

e servizi, tale sistema operativo permette agli utenti di personalizzare l'esperienza di utilizzo e migliorare la produttività.

- **Xcode:** è l'ambiente di sviluppo integrato (IDE) ufficiale di Apple utilizzato per la creazione di applicazioni sia per macOS che per iOS. Questo potente strumento mette a disposizione degli sviluppatori una serie di funzionalità avanzate per la scrittura del codice, il debugging, la creazione di interfacce utente e altro ancora. Xcode offre un'interfaccia intuitiva e un insieme di strumenti integrati che semplificano notevolmente il processo di sviluppo delle applicazioni, consentendo agli sviluppatori di concentrarsi sulla creazione di app di alta qualità.
- **Swift:** è un linguaggio di programmazione moderno, sicuro e performante sviluppato da Apple. È progettato per semplificare il processo di sviluppo delle applicazioni su diversi sistemi operativi Apple, tra cui macOS, iOS, watchOS e tvOS. Una delle principali caratteristiche è la sua sintassi pulita e concisa, che rende il codice più leggibile e facile da comprendere. Ciò permette agli sviluppatori di scrivere applicazioni in modo più efficiente e di ridurre la possibilità di errori. Swift supporta una vasta gamma di paradigmi di programmazione, tra cui la programmazione ad oggetti, la programmazione funzionale e la programmazione orientata agli eventi. Inoltre, offre un sistema di gestione della memoria automatico (chiamato ARC, Automatic Reference Counting) che riduce il carico di lavoro degli sviluppatori nell'allocazione e deallocazione della memoria. Questo aiuta a prevenire i problemi comuni di gestione della memoria, come i memory leak, e semplifica lo sviluppo di applicazioni più stabili e performanti. Swift supporta anche la programmazione reattiva e l'uso di librerie e framework esterni attraverso il sistema di gestione dei pacchetti Swift Package Manager. Ciò consente agli sviluppatori di integrare facilmente funzionalità aggiuntive nelle proprie applicazioni e di sfruttare le risorse della community per accedere a soluzioni già esistenti.

1.2.2 Framework

Oltre a queste tecnologie principali, abbiamo anche sfruttato i seguenti framework specifici:

- **SwiftUI:** è un framework di sviluppo per la creazione di interfacce utente nelle applicazioni iOS, macOS, watchOS e tvOS. Offre un approccio dichiarativo per la creazione di interfacce, ovvero è possibile descrivere come appare la schermata e come deve comportarsi mediante un codice dichiarativo. Una delle caratteristiche più importanti di tale framework è la sua semplicità e la facilità d'uso. Sfrutta diversi concetti come la composizione di viste, la gestione automatica dello stato e la capacità di anteprima in tempo reale, utile allo sviluppatore per visualizzare le modifiche appena apportate. Con SwiftUI è possibile creare interfacce utente sfruttando la combinazione di diversi componenti chiamati viste. Quest'ultime possono essere annidate o composte insieme. Un'altra caratteristica importante di SwiftUI è la sua gestione automatica dello stato, infatti tiene traccia dei diversi state delle viste e si occupa automatica dell'aggiornamento dell'interfaccia. Questo semplifica la gestione delle interazioni utente. Inoltre, offre dei vantaggi in termini di prestazioni e multipiattaforma. Le applicazioni SwiftUI sono altamente ottimizzate e sfruttano al massimo le funzionalità del sistema operativo.

- **HealthKit:** è un framework specifico per iOS che fornisce agli sviluppatori un’interfaccia per accedere ai dati relativi alla salute e al fitness dei dispositivi Apple. Con HealthKit, è possibile recuperare informazioni come i battiti cardiaci, le calorie bruciate, i dati di attività fisica, il sonno e altro ancora. Questi dati possono essere utilizzati per creare applicazioni di monitoraggio della salute, programmi di fitness personalizzati e altre soluzioni innovative nell’ambito del benessere.
- **CoreML:** è un framework di machine learning [2] integrato in iOS e macOS, progettato per consentire agli sviluppatori di incorporare modelli avanzati di intelligenza artificiale nelle proprie applicazioni. Questo framework offre una serie di funzionalità potenti che vanno oltre il semplice riconoscimento di immagini e il riconoscimento del linguaggio naturale. Con CoreML, gli sviluppatori possono creare applicazioni in grado di analizzare dati complessi, effettuare previsioni, prendere decisioni intelligenti e fornire esperienze personalizzate agli utenti. Ad esempio, è possibile utilizzare CoreML per creare applicazioni di riconoscimento facciale, che consentono agli utenti di sbloccare il loro dispositivo utilizzando la loro impronta facciale o per applicazioni di riconoscimento di oggetti, che permettono di identificare gli oggetti presenti in un’immagine o in un video. Un aspetto particolarmente interessante di questo framework è la sua capacità di sfruttare l’accelerazione hardware disponibile sui dispositivi Apple, come il Neural Engine presente nei processori Apple. Questo permette di eseguire le operazioni di machine learning in modo più efficiente e veloce, offrendo prestazioni ottimizzate e un’esperienza utente fluida. Inoltre, CoreML supporta una vasta gamma di modelli di machine learning, inclusi modelli addestrati utilizzando framework come TensorFlow e PyTorch. Questo apre le porte a una vasta gamma di possibilità per gli sviluppatori, che possono utilizzare modelli pre-addestrati o addestrare i propri modelli personalizzati per rispondere alle specifiche esigenze delle loro applicazioni. CoreML semplifica anche l’integrazione dei modelli di machine learning nelle applicazioni. Fornisce strumenti e librerie per la conversione dei modelli in un formato compatibile con CoreML e offre un’interfaccia di programmazione (API) intuitiva che permette di utilizzare facilmente i modelli nelle applicazioni.
- **Charts:** è un framework di visualizzazione dei dati [1] disponibile per iOS e macOS. Offre una vasta gamma di grafici interattivi e personalizzabili, tra cui grafici a torta, grafici a barre, grafici a linee, grafici radar e altro ancora. Con Charts, gli sviluppatori possono visualizzare in modo efficace e accattivante i dati all’interno delle loro applicazioni, fornendo un’esperienza utente coinvolgente e comprensibile.
- **AudioToolbox:** è un framework per la gestione dei suoni e degli effetti audio su macOS e iOS. Fornisce un set di strumenti per la riproduzione, registrazione e manipolazione dell’audio all’interno delle applicazioni. Con AudioToolbox, gli sviluppatori possono creare esperienze audio coinvolgenti, inclusi effetti sonori, riproduzione di musica, registrazione audio e altro ancora. Questo framework offre anche funzionalità per la gestione di input e output audio, come la gestione dei dispositivi di input e output e il controllo del volume.

Questi framework offrono strumenti e funzionalità specifiche che consentono agli sviluppatori di creare applicazioni innovative e performanti su dispositivi Apple, ampliando le capacità delle app attraverso la gestione dei dati relativi alla salute, l’integrazione di

modelli di machine learning, la visualizzazione dei dati tramite grafici interattivi e la gestione degli effetti audio.

CAPITOLO 2

PROGETTAZIONE DELLA SOLUZIONE

«Lavorare in squadra è bello perché c'è sempre qualcuno dalla tua parte.»
- Margaret Carty

La fase di progettazione è fondamentale per il successo di qualsiasi progetto. Durante questa fase iniziale, vengono stabiliti gli obiettivi del progetto e definiti i requisiti e le aspettative degli utenti finali, fondamentale per guidare le decisioni che verranno prese lungo il percorso di sviluppo. Inoltre, la fase di progettazione permette la pianificazione di tutte le attività necessarie per il completamento del progetto, come per esempio l'identificazione delle risorse necessarie e la stima dei tempi di consegna. Un altro aspetto rilevante della fase di progettazione è l'ottimizzazione delle risorse disponibili con la quale, attraverso una pianificazione precisa ed una corretta assegnazione delle risorse, riesce ad evitare sprechi e massimizzare l'efficienza. Questo include l'allocazione del personale e del tempo necessario per il completamento del progetto. Proprio per questo tale fase favorisce la comunicazione e la collaborazione efficaci tra le diverse parti interessate e i membri del team del progetto. Bisogna tenere in considerazione che il coinvolgimento di tutte le persone del progetto permette la condivisione di idee, la risoluzione di problemi e l'allineamento degli obiettivi. Dunque, in tale fase iniziale una buona comunicazione e collaborazione creano una base solida per un flusso di lavoro senza la presenza di intoppi in fase di implementazione.

Per questo motivo ad ogni componente del gruppo è stato assegnato uno specifico ruolo prima di partire con la progettazione. Personalmente, durante il progetto, ho ricoperto il ruolo di "Lead Developer" e "Developer". In generale il "Lead Developer" ha il compito di coordinare le attività del team, garantire la chiarezza degli obiettivi e dei requisiti del progetto, prendendo decisioni tecniche. Il "Developer" invece si impegna attivamente nello sviluppo di codice e nella realizzazione delle funzionalità richieste.

2.1 CBL



Figura 2.1: Challenge Based Learning

CBL [7] è l'acronimo di “Challenge Based Learning” (Apprendimento basato su sfide). L'approccio CBL mira a coinvolgere gli studenti in un apprendimento autentico, collegando gli argomenti di studio con problemi del mondo reale e promuovendo la collaborazione, la creatività e le competenze di problem solving. Attraverso le tre fasi di “Engage” (Coinvolgere), “Investigate” (Indagare) e “Act” (Agire), gli studenti sono in grado di acquisire conoscenze approfondite e sviluppare abilità che possono essere applicate in situazioni reali.

1. **Engage:** In questa fase si identifica e definisce chiaramente il problema da affrontare mediante una comprensione approfondita del contesto. Infatti, attraverso domande essenziali, si passa da un'idea generale ad una sfida concreta ed attuabile. Vi sono 3 sottostadi:
 - (a) **Big Idea:** Il miglioramento dell'allenamento ottimizzando l'organizzazione e fornendo metodi per essere seguiti, sia da un coach fisico che non.
 - (b) **Essential Question:** Dopo aver stilato diverse domande inerenti con la big idea, si è arrivati alla definizione della domanda essenziale, ovvero “Come l'organizzazione ed il coaching influenzano il risultato ottenuto facendo sport?”
 - (c) **Challenge:** La challenge trasforma una domanda essenziale in un invito all'azione per approfondire lo studio del soggetto. Una challenge è immediata, attuabile e crea entusiasmo.
2. **Investigate:** In questa fase vengono generate idee creative e diverse per affrontare il problema permettendo il coinvolgimento del pensiero laterale, della divergenza delle idee e dell'apertura a nuove possibilità. Possono essere utilizzate strategie

come il brainstorming, il mind mapping o le sessioni di discussione di gruppo per generare una vasta gamma di soluzioni potenziali. L'obiettivo di questa fase è sviluppare esperienze di apprendimento contestualizzate e condurre una ricerca rigorosa basata sui contenuti e sui concetti per creare una base per soluzioni concrete e sostenibili. Vi sono 3 sottofasi:

- (a) **Guiding questions:** Fase in cui si definiscono un insieme di domande utili per sviluppare una soluzione, quindi l'obiettivo è capire la realtà d'interesse. In seguito, vengono unite tutte le domande simili tra loro e vanno categorizzate e posizionate in ordine di priorità.
 - (b) **Guiding Activities/Resources:** Identificare un insieme di azioni rigorose e risorse per rispondere alle domande guida precedentemente ottenute.
 - (c) **Analysis:** Utilizzando ciò che è stato appreso dalle risposte alle domande guida, creare una ricerca e, dopo aver letto i risultati di quest'ultima, sviluppare un elenco che riassume le scoperte più rilevanti ai fini del progetto.
3. **Act:** In questa fase vengono sviluppate soluzioni basate sulla sintesi di ricerca e sulla narrazione della fase precedente, implementate con un pubblico autentico ed i risultati vengono valutati. Vi sono 3 sottofasi:
 - (a) **Solution:** Definire una soluzione basata sulla ricerca.
 - (b) **Implementation:** Implementazione della soluzione.
 - (c) **Evaluation:** Valutazione dei risultati.

Riguardo il progetto:

2.1.1 Engage

- **Big Idea:** Migliorare l'allenamento ottimizzando l'organizzazione e fornendo un metodo per essere supportati, con o senza un coach reale.
- **Essential Questioning:**
 - Essere seguiti da un coach migliora l'allenamento?
 - Qual è l'utilità del coach nello sport?
 - Quali sono i principali fattori che influenzano il fitness?
 - Come possono i principianti senza un personal trainer assicurarsi che le loro esecuzioni siano corrette?
 - Quali sono i diversi modi nel fitness per tenere traccia dei progressi?
 - Quali sono i fattori che influenzano la motivazione del workout?
 - Come ci si può allenare da soli senza andare in palestra?
- **Essential Question:** Come l'organizzazione in palestra ed il coaching influenzano i risultati ottenuti praticando sport?
- **Challenge:** Migliorare l'allenamento attraverso un'organizzazione efficiente, la storia degli allenamenti e i metodi corretti di esecuzione per qualsiasi tipo di allenamento.

2.1.2 Investigate

- **Guiding Questions:** (disposte in ordine di priorità)

1. Quanto è importante il fitness nella vita quotidiana?
2. Quanto è importante seguire una dieta equilibrata nella vita quotidiana?
3. Quanto può essere conveniente avere un piano alimentare e di esercizio unificato e registrato in un unico luogo facilmente accessibile?
4. Può essere utile avere un diario per registrare i carichi di allenamento in palestra al fine di ottenere un resoconto finale comparabile con quelli precedenti?
5. Può essere dimenticata l'esecuzione corretta di un esercizio?
6. Può essere utile avere un sistema automatico che conta le serie, le ripetizioni ed il tempo di recupero?
7. Può essere utile avere un modo per ricevere un piano di allenamento dal proprio coach e fornirgli i propri dati per studiare nuovi piani di allenamento?
8. Quanto è importante avere un piano di allenamento settimanale al fine di ottenere buoni risultati?
9. Può essere utile avere un sistema che guida l'individuo durante l'esecuzione dell'esercizio al fine di mantenere la corretta postura?
10. Se fosse presente un piano alimentare che richiede integratori, sarebbe utile avere un promemoria per prendere le pillole?
11. Può essere utile visitare una pagina dedicata in cui è possibile leggere recensioni di altri atleti e vedere i lavori precedenti di un allenatore per poterlo scegliere?
12. Può essere motivante conoscere i risultati fisici delle altre persone?
13. È utile essere motivati durante gli allenamenti?

- **Guiding Activities/Resources:** Mediante il form di Google e domande faccia a faccia sono state raccolte le risposte alle domande più importanti, arrivando a tali conclusioni:

- Le persone hanno molta difficoltà ad organizzare il loro allenamento.
- Non tutti hanno difficoltà a leggere il proprio foglio di allenamento, ma vi sono persone che lo perdono.
- Molto spesso le persone dimenticano il peso alzato in fase di allenamento.
- Generalmente le persone si allenano dalle 2 alle 3 volte a settimana.
- Le persone non si sentono motivate durante l'allenamento, eccetto per qualche caso in cui l'individuo è molto appassionato.
- La maggior parte delle persone si dimostra interessata ai propri progressi.
- In generale, le persone sono seguite da un allenatore, solo una piccola parte di loro si allena da soli.
- La maggior parte delle persone non ritiene importante assumere integratori, ad eccezione di coloro che sono molto appassionati alla palestra.

- Alcune persone non incontrano mai l’allenatore o, quando ne hanno bisogno, non riescono a contattarlo.
 - La maggior parte delle persone perde molto spesso il conto delle proprie ripetizioni.
 - Con il passare del tempo, molte persone dimenticano l’esecuzione corretta di un esercizio.
 - Non tutti si concentrano sui miglioramenti degli altri, ad eccezione di coloro che sono molto competitivi.
 - Non tutti sono motivati dai miglioramenti degli altri.
- **Analysis:** Le risposte evidenziano la difficoltà delle persone nell’organizzare gli allenamenti, il bisogno di motivazione e la difficoltà nel ricordare la struttura dei loro allenamenti.

2.1.3 Act

- **Solution:** Un’app che raggruppa gli allenamenti delle persone, gestisce il peso e ripetizioni e fornisce l’esecuzione corretta di ogni esercizio. Inoltre, permette di tenere traccia delle calorie bruciate nel passato e quelle che verranno bruciate nel futuro al fine di motivare le persone.
- **Group Reflection:** Crediamo fermamente che il concetto che offriamo sia il modo migliore per raggiungere la mentalità corretta durante l’allenamento. Inoltre, possiamo affermare che con ulteriori miglioramenti che desideriamo aggiungere, l’applicazione potrà essere sempre più utile per ogni tipo di allenamento e potrà aiutare le persone a essere molto motivate e competitive durante la pratica di qualsiasi sport.

2.2 Design

Dopo aver definito l’idea e trovato la soluzione, si passa alla fase di design [8]. L’interaction design è un’area multidisciplinare che definisce come i software debbano interagire con le persone. Quando si progetta un’interfaccia bisogna innanzitutto analizzare quali sono i requisiti funzionali, ovvero ciò che l’applicazione deve fare e non come deve farlo, i requisiti non funzionali, ovvero i vincoli dell’applicazione o una caratteristica a cui è associato un livello di qualità, i requisiti tecnici che indicano con quali tecnologie l’applicazione dovrà interagire ed infine i requisiti ambientali che tratta il posto in cui il prodotto verrà utilizzato. In seguito, bisogna descrivere il profilo degli utenti che useranno l’applicazione. Esistono 4 tipologie di utenti:

- **Novizio:** richiede istruzioni passo dopo passo, probabilmente con suggerimenti ed un’interazione vincolata supportata da informazioni chiare.
- **Esperto:** richiede un’interazione flessibile con maggiori poteri di controllo.
- **Frequente:** per tale tipologia è importante fornire scorciatoie come i tasti di funzione anziché aspettarsi che digit lunghe stringhe di comandi o che navighi attraverso una struttura di menù.
- **Casuale:** richiede istruzioni chiare e comandi facilmente comprensibili.

L'obiettivo del progetto è realizzare un'app in grado di risolvere i problemi citati nel CBL sfruttando la tecnologia iOS. Dunque, è necessario seguire le linee guida designate per i dispositivi iOS per la realizzazione di interfacce grafiche.

Troviamo diversi aspetti riguardo il design:

2.2.1 Principi di design

- **Integrità estetica:** si riferisce a quanto bene l'aspetto ed il comportamento di un'app si adattano alla sua funzione. Ad esempio, un'app progettata per svolgere compiti seri può mantenere l'utente concentrato utilizzando grafiche discrete e non invadenti, controlli standard e comportamenti prevedibili. D'altro canto, un'app coinvolgente può offrire un aspetto avvincente che permette all'utente di divertirsi ed emozionarsi.
- **Consistenza:** un'app consistente implementa standard e paradigmi familiari utilizzando elementi dell'interfaccia forniti dal sistema, icone note, stili dello standard ed una terminologia uniforme. In generale, gli elementi con funzioni simili dovrebbero avere un aspetto simile.
- **Manipolazione diretta:** La manipolazione diretta del contenuto sullo schermo coinvolge attivamente gli utenti e favorisce una maggiore comprensione. Quando gli utenti ruotano il dispositivo o utilizzano gesti per interagire con il contenuto visualizzato, sperimentano direttamente l'effetto delle loro azioni.
- **Feedback:** riconosce le azioni degli utenti e mostra i risultati per tenerli informati. Le app integrate di iOS sono progettate per offrire un feedback percepibile in risposta ad ogni interazione dell'utente. Durante le operazioni a lungo termine, vengono utilizzati indicatori di avanzamento per comunicare lo stato dell'operazione in corso. Inoltre, l'uso di animazioni e suoni può aiutare a chiarire i risultati delle azioni compiute dagli utenti. Il feedback è essenziale per fornire agli utenti una risposta immediata e comprensibile alle loro azioni, migliorando così l'esperienza complessiva e mantenendo gli utenti informati sullo stato dell'applicazione.
- **Metafore:** funzionano bene su iOS perché le persone interagiscono fisicamente con lo schermo. Spostano le viste per scoprire il contenuto sottostante, trascinano e scorrono il contenuto, attivano interruttori, muovono cursori e scelgono valori all'interno di selettori. Possono persino sfogliare pagine di libri e riviste come se fossero fisicamente presenti. L'utilizzo di metafore familiari semplifica l'apprendimento e l'uso dell'app, in quanto sfruttano le esperienze già conosciute dagli utenti. Questo approccio consente di creare un'interazione intuitiva e fluida, in cui le azioni richieste sono coerenti con le aspettative dell'utente. Ogni volta che si utilizza una metafora per dare informazioni agli utenti, bisogna capire il background culturale nella quale stiamo lavorando.
- **Controllo dell'utente:** nel mondo iOS sono gli utenti e non le app ad avere il controllo. Le app possono suggerire azioni e avvertire delle conseguenze, ma è un errore prenderne il controllo decisionale al posto dell'utente. Le migliori app trovano un giusto equilibrio tra consentire agli utenti di prendere decisioni e prevenire esiti indesiderati. Un'app può far sentire le persone al controllo attraverso diverse modalità, ad esempio mantenendo gli elementi interattivi familiari e prevedibili, richiedendo una conferma per azioni distruttive e offrendo un'opzione

semplice per annullare operazioni, anche se sono già in corso. L'obiettivo è far sentire gli utenti al controllo, sicuri e capaci di correggere eventuali errori in modo semplice ed efficace durante l'utilizzo dell'app.

2.2.2 Design Visivo

- **Layout:** è importante configurare gli elementi dell'interfaccia ed i layout in modo tale che si adattino alle dimensioni dei diversi dispositivi. Anche se l'app fosse disponibile per un dispositivo specifico, bisogna assicurarsi che funzioni su tutte le dimensioni dello schermo per quel determinato dispositivo. L'obiettivo è garantire un'esperienza coerente e di alta qualità per gli utenti, indipendentemente dal dispositivo che viene utilizzato.
- **Multitasking:** gli utenti sono abituati a passare da un'app all'altra. Su iPad è possibile utilizzare due app contemporaneamente attraverso diverse modalità come Slide Over, Split View e Picture.

2.2.3 Modalità

- **Modalità:** è una tecnica di design che presenta i contenuti in una modalità temporanea separata dal contesto corrente dell'utente e richiede un'azione esplicita per uscirne. La modalità temporanea può essere utilizzata per offrire una visualizzazione dedicata per una specifica attività o per fornire informazioni critiche che richiedono l'attenzione e l'eventuale azione dell'utente. L'obiettivo è fornire un'esperienza chiara e focalizzata, consentendo agli utenti di concentrarsi sulle attività importanti e di gestire le informazioni in modo efficace.

2.2.4 Branding

- **Branding:** bisogna utilizzare un branding raffinato e poco invasivo. L'obiettivo è creare un'esperienza coerente e familiare per gli utenti iOS, in modo che si sentano a proprio agio nell'utilizzare l'applicazione e che si integrino armoniosamente con il resto del sistema operativo.

2.2.5 Colore

- **Colore:** è un ottimo modo per impartire vitalità, fornire continuità visiva, comunicare informazioni sullo stato, fornire feedback in risposta alle azioni degli utenti ed aiutare le persone a visualizzare i dati. È importante la coerenza nell'utilizzo del colore in quanto essa contribuisce a consolidare l'identità visiva dell'app e conseguentemente a garantire una connessione visiva con gli utenti creando un'esperienza unificata e professionale.

2.2.6 Terminologia e Tipografia

- **Terminologia:** ogni parola nell'applicazione fa parte di una conversazione con gli utenti ed è fondamentale usare tale conversazione per farli sentire a proprio agio. Bisogna scegliere sempre frasi familiari, comprensibili con un tono informale ed amichevole, prestare però attenzione all'uso dell'umorismo. Quest'ultimo può variare tra culture, quindi bisogna assicurarsi che sia comprensibile e adatto a tutti gli utenti.

- **Tipografia:** San Francisco (SF) è una famiglia di caratteri sans serif. SF Pro è il carattere di sistema utilizzato in iOS, macOS e tvOS, mentre SF Compact è il carattere di sistema utilizzato in watchOS.

2.2.7 Navigazione

- **Navigazione gerarchica:** è importante fare una scelta per schermata fino a raggiungere la destinazione desiderata. Per andare da una destinazione all'altra si hanno due opzioni: tornare indietro seguendo lo stesso percorso che è stato fatto in precedenza, oppure ricominciare dall'inizio e fare scelte diverse lungo il percorso.
- **Navigazione flat:** passaggio tra diverse categorie di contenuti.
- **Navigazione basata sui contenuti o sull'esperienza:** spostarsi liberamente tra i contenuti, o i contenuti stessi definiscono la navigazione.

2.2.8 Barre

- **Navigation bar:** compare nella parte superiore dello schermo di un'applicazione, sotto la barra di stato, e consente la navigazione attraverso una serie di schermate gerarchiche. Talvolta il lato destro di una barra di navigazione contiene un pulsante per la gestione del contenuto della vista.
- **Search bar:** consente alle persone di effettuare ricerche all'interno di una vasta collezione di valori digitando del testo in un campo. La barra di ricerca può essere visualizzata da sola o all'interno di una barra di navigazione o di una vista del contenuto. Utilizza una barra di ricerca al posto di un campo di testo per implementare la funzione di ricerca. Viene abilitato il pulsante “Cancella” per consentire agli utenti di cancellare il testo inserito.
- **Scope bar:** consente all'utente di raffinare il campo di ricerca.
- **Sidebar:** offre la navigazione a livello di applicazione ed un accesso rapido alle collezioni di contenuti di alto livello. Selezionando un elemento nella barra laterale, le persone possono navigare verso un contenuto specifico.
- **Tab bar:** è una componente dell'interfaccia che si trova nella parte inferiore dello schermo di un'applicazione. Essa consente alle persone di passare rapidamente tra diverse sezioni dell'applicazione. Le linee guida dell'interfaccia utente consigliano di utilizzare una tab bar esclusivamente per la navigazione tra le sezioni dell'app. È importante trovare un equilibrio nella quantità di schede che vengono fornite. Troppe schede possono ridurre l'area cliccabile di ciascuna scheda e rendere l'interfaccia più complessa da navigare. D'altra parte, un numero troppo limitato di schede può far sembrare l'interfaccia frammentata e rendere difficile per gli utenti accedere rapidamente alle diverse sezioni dell'app. In generale, è consigliato utilizzare da tre a cinque schede su iPhone e, se necessario, un numero leggermente superiore su iPad. È sconsigliato inserire la sezione home nella tab bar.
- **Toolbar:** compare nella parte inferiore dello schermo di un'app e contiene pulsanti per eseguire azioni pertinenti alla vista corrente o al contenuto al suo interno.

Le barre degli strumenti spesso vengono nascoste quando le persone non ne hanno bisogno.

2.2.9 Viste

- **Action sheet:** permette di ottenere uno specifico avviso che presenta due o più opzioni correlate al contesto corrente.
- **Activity view:** un'attività è un'azione, come ad esempio Copia, che è utile nel contesto attuale.
- **Alarm:** trasmette informazioni importanti relative allo stato dell'applicazione o del dispositivo e spesso richiedono un feedback. Un avviso è composto da un titolo, un messaggio opzionale, uno o più pulsanti e campi di testo opzionali per raccogliere input.
- **Collections:** gestisce un insieme ordinato di contenuti, come ad esempio un set di foto, e li presenta in un layout personalizzabile.
- **Page:** fornisce un modo per implementare la navigazione lineare tra le pagine di contenuto, come in un documento, un libro, un blocco note o un calendario.
- **Popover:** Un popover è una vista temporanea che appare sopra altri contenuti sullo schermo quando si tocca un controllo o un'area specifica. Il popover va evitato su iPhone.
- **Scroll view:** consente agli utenti di sfogliare contenuti, come testo in un documento o una collezione di immagini, che è più grande dell'area visibile.
- **Table:** presenta i dati come una lista di righe scorrevole a singola colonna che può essere divisa in sezioni o gruppi. Utilizza una tabella per visualizzare in modo pulito ed efficiente grandi o piccole quantità di informazioni sotto forma di lista.

2.3 Mockup

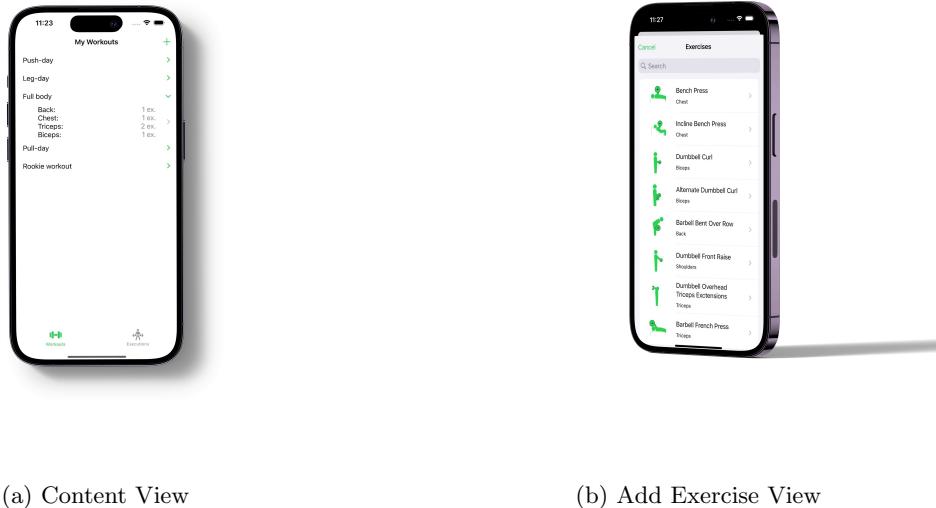


Figura 2.2: Views esercizio

La content View (Figura 2.3a) presenta una tab bar con la quale è possibile navigare tra le diverse viste, una disclosure group ed un pulsante in alto a destra con la quale si accede alla Add Exercise View (Figura 2.3b). In questa view è presente una search bar ed un pulsante per poter ritornare indietro.

La Workout View(Figura 2.3 a) presenta una tab bar, un pulsante in alto a sinistra che permette di accedere alla view precedentemente visitata ed un pulsante "play" in alto a destra che permette di andare alla Play View(Figura 2.3 a). Quest'ultima presenta una tab bar ed un pulsante in alto a destra per accedere alla view precedentemente visitata.

2.4 Strutture Dati

Le strutture dati svolgono un ruolo fondamentale nella programmazione e nell'elaborazione dei dati.

2.4.1 Proprietà

Tra i diversi ruoli principali abbiamo:

- **Accesso e ricerca efficiente:** Le strutture dati forniscono meccanismi per l'accesso efficiente ai dati. Ad esempio, gli array consentono un accesso diretto agli elementi tramite un indice numerico, mentre i dizionari consentono di cercare e recuperare valori in base a una chiave specifica. La scelta della struttura dati appropriata influenza sulla velocità di esecuzione delle operazioni di ricerca e accesso.
- **Manipolazione e modifica dei dati:** Le strutture dati offrono operazioni per l'inserimento, l'eliminazione e la modifica dei dati. Ad esempio, gli array forniscono metodi per l'aggiunta o rimozione di elementi in posizioni specifiche, mentre

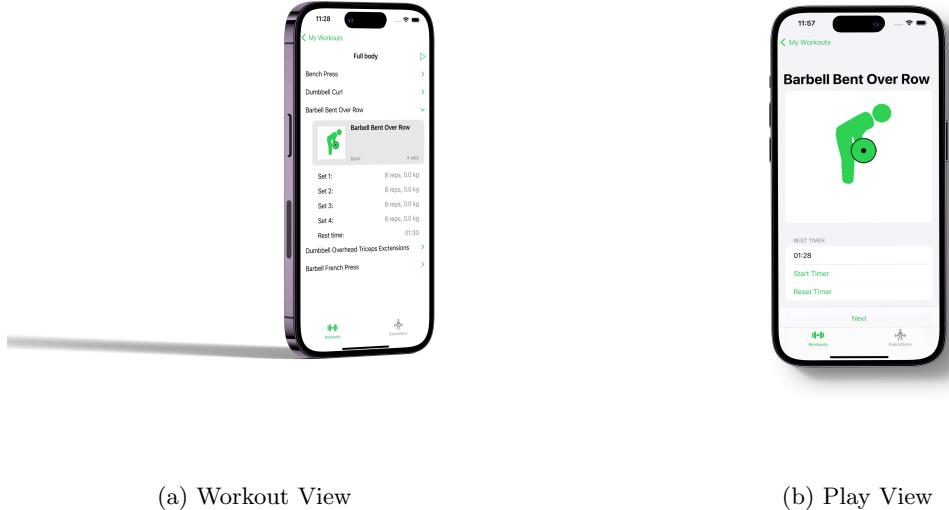


Figura 2.3: Views dettagli esercizio

le code consentono l'inserimento di elementi in coda e la rimozione dalla testa. Le strutture dati definiscono le regole e le operazioni per mantenere l'integrità e la coerenza dei dati.

- **Implementazione di algoritmi complessi:** Molte operazioni e algoritmi complessi richiedono l'uso di strutture dati specifiche. Ad esempio, gli alberi binari sono utilizzati nell'algoritmo di ricerca binaria e nelle strutture dati come gli alberi di decisione. Le strutture dati offrono un modo per rappresentare e manipolare i dati in modo efficiente, consentendo l'implementazione di algoritmi complessi.
- **Ottimizzazione delle prestazioni:** Le strutture dati possono essere ottimizzate per migliorare le prestazioni dell'applicazione. Ad esempio, un algoritmo di ordinamento può trarre vantaggio dall'utilizzo di una struttura dati come l'heap o l'albero binario per migliorare l'efficienza. La scelta delle strutture dati appropriate può ridurre i tempi di esecuzione e l'utilizzo delle risorse.

Analizzando proprio gli algoritmi di ordinamento [6] questi hanno più classificazioni. Una prima distinzione dipende da quale dispositivo viene usato per mantenere la sequenza di elementi: si tratta di ordinamento interno se la sequenza è contenuta completamente in memoria centrale, e di ordinamento esterno se invece la sequenza è in un file. Una seconda distinzione è l'ordinamento sul posto, dove la sequenza ordinata sostituisce la sequenza di partenza, occupando la stessa struttura dati e al più usando strutture dati con complessità spaziale $O(\log n)$. Invece, negli algoritmi di ordinamento non sul posto viene creata una nuova struttura dati per memorizzare temporaneamente la sequenza, con un'occupazione di $\omega(n)$. In genere, per l'ordinamento interno sono preferibili algoritmi sul post, mentre per l'ordinamento esterno sono spesso usati algoritmi non sul post.

2.4.2 Tipologia

In Swift, il linguaggio di programmazione sviluppato da Apple ed utilizzato per la realizzazione del progetto, sono disponibili diverse strutture dati per la gestione e la manipolazione dei dati. Tra le più comuni troviamo:

- **Array:** collezione ordinata di elementi omogenei. Gli array possono contenere elementi di tipo numerico, stringhe, un tipo di oggetti e così via. Gli elementi di un array vengono memorizzati in modo consecutivo nella memoria e sono accessibili mediante un indice numerico. In Swift vi sono gli array dinamici, ovvero array che modificano la propria dimensione in base alle necessità.
- **Dictionary:** collezione non ordinata di coppie chiave-valore. Le chiavi in un dizionario sono uniche e vengono utilizzate per accedere ai valori associati. I dizionari sono utilizzati quando si ha la necessità di accedere ad un elemento mediante la chiave e non mediante un indice numerico.
- **Set:** collezione non ordinata di elementi unici. Sono particolarmente utili quando si desidera rimuovere duplicati da una sequenza di valori. Tale struttura dati non è ordinata.
- **Stack:** struttura dati con politica “last-in, first-out” (LIFO). Gli elementi vengono inseriti e rimossi solo dalla cima dello stack. Il comportamento è simile ad una pila di oggetti fisici in cui è possibile aggiungere o rimuovere solo l'elemento superiore.
- **Queue:** struttura dati con politica “first-in, first-out” (FIFO). In Swift, è possibile implementare una coda con l'utilizzo di un array o una struttura dati personalizzata.
- **Struct:** sono un tipo di dato personalizzato che permette di definire una struttura dati specifica all'interno di un'applicazione. Le struct sono utilizzate principalmente per encapsulare dati correlati ed offrire funzionalità associate ad essi.

2.4.3 Differenza tra Struct e Classi

Classi e struct presentano alcune differenze. Innanzitutto, le struct sono tipi di valori, mentre le classi tipo di riferimento. Questo significa che quando viene assegnata una struct ad una variabile, viene creata una copia indipendentemente dal suo valore. Al contrario, quando una classe viene assegnata ad una variabile, viene condiviso un riferimento al valore esistente. Inoltre, le struct forniscono un costruttore predefinito automatico che inizializza tutte le proprietà della struct, mentre le classi richiedono l'implementazione di costruttori personalizzati per inizializzare le proprietà. Le classi, a differenza delle struct, supportano l'ereditarietà, il che significa che una classe può ereditare proprietà e metodi da una classe base e aggiungere o sovrascrivere funzionalità. Anche se entrambe possono conformarsi a protocolli in Swift, le struct sono più adatte per la conformità a protocolli quando si desidera definire un tipo di valore con proprietà e metodi correlati, mentre le classi sono più adatte quando si desidera definire un tipo di riferimento che può essere utilizzato in modo polimorfico attraverso l'ereditarietà. Infine, le proprietà delle struct sono per impostazione predefinita immutabili, a meno che non vengano dichiarate come variabili utilizzando la parola chiave var. Le proprietà delle classi possono essere modificate anche senza dichiararle come variabili.

2.4.4 Modello dei dati nell'applicazione

All'interno dell'applicazione, per la gestione dei diversi workout presenti nella Content View, è stato utilizzato un modello di dati che comprende una classe e diverse strutture. Questo modello è stato progettato per organizzare in modo efficace le informazioni riguardanti gli allenamenti, consentendo una gestione completa e flessibile. La classe principale utilizzata per rappresentare un workout contiene un array di struct chiamate "Workout". Ogni istanza di questa struct rappresenta un singolo workout e contiene un id, un nome, una data di creazione, un oggetto DateComponents (per rappresentare le componenti di una data) ed un array di struct chiamate "Exercise". Quest'ultima rappresenta un esercizio all'interno del singolo workout ed è composto da un id, una struttura "ExerciseStatic" utilizzata per generare la scheda di allenamento in base ai dati presenti nella sezione "Profilo", variabili per il tempo di recupero (minuti e secondi) ed un array di struct con il nome di "Sets". A sua volta, la struct "Sets" contiene un id e due variabili per indicare il peso utilizzato nelle serie ed il numero di ripetizioni. Un altro costrutto utilizzato all'interno del modello dati è l'enum chiamato "Bodyparts" ed è stato utilizzato per classificare le diverse parti del corpo che intervengono durante l'esecuzione di un particolare esercizio. Infine, per poter memorizzare e visualizzare mediante le calorie bruciate nei tre giorni precedenti al giorno corrente e quelle che verranno bruciate nei tre giorni successivi, sono state utilizzate due struct. La prima è "Calor" e contiene un id e due variabili che indicano rispettivamente il numero di calorie ed il giorno della settimana mentre l'altra struct, con il nome di "Ser", contiene un id, un nome e un array di struct ("Calor").

CAPITOLO 3

IMPLEMENTAZIONE E DESCRIZIONE DELLE FUNZIONALITÀ

«La semplicità è la suprema sofisticazione.»

- Leonardo Da Vinci

La fase di implementazione è una delle fasi fondamentali di un progetto, durante la quale si traducono i requisiti funzionali e non funzionali ed il design concettuale in una soluzione concreta. In questa fase, vengono scritte e testate le linee di codice per realizzare le funzionalità definite nella fase di progettazione. Durante l'implementazione, è importante seguire buone pratiche di sviluppo software al fine di garantire la qualità, l'efficienza e la manutenibilità del codice. In questa fase vi sono vari aspetti da curare come la scelta del linguaggio di programmazione che, in questo caso, per le ragioni spiegate precedentemente è stato utilizzato Swift e l'organizzazione del codice, in quanto è importante strutturarlo in modo modulare e organizzato. Questo permette di suddividere le funzionalità in componenti distinti e favorisce la riusabilità del codice. Inoltre, l'utilizzo di una convenzione di denominazione coerente e di commenti adeguati aiuta a rendere il codice più comprensibile per gli sviluppatori futuri (nel caso in cui il progetto dovesse essere portato avanti). Un altro aspetto da tenere in considerazione è il testing, in quanto effettuare test regolari permette di verificare che il test rispetti i requisiti funzionali e non funzionali. In un progetto, i requisiti più importanti sono quelli non funzionali in quanto, se non rispettati, il sistema non funzionerebbe. I test possono essere automatizzati mediante l'uso di framework di testing o eseguiti manualmente. L'obiettivo è individuare e correggere eventuali errori o bug all'interno del codice. Infine, un altro aspetto importante è la documentazione poiché aiuta gli sviluppatori a comprendere il funzionamento delle diverse parti del software, semplifica la manutenzione e agevola eventuali future estensioni o modifiche. È possibile utilizzare commenti nel codice o scrivere documentazione esterna, come guide per gli sviluppatori o documenti tecnici.

3.1 Funzionamento View

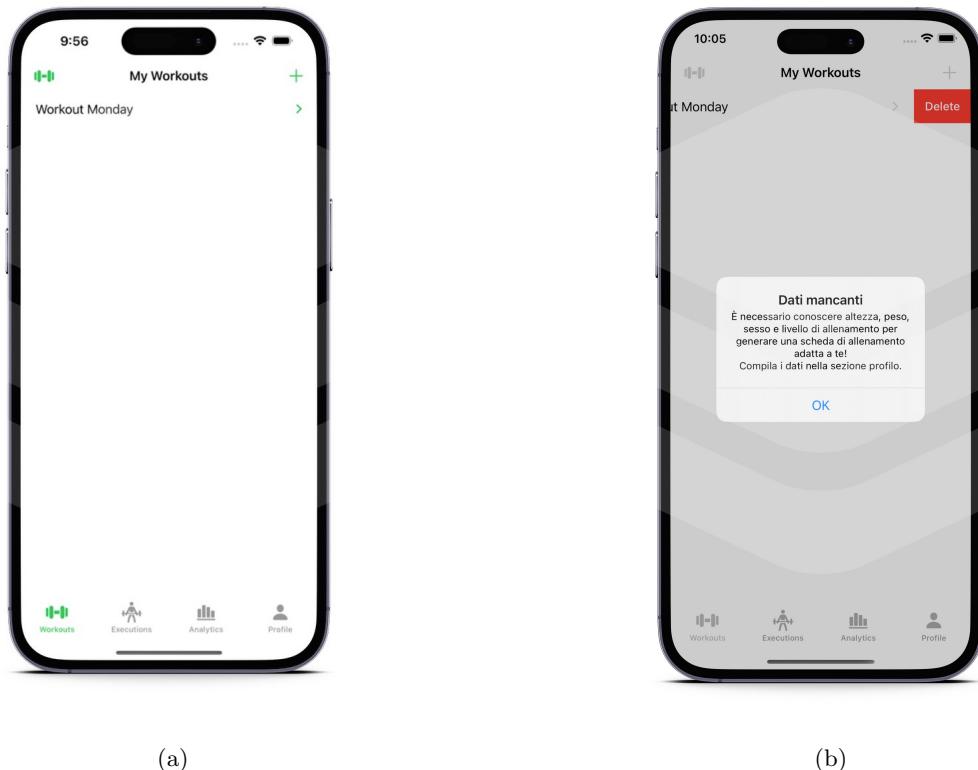
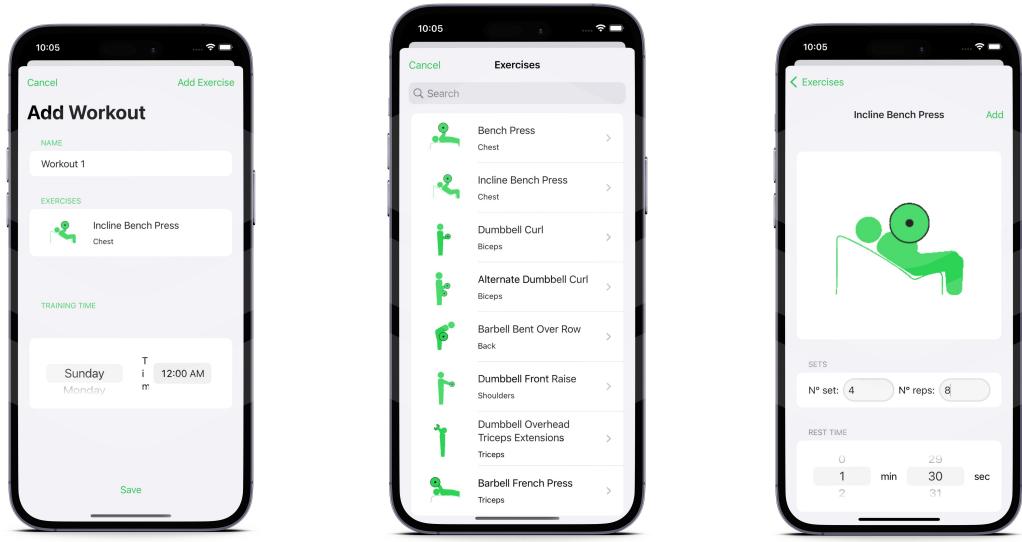


Figura 3.1: ContentView

3.1.1 Content View

Per accedere alla Content View (Figura 3.1), è necessario selezionare l'icona Workouts nella barra inferiore delle schede dell'applicazione. In questa sezione, è possibile aggiungere uno standard workout premendo il pulsante contrassegnato dall'icona del peso nell'angolo in alto a sinistra, a condizione che la sezione 'Profile' sia stata precedentemente compilata. La lista dei diversi workout è visualizzabile grazie alla Navigation View. Ogni workout viene rappresentato come un elemento della lista e può essere espanso tramite una Disclosure Group per mostrare i diversi esercizi che lo compongono. Cliccando su un singolo workout, si può accedere alla Workout View associata, che offre una visione dettagliata delle attività e degli esercizi da svolgere. Per aggiungere un nuovo workout, è possibile premere sull'icona '+' nell'angolo in alto a destra. Questa azione consente di accedere all'AddWorkout View, dove è possibile inserire i dettagli del nuovo workout da creare.



(a) AddWorkout View

(b) ExerciseProperty View

(c) AddExercise View

Figura 3.2: Aggiunta Workout

3.1.2 AddWorkout View

Dopo aver premuto il pulsante per l’aggiunta del Workout, appare l’AddWorkout View (Figura 3.2a). In tale view è possibile aggiungere un esercizio premendo il pulsante in alto a destra “Add Exercise”, è possibile tornare alla view precedente mediante il pulsante “Cancel” in alto a sinistra, si può aggiungere il nome del Workout (di default è una stringa vuota), vi è la possibilità di visualizzare gli esercizi aggiunti fino a quel momento, è disponibile la funzionalità di configurazione del momento in cui, sul telefono dell’utente(se ha dato il consenso) deve apparire la notifica ed infine è possibile, mediante il pulsante Save, salvare il workout ed essere rimandati alla schermata di Content View(Figura 3.1a).

3.1.3 ExercisePropertyView

Per aggiungere un esercizio è disponibile l’ExercisePropertyView (Figura 3.2b). Qui, mediante il filtro, è possibile eseguire la ricerca dei diversi esercizi per nome. Una volta che è stato trovato l’esercizio di interesse, cliccandoci, si viene rimandati all’AddExerciseView. Premendo invece il pulsante Cancel in alto a sinistra è possibile tornare alla view precedente.

3.1.4 AddExerciseView

Per personalizzare le proprietà del singolo esercizio è presente l’AddExercise View (Figura 3.2c). Con il pulsante Exercise in alto a sinistra è possibile tornare alla view precedente mentre il pulsante Add in alto a destra permette di inserire l’esercizio all’interno del workout corrente e di passare all’ AddWorkout View (Figura 3.2a). Al di sotto della gif che spiega il movimento del singolo esercizio, è possibile selezionare il numero di serie e di ripetizioni di quest’ultimo. Infine, vi è la possibilità di selezionare

il tempo di recupero che vi è tra una serie e l'altra, utile per il timer presente nella Play View.

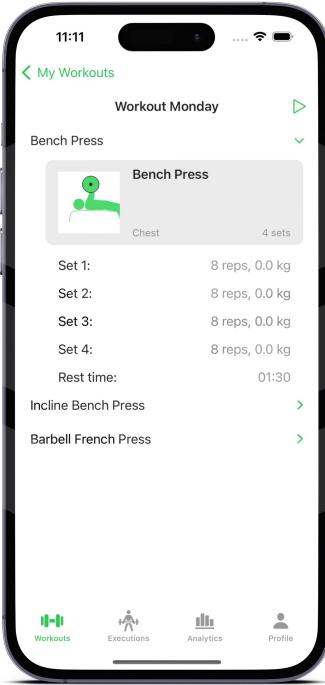


Figura 3.3: Workout View

3.1.5 Workout View

La vista Workout View (Figura 3.3) viene utilizzata per visualizzare un particolare allenamento. Contiene una lista di esercizi, ciascuno dei quali è una DisclosureGroup che mostra ulteriori dettagli nel momento in cui viene espanso. La view è incapsulata in una NavigationView per consentire la navigazione tra le diverse view. All'interno della DisclosureGroup sono visualizzati i dettagli del singolo esercizio che viene espanso, inclusi il numero di set, il numero di ripetizioni ed il peso usato precedentemente per quell'esercizio (permette di memorizzare lo storico). Viene utilizzato un ciclo “ForEach” per iterare sugli esercizi e visualizzarli nel modo corretto. È presente anche la struttura “ExerciseWorkoutCell” per rappresentare ogni singolo esercizio all'interno della lista dell'allenamento. Contiene una GIF che spiega il corretto movimento, il nome dell'esercizio, la parte coinvolta ed il numero di set e presenta uno sfondo grigio leggero. Infine, troviamo un pulsante in alto a sinistra per tornare alla view precedente, una tab bar al di sotto per muoversi tra le diverse sezione ed un pulsante con l'icona play in alto a destra per avviare la sessione di allenamento.

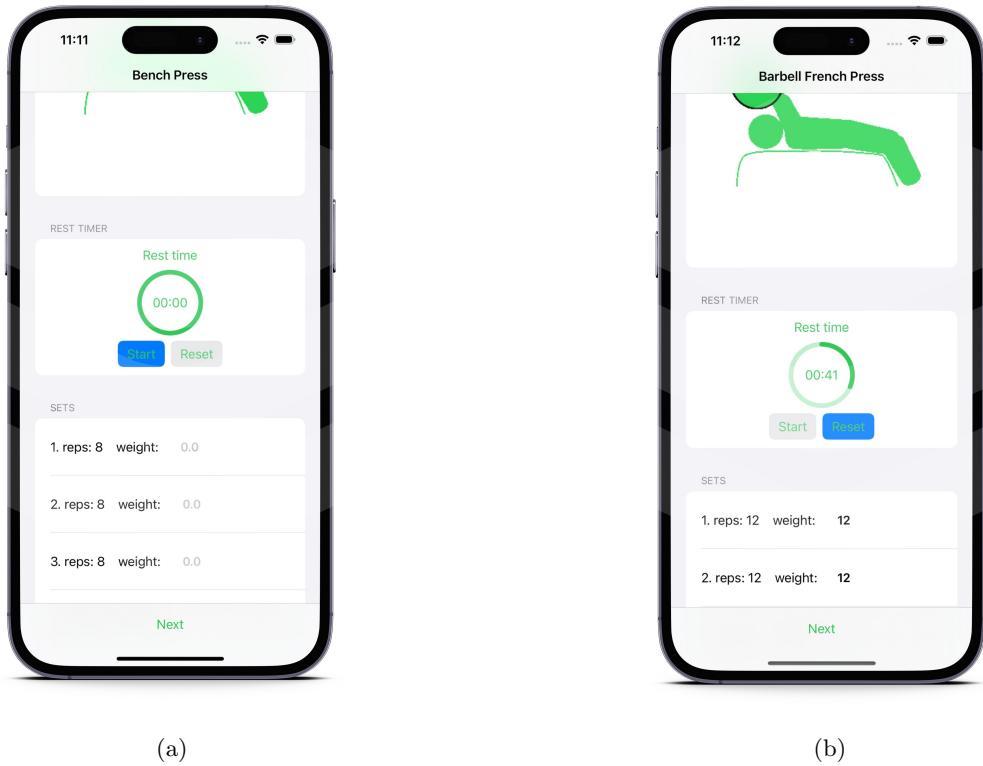


Figura 3.4: Play View

3.1.6 Play View

La Play View (Figura 3.4) permette di seguire l’utente durante la sessione di allenamento. La view è strutturata utilizzando un form che organizza gli elementi dell’interfaccia in una forma tabellare. All’interno vi è una sezione per la GIF che mostra il movimento dell’esercizio corrente, una sezione chiamata “Rest Time” per il timer utile per tenere traccia del tempo di recupero ed una sezione chiamata “Sets” per segnare il peso alzato ad ogni singola serie, utile per tener traccia dei miglioramenti nel tempo. È importante notare che il timer lo si ottiene utilizzando la “Timer View”. Una delle funzionalità più importanti della Play View è quella di poter andare all’esercizio successivo premendo il pulsante next posizionato in basso. Infatti, quando l’utente preme il pulsante next, viene richiamata la funzione “nextset ()” con il seguente codice:

```
func nextset() {
    for index in 0..<workoutStore.workouts[workoutindex]
        .exercises[exerciseindex]
        .sets.count {
            if newWeight[index] == "" {
                newWeight[index] = "0"
            }
            workoutStore.workouts[workoutindex].
                exercises[exerciseindex].sets[index].weight =
                Double(newWeight[index])!
        }
}
```

```

if workoutStore . workouts [ workoutindex ] .
exercises . count - 1 == exerciseindex {
    isFinalView . toggle ()
} else {
    isNextView . toggle ()
}
DataManager . shared .
saveWorkouts ( workoutStore . workouts )
}

```

In questo caso viene eseguito un ciclo for per iterare su ciascun set dell'esercizio corrente. All'interno del ciclo, viene controllato se l'utente ha inserito un peso per il set corrente. Nel caso in cui la stringa newWeight[index] è vuota, il peso viene messo a 0, mentre se è diversa da "", la stringa newWeight[index] viene convertita in un valore Double corrispondente. Dunque, viene aggiornato il peso del set corrente nell'oggetto workoutStore. In seguito, si esegue un controllo per verificare se l'esercizio corrente è l'ultimo del workout e, in caso affermativo, viene impostata la variabile isFinalView a true per mostrare la WorkoutAward View, mentre in caso negativo, viene impostata la variabile isNextView a true per passare all'esercizio successivo. Infine, viene chiamato il metodo saveWorkouts() dell'oggetto DataManager.shared per memorizzare gli allenamenti aggiornati.

Infine, è necessario anche analizzare la parte di codice della funzione "startTimer ()" presenta all'interno del file contenente TimerView:

```

private func startTimer () {
    isTimerRunning = true
    _ = Timer . scheduledTimer ( withTimeInterval : 1 , repeats :
true ) { timer in
        if isTimerRunning && timeRemaining > 0 {
            timeRemaining -= 1
        } else {
            timer . invalidate ()
            isTimerRunning = false
        }
    }
}

```

La funzione "startTimer ()" quando viene chiamata, viene impostata una variabile di stato isTimerRunning a true, indicando che il timer è in esecuzione. Successivamente, viene creato un oggetto Timer utilizzando il metodo scheduledTimer (withTimeInterval:repeats:execute:). Questo oggetto timer scatta a intervalli regolari, che nel nostro caso sono impostati a un secondo. Ogni volta che il timer scatta viene verificato se il timer è in esecuzione (isTimerRunning) e se il tempo rimanente (timeRemaining) è maggiore di zero. Se almeno una di queste due condizioni è verificata, il tempo viene decrementato di 1. In seguito, se il tempo rimanente non è zero o se il timer è stato fermato manualmente tramite il pulsante Reset, questo viene fermato invalidandolo mediante il metodo invalidate () chiamato sull'oggetto Timer. Inoltre, viene impostata la variabile di stato isTimerRunning su false per indicare che il timer è stato fermato.

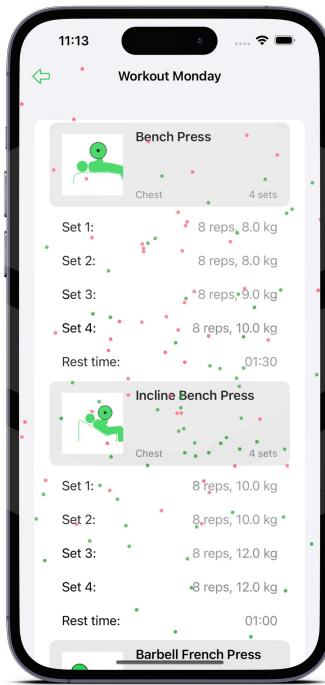
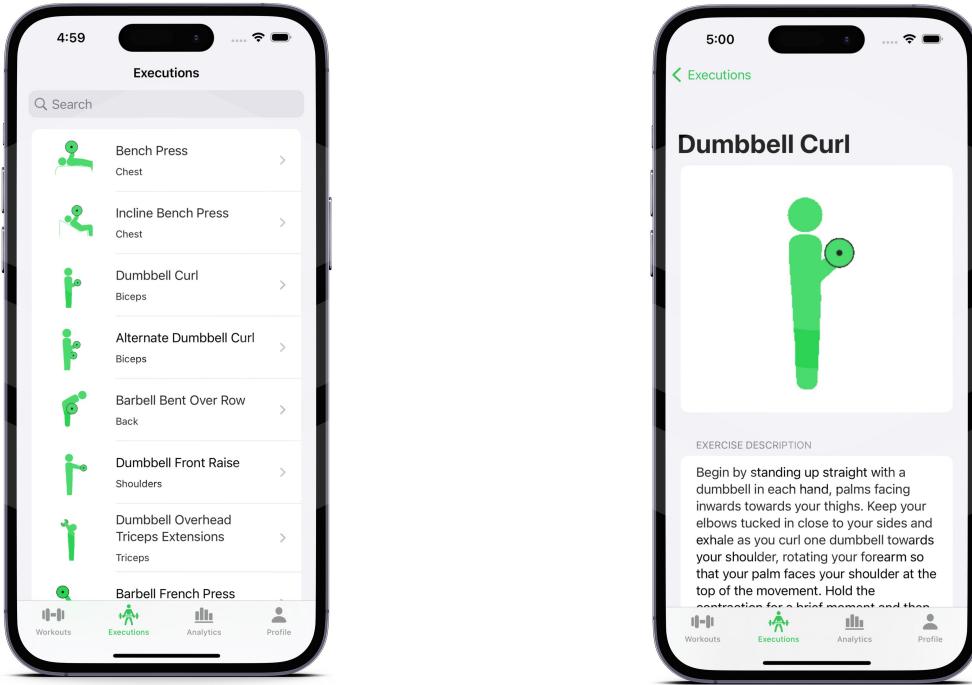


Figura 3.5: WorkoutAward View

3.1.7 WorkoutAward View

La WorkoutAward View (Figura 3.5) rappresenta la schermata per il resoconto del workout appena terminato. La view è organizzata utilizzando uno ZStack in modo tale da contenere le altre view sovrapponendole una sopra all'altra. All'interno della WorkoutAward View è presente una NavigationView che contiene una List. Quest'ultima viene popolata con gli esercizi presenti nell'oggetto Workout (passato come parametro). Per ogni esercizio, viene creata una cella di tipo ExerciseWorkoutCellAward. Quest'ultima è una view che rappresenta una singola cella d'allenamento. All'interno di quest'ultima vengono visualizzate informazioni sul singolo esercizio come l'immagine, il nome, la parte del corpo coinvolta ed il numero di set. La cella mostra uno sfondo chiaro con angoli arrotondati. La WorkoutAward View include due cerchi colorati (Circle) che rappresentano gli effetti visivi dei fuochi d'artificio. Questi sono posizionati in specifiche posizioni all'interno dello stack e sono modificati dall'effetto ParticlesModifier, ovvero un viewModifier che applica un effetto di particelle di fuochi d'artificio ai cerchi colorati. L'effetto crea un'animazione che fa muovere e sparire le particelle dei fuochi d'artificio. Infine, la struttura FireworkParticlesGeometryEffect serve ad implementare l'effetto di particelle. Tenendo conto del tempo, della velocità e della direzione desiderata, calcola la trasformazione delle particelle nel tempo.



(a) Execution View

(b) SingleExecution View

Figura 3.6: Exercise

3.1.8 Execution View

La Execution View (Figura 3.6a) è una schermata in cui gli utenti possono cercare un particolare esercizio all'interno di una lista ed è possibile arrivarci cliccando sull'icona execution posizionata sulla tab bar in basso. All'interno della view vi è una casella di ricerca(searchText) dove gli utenti possono inserire del testo per filtrare gli esercizi statici in base al nome. Nel momento in cui l'utente digita la prima lettera, la view si aggiorna automaticamente, mentre se tale casella è vuota, verranno mostrati tutti gli esercizi statici contenuti nell'oggetto ExerciseDB. Inoltre, la view utilizza la NavigationView per organizzare la schermata dove, all'interno, vi è una lista che mostra gli esercizi filtrati. Per ognuno di essi viene creata una cella personalizzata chiamata ExerciseStaticCell che mostra l'immagine dell'esercizio, il nome e la parte del corpo coinvolta. Infine, la cella è interattiva e consente agli utenti di selezionare un esercizio per visualizzarne i dettagli nel SingleExecutionView.

3.1.9 SngleExecution View

La SingleExecutionView(Figura 3.6b) mostra i dettagli di un singolo esercizio statico. Viene utilizzata una NavigationView per consentire la visualizzazione del titolo nella barra superiore. All'interno di tale view vi è un form per semplificare la creazione di moduli interattivi con sezioni, righe ed intestazioni. All'interno del form è presente una gif che spiega l'esecuzione corretta dell'esercizio ed una descrizione dettagliata dell'esercizio. Infine, in alto a sinistra è presente un pulsante per tornare alla view precedente.



Figura 3.7: Profile View

3.1.10 Profile View

La Profile View(Figura 3.7) è accessibile cliccando sull’icona “Profile” presente sulla tab bar e rappresenta la schermata del profilo dell’utente. La view utilizza l’ambiente condiviso (@EnvironmentObject) per accedere al WorkoutStore e all’oggetto Profilo, che gestiscono rispettivamente i dati degli allenamenti ed il profilo. Inoltre, viene creata un’istanza della classe Notifiche per la gestione delle notifiche. La view utilizza la NavigationView e, al suo interno vi è un form diviso in diverse sezioni che rappresentano le informazioni personali dell’utente, le misure antropometriche e le circonferenze. In particolare, nella sezione “Informazioni Personaliali” sono mostrati i campi per il nome, il sesso ed il livello di esperienza dell’utente, nella sezione “Misure antropometriche” vi sono i campi per l’altezza ed il peso che permette di salvare il peso dell’utente e mostrare il suo andamento su un grafico, mentre nella sezione “Circonferenze” vi sono i campi per le diverse circonferenze del corpo come collo, spalle, bicipite, tricipite avambraccio e polpaccio. Tutte le modifiche dei campi vengono salvate tramite il metodo DataManager.shared.saveProfile.

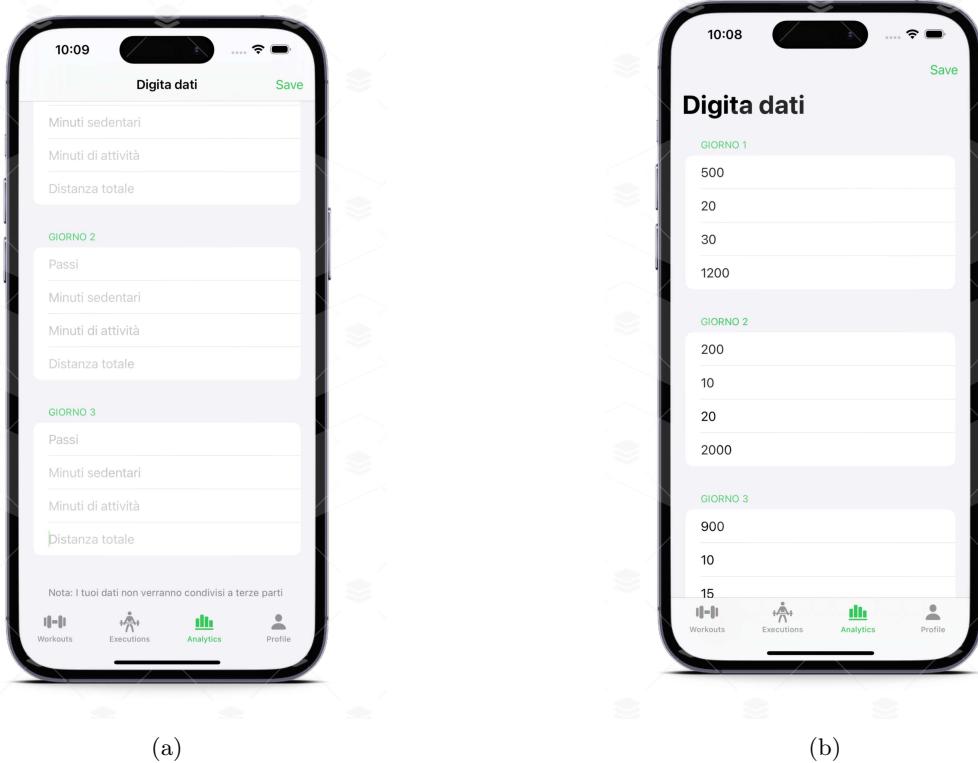


Figura 3.8: Analytics View

3.1.11 Analytics View

È possibile accedere all' Analytics View(Figura 3.8) cliccando sull'icona “Analytics” presente sulla tab bar. Una volta aperta questa view, viene richiesta all’utente l’autorizzazione per accedere ai dati di HealthKit. Se l’autorizzazione viene concessa, l’app esegue delle query per recuperare i dati relativi alle calorie bruciate dall’utente nei tre giorni precedenti al giorno corrente. L’interfaccia dell’Analytics View è suddivisa in diverse sezioni, ognuna delle quali rappresenta un giorno diverso (i tre giorni successivi alla data corrente). Per ciascun giorno, l’utente può inserire i dati riguardanti il numero di passi che prevede di fare, i minuti di inattività, i minuti di attività fisica e la distanza totale che prevede di percorrere. Questi dati vengono memorizzati all’interno delle corrispondenti variabili di stato. Quando l’utente ha compilato tutte le sezioni per i tre giorni (anche lasciandone alcune vuote se necessario), può premere il pulsante ”Save” situato in alto a destra per salvare i dati e visualizzare una schermata di previsione. Quando viene premuto il pulsante ”Save”, i dati inseriti dall’utente vengono elaborati utilizzando la funzione `passToButton()`, che crea una struttura (struct) contenente i dati inseriti dall’utente per i tre giorni. Successivamente, viene chiamata la funzione `prediction()`, che utilizza un modello di machine learning implementato con CoreML per eseguire la previsione delle calorie bruciate. Il modello riceve in input i dati di attività fisica inseriti dall’utente e restituisce una previsione delle calorie bruciate per ciascuno dei tre giorni. Le previsioni vengono quindi visualizzate all’interno di un’altra vista chiamata Prediction View.

Machine learning

Il machine learning [11] è un sottoinsieme dell'intelligenza artificiale il cui obiettivo è la creazione di sistemi che apprendono o migliorano le proprie performance in base ai dati che utilizzano. Nel machine learning vi sono diverse tipologie di problemi:

- **Regressione:** si basa sui principi fondamentali della fisica che aiutano a fare previsione sul futuro sulla base dei dati attuali. Permette anche di trovare la correlazione tra due variabili per definire una relazione causa-effetto. È possibile tracciare un grafico basato su queste variabili e fare una previsione di un output continuo, basato sulla variabile predittiva. Tuttavia, vi sono diverse forme di regressione, che vanno dal calcolo dei dati lineari a modelli più complessi come quelli polinomiali.
- **Classificazione:** è un metodo che assegna un valore di classe in base ai dati di input, fornendo previsioni definitive su una determinata azione. Tuttavia, la classificazione non si limita a due categorie, ma può coinvolgere più categorie a cause del calcolo della probabilità.
- **Clustering:** è una tecnica di machine learning non supervisionata che consente di raggruppare i dati simili per fare una previsione, invece di dati passati. L'algoritmo sfrutta caratteristiche visive o altre informazioni rilevanti per progettare una soluzione. Uno dei metodi più popolari per raggruppare gli input è il k-means che permette di specificare il numero di cluster desiderati (indicato come valore k) e categorizzare i dati in base a tale valore.
- **Riduzione della dimensionalità:** è il processo di riduzione del numero di variabili casuali durante la classificazione dei dati. Maggiore risulta essere il numero di variabili, più complessi diventano i risultati, rendendo difficile il loro consolidamento. La selezione e l'estrazione delle caratteristiche costituiscono la parte fondamentale della riduzione della dimensionalità nell'apprendimento automatico. Tale processo permette di eliminare variabili irrilevanti.

Una delle pratiche più comuni nel machine learning, soprattutto in ambiti come la classificazione e la regressione, è l'ensemble. Quest'ultima è una tecnica che permette di combinare i dati utilizzando variabili di previsione proveniente da vari modelli. Dunque, l'ensemble combina diversi modelli predittivi per formare un output predittivo altamente curato ed ottimizzato. Questo approccio viene utilizzato per prendere decisioni tenendo conto di diversi fattori. Un esempio tipo di algoritmo di ensemble learning è il Random Forest, che combina diversi alberi decisionali basati su differenti set di dati. Grazie a questa tecnica, l'output predittivo è di qualità superiore rispetto alle stime ottenute da un singolo albero decisionale. Infatti, un singolo algoritmo di apprendimento automatico può essere accurato in una specifica situazione, però produrre risultati errati in un diverso contesto. Per ridurre al minimo queste imprecisioni, gli scienziati dei dati sfruttano il metodo dell'ensemble per avere una previsione più corretta.

Nella teoria dell'ensemble, i weak learners sono modelli che possono essere usati come modelli costitutivi per la progettazione di modelli più complessi combinandoli tra loro. La maggior parte delle volte, da soli non si comportano in modo corretto a causa o di un bias elevato o perché hanno troppa varianza per essere robusti. Per impostare un metodo di ensemble, è necessario prima selezionare i modelli di base da aggregare.

Il più delle volte si utilizza un algoritmo di apprendimento di base in modo da avere dei weak learners omogenei che sono formati in modi diversi, così da ottenere un modello d'insieme chiamato omogeneo. Tuttavia, esistono anche altri metodi che utilizzano diversi tipi di algoritmi di apprendimento di base in modo tale da avere weak learners eterogenei che vengono combinati in un modello di insiemi eterogenei. Un punto fondamentale è che la scelta di weak learners deve essere coerente con il modo in cui aggregiamo questi modelli. Infatti, nel caso in cui si scelga un modello di base con basso bias ma alta varianza, bisogna scegliere un metodo di aggregazione che tende a ridurre la varianza mentre, in caso di scelta di un modello di base con bassa varianza ma alto bias, deve essere scelto un metodo di aggregazione che tende a ridurre il bias. Esistono tre diverse tecniche di combinazione dei weak learners:

- **Bagging**: spesso considera weak learners omogenei, li impara indipendentemente l'uno dall'altro in modo parallelo e li combina seguendo un processo di media deterministica.
- **Stacking**: spesso considera weak learners eterogenei, li impara in parallelo e li combina addestrando un meta-modello per produrre una previsione basata sulle diverse previsioni dei modelli weak.
- **Boosting**: nei metodi sequenziali i diversi weak models combinati non sono più adattati indipendentemente l'uno all'altro. L'idea, dunque, è di adattare tali modelli in modo iterativo così che l'addestramento del modello in un particolare passaggio dipenda dai modelli adattati precedentemente. Il boosting risulta essere l'approccio più famoso tra questi e permette di produrre un modello di ensemble che è meno distorto rispetto ai weak learners che lo compongono. I metodi di boosting hanno lo stesso scopo dei metodi di bagging, ovvero costruire una famiglia di modelli che vengono aggregati per ottenere uno strong learners con prestazioni migliori. Tuttavia, a differenza del bagging, il boosting è una tecnica che consiste nell'adattare in modo sequenziale più weak learners in modo adattivo, così che ogni modello nella sequenza viene adattato dando maggiore importanza alle osservazioni nel dataset che sono state gestite male dai modelli precedenti nella sequenza. Dunque, ogni nuovo modello concentra l'attenzione sulle osservazioni più difficili così da ottenere un modello finale con bias inferiore.

Dunque, si può affermare che il bagging è una tecnica che tende ad ottenere modelli di ensemble con meno varianza rispetto ai suoi componenti, mentre il boosting e lo stacking cercheranno di produrre modelli forti con meno bias rispetto ai loro componenti (anche se la varianza può a sua volta essere ridotta).

Boosting

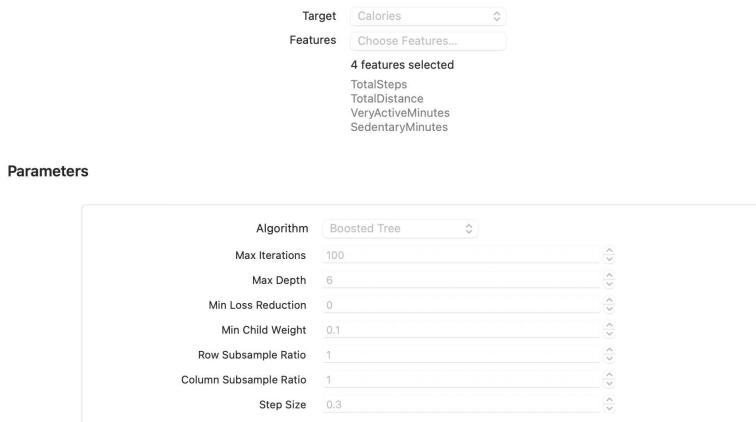


Figura 3.9: Modello

Tra le diverse tecniche messe a disposizione dal framework CoreML quali Random Forest, Decision Tree, Linear Regression e Boosted Tree, è stato scelto proprio quest'ultimo. Prima di trattare nel dettaglio tali tecniche, è utile introdurre il concetto degli Alberi decisionali [9].

L'apprendimento automatico sta diventando sempre più sofisticato ed offre un supporto prezioso nel processo decisionale. Un algoritmo utilizzato in questo contesto è l'albero decisionale, ovvero un layout di diversi risultati associati a una serie di scelte interconnesse. Le organizzazioni e gli individui possono utilizzare questo algoritmo per ponderare le loro azioni sulla base di diversi fattori come probabilità, benefici e costi. Un albero decisionale è composto da tre tipi di nodi: nodi causali, nodi decisionali e nodi finali. I nodi casuali rappresentano le probabilità di un determinato risultato e sono rappresentati da un cerchio. I nodi decisionali, rappresentati da una forma quadrata, indicano le scelte che devono essere prese. Infine, i nodi finali rappresentano i risultati di una decisione. L'obiettivo principale di un albero decisionale è sviluppare un modello predittivo che possa prevedere il valore o la classe di una variabile target attraverso la comprensione delle regole decisionali fondamentali derivate da dati di addestramento. In particolare, come tipologia di albero decisionale abbiamo l'albero decisionale a variabile continua e albero decisionale a variabile categorica. Infine, gli alberi decisionali presentano vantaggi e limitazioni. Tra i diversi punti di forza, vi è la chiara comprensione delle variabili critiche per la classificazione o previsione, la capacità di gestire variabili categoriche e continue, non è richiesto un calcolo eccessivo per l'esecuzione delle classificazioni e vengono generate regole facilmente comprensibili. D'altro canto, tra gli svantaggi troviamo che possono presentare errori, la formazione di un albero decisionale può essere molto costosa dal punto di vista dei calcoli e la potatura degli algoritmi è abbastanza costosa.

Boosted Tree

È utile ora trattare il concetto di Boosted Tree:
il boosting [10], come introdotto precedentemente, è un metodo utile per trasformare modelli deboli in forti. Nella pratica, ogni modello si adatta ad una versione modificata del set di dati originale. Un algoritmo di boosting molto usato è l'AdaBoost, che assegna un peso uguale a ciascuna osservazione durante il processo di addestramento dei modelli. Dopo l'addestramento del primo modello, si aumenta il peso di ogni osservazione che si trova difficile da classificare mentre diminuiscono i pesi per quelle in cui la classificazione non è complessa. L'idea è proprio quella di migliorare le previsioni del primo albero. Dunque, il nuovo modello consiste nell'albero uno più l'albero due. Vengono calcolati gli errori di classificazione dal nuovo modello di insieme così da sviluppare un terzo albero per la previsione dei residui modificati. Questa procedura viene eseguita per una particolare quantità di iterazioni (Figur3.9). Le previsioni finali del modello combinato sono un'aggregazione pesata delle previsioni dei modelli precedenti. Invece, il Gradient Boosting è un algoritmo di boosting che addestra sequenzialmente, in modo additivo e graduale diversi modelli, cercando di correggere le carenze dei modelli deboli precedenti. La differenza sostanziale tra Gradient Boosting e AdaBoost è nel modo in cui vengono determinate le carenze dei modelli deboli. AdaBoost utilizza i pesi sui dati, mentre il Gradient Boosting si basa sui gradienti delle funzioni di perdita. In particolare, la funzione di perdita misura la discrepanza tra le previsioni del modello e i dati reali e dipende dal tipo di problema in esame. Il Gradient Boosting è usato principalmente per ottimizzare diverse funzioni di costo. È utile soprattutto per risolvere problemi di regressione (come nel caso dell'applicazione in esame) e classificazione, migliorando l'accuratezza predittiva del modello. Il boosted tree è dunque una tecnica che usa come weak learners gli alberi decisionali e, nel caso di coreML, sfrutta proprio l'algoritmo del Gradient Boosting.



Figura 3.10: Set dati

Set di dati

Per creare, ottimizzare e valutare i modelli di predizione, è utile definire le tre tipologie di insiemi di dati:

- 1. Training set:** è un insieme di dati che viene utilizzato per addestrare un modello di Machine Learning. Il modello utilizza questi dati per “imparare” come associare input e output desiderati. Il training set rappresenta un sottoinsieme dei dati totali a disposizione, che viene utilizzato per la formazione del modello. Durante il processo di addestramento, il modello utilizza i dati del training set per adattare i suoi parametri in modo da ottenere la migliore performance possibile.
- 2. Test set:** è l’insieme di dati (diverso da quello del training set) con cui si valuta la capacità di predizione di un modello. In questa fase viene eseguito il testing error, calcolato come la media della differenza al quadrato tra le previsioni del modello e i valori osservati. Un errore di test basso indica che il modello ha una buona capacità di fare previsioni accurate su dati mai visti prima, mentre un errore di test alto indica una scarsa capacità di generalizzazione. Si deduce che il corrispondente del testing error in esame è comparabile con l’MSE calcolato sul training set. Generalmente l’MSE del training set è più basso di quello del test set, in quanto i vari parametri del modello sono stati stimati sui dati proprio del training, però si può osservare che una differenza minima può far dire che non siamo in overfitting, ovvero che non abbiamo iper-allenato il nostro modello sui dati iniziali. In caso contrario, ovvero iper addestrando il nostro modello, andiamo a ridurre il bias ma conseguentemente la varianza aumenterà. Ciò comporta che il nostro modello sarà ottimo per i dati utilizzati nel training ma pessimo per altri dati.
- 3. Validation set:** insieme di dati che viene utilizzato per valutare la prestazione del modello su dati diversi da quelli di addestramento, ma non viene utilizzato per prendere decisioni finali sul modello.

In figura 3.10 vengono mostrate le diverse percentuali di dati che possono essere assegnate ad ogni tipo di set appena citato.

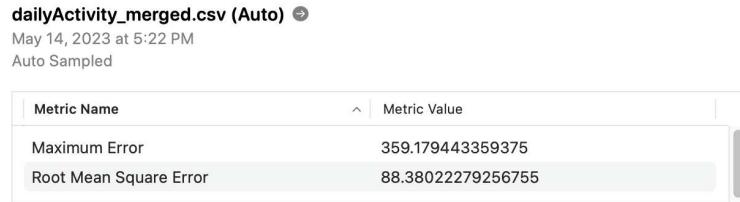


Figura 3.11: Valutazione

Valutazione

Il modello è stato allenato utilizzando un set di dati preso dal sito Kaggle, che comprende diverse features. Durante l'analisi del dataset è emerso che le variabili più significative per stimare le calorie bruciate sono TotalSteps, TotalDistance, VeryActiveMinutes, SedentaryMinutes ed ovviamente il target Calories (Figura 3.9).

A sostegno della scelta di utilizzare il Boosted Tree, è possibile analizzare la valutazione del modello fornito in output da CoreML (Figura 3.10). In questo caso, l'errore quadratico medio è di circa 88. Questo è un ottimo risultato e conferma che è stato corretto utilizzare la tecnica Boosted Tree .

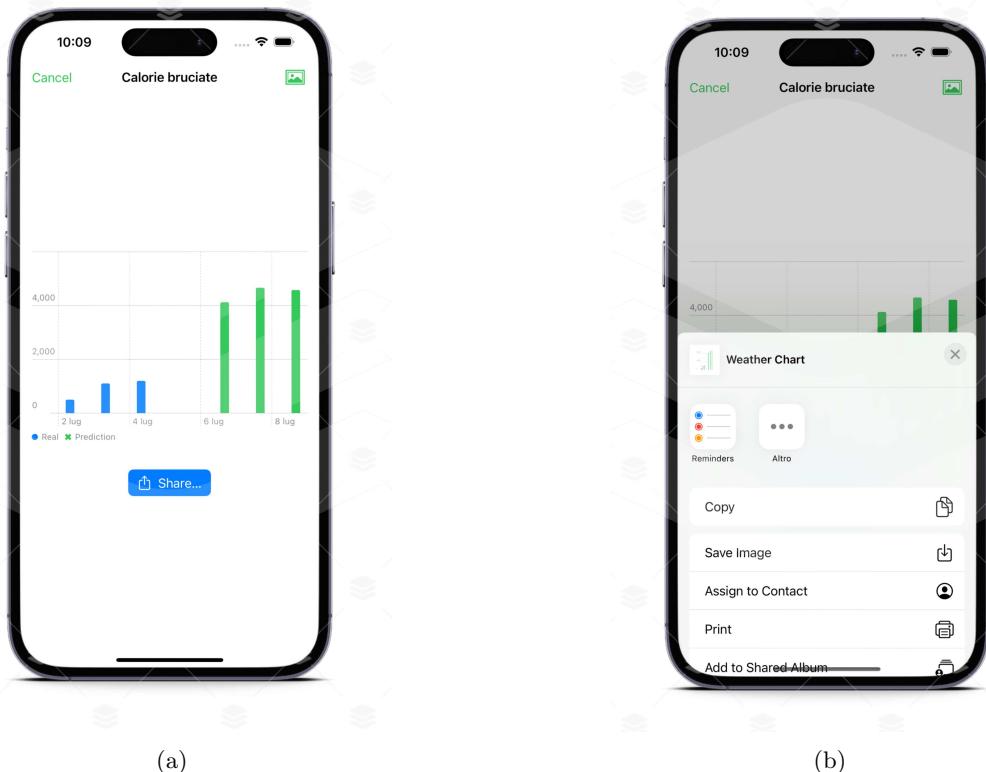


Figura 3.12: Prediction View

3.1.12 Prediction View

Si accede alla Prediction View (Figura 3.9) dall’ Analytics View premendo il pulsante save. Tale view confronta, mediante un grafico, le informazioni che l’utente ha bruciato nei tre giorni precedenti al giorno corrente e quelle che brucerà nei tre giorni successivi al giorno corrente. Per la realizzazione di tale funzionalità viene usato il componente Chart che permette di creare un grafico a barre verticali. Ogni barra rappresenta un giorno e mostra le informazioni delle calorie bruciate sull’asse Y. Al di sotto è presente anche un pulsante che, se premuto, permette di condividere l’immagine del grafico su ogni canale. Per realizzare ciò viene utilizzata la funzione ShareLink, parte del framework SwiftUI. Nel momento in cui il pulsante viene premuto, viene generata un’immagine del grafico usando la funzione generateSnapshot () e successivamente viene creato l’oggetto SharePreview che rappresenta l’anteprima dell’elemento da condividere. In seguito, viene mostrato all’utente un menu di condivisione ed una volta che l’utente ha scelto il metodo, l’app condivide l’immagine. Infine, è stata implementata un’altra funzionalità rappresentata da un pulsante situato nell’angolo in alto a destra. Questo pulsante consente di salvare l’immagine del grafico nella galleria fotografica dell’iPhone. Quando viene premuto, il telefono vibra e appare un alert che segnala il successo del salvataggio dell’immagine.

Nota

Desidero sottolineare che ho partecipato attivamente insieme ai miei colleghi alla realizzazione di dievrse componenti dell’applicazione, tra cui Content View, Add View, SingleExecution View, AddExercise View, Workout View, Execution View, Play View

e Timer View. Inoltre, ho sviluppato in modo indipendente le sezioni Analytics View, Prediction View e WorkoutAward View.

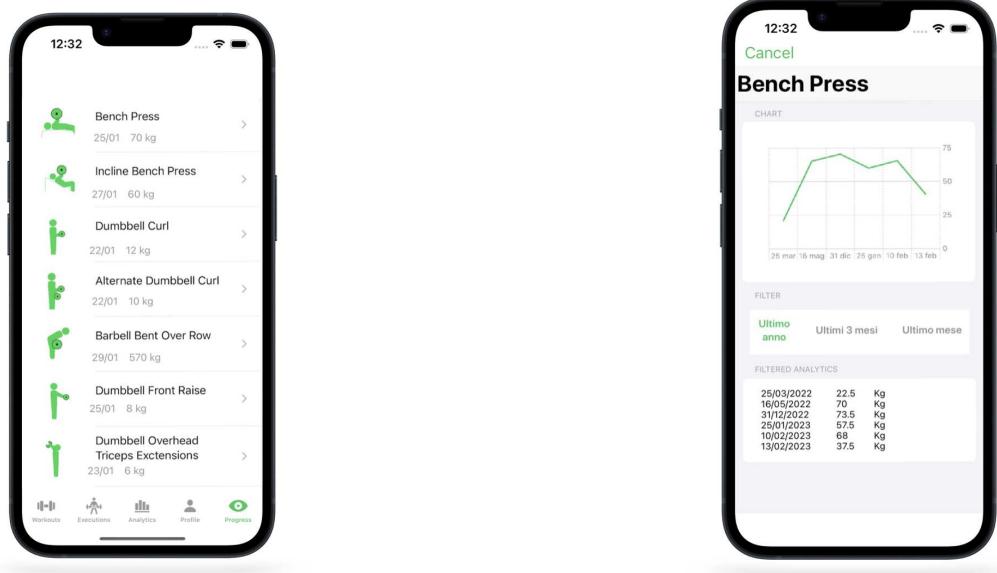
CAPITOLO 4

CONCLUSIONI E PROSPETTIVE FUTURE

«Per realizzare grandi cose, non dobbiamo solo agire, ma anche sognare; non solo progettare ma anche credere.»
- Anatole France

Il progetto Gymzone è un'applicazione con diverse funzionalità che mira a fornire un servizio di coaching virtuale durante gli allenamenti. L'app offre la possibilità di monitorare il tempo di recupero, il numero di set e ripetizioni durante un'esercizio, visualizzare gli esercizi correnti e fornire spiegazioni sul corretto movimento. Inoltre, l'app tiene traccia del dispendio calorico dell'utente nei tre giorni precedenti al giorno corrente e nei tre giorni successivi per aiutarlo nella gestione della dieta e fornire previsioni future.

Nonostante le numerose funzionalità già presenti, l'applicazione ha ampi margini di crescita e offre spunti per ulteriori implementazioni.



(a)

(b)

Figura 4.1: Progress View

In particolare, l’obiettivo è potenziare la funzionalità di coaching virtuale, introducendo una nuova sezione dedicata alla visualizzazione dei progressi con i pesi per ciascun esercizio. Come mostrato nella Figura 4.1, l’idea è quella di creare una nuova view (Figura 4.1a) a cui si può accedere cliccando sull’icona Progress nella tab bar posta in basso. Questa vista presenta un elenco di esercizi, e selezionandone uno, si accede a una vista dettagliata (Figura 4.1b) che mostra una cronologia dei pesi sollevati per quel particolare esercizio. Per rendere l’andamento dei progressi ancora più chiaro, è presente anche un grafico che illustra visivamente la crescita nel tempo. Con questa implementazione, gli utenti avranno la possibilità di monitorare facilmente i propri progressi nei pesi per ciascun esercizio, consentendo loro di valutare i miglioramenti nel corso del tempo.

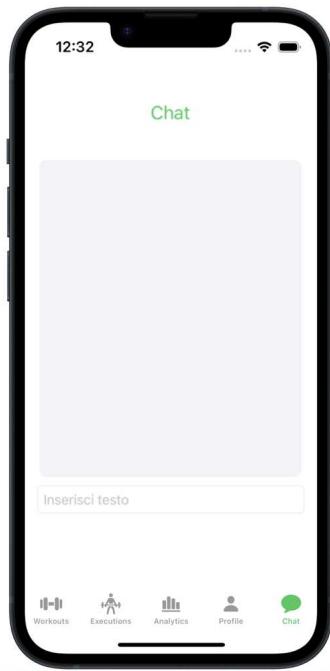


Figura 4.2: Chat View

Un ulteriore miglioramento riguarda l’implementazione di una funzione di chat all’interno dell’applicazione(Figura 4.2), che permette agli utenti di comunicare con un coach online selezionato tra quelli iscritti all’app. Questo servizio consente una comunicazione diretta tra l’utente e il coach scelto, fornendo un canale interattivo per porre domande, ricevere consigli e condividere informazioni pertinenti all’allenamento. Inoltre, ogni coach può ricevere una valutazione da parte degli utenti in base al servizio fornito, offrendo così un feedback importante per migliorare l’esperienza complessiva dell’app. Una volta che l’utente decide di affidarsi a un coach specifico, quest’ultimo ha la possibilità di creare una scheda personalizzata per l’utente. Questa scheda tiene conto delle caratteristiche individuali dell’utente, come il livello di esperienza in palestra, gli obiettivi personali e le preferenze di allenamento. Una volta creata, la scheda personalizzata viene caricata direttamente nel profilo dell’utente, consentendo così un accesso rapido e facile alle informazioni e alle istruzioni specifiche fornite dal coach. Questa implementazione aggiuntiva della funzione di chat e della scheda personalizzata contribuisce a migliorare l’interazione e l’esperienza degli utenti nell’applicazione Gymzone, offrendo un supporto più completo e mirato durante l’allenamento e promuovendo una comunicazione efficace tra utenti e coach.

Infine, un’altra possibile modifica da apportare per il lancio dell’app sul mercato riguarda l’adozione di un sistema di gestione di database cloud (DBMS) al posto dell’attuale utilizzo della classe DataManager per garantire la persistenza dei dati. Questo migliorerebbe l’affidabilità e la scalabilità dell’applicazione.

BIBLIOGRAFIA E SITOGRAFIA

- [1] Apple. *Charts*. URL: <https://developer.apple.com/documentation/charts>.
- [2] Apple. *CoreML Swift*. URL: <https://developer.apple.com/documentation/coreml>.
- [3] Apple. *Iphone 13*. URL: <https://www.apple.com/it/shop/buy-iphone/iphone-13>.
- [4] Apple. *MacBook Pro*. URL: https://support.apple.com/kb/SP824?locale=it_IT.
- [5] Michel Goossens, Frank Mittelbach **and** Alexander Samarin. *The LaTeX Companion*. Reading, MA: Addison-Wesley, 1993.
- [6] P. Foggia M. Vento. *Algoritmi e Strutture Dati*.
- [7] Carmen De Maio. *1stNanoChallengeSalerno*.
- [8] Marco Romano. *Human Interfaces Guidelines*.
- [9] Data Science. *Albero decisionale*. URL: <https://datascience.eu/it/matematica-e-statistica/albero-delle-decisioni/>.
- [10] Data Science. *Gradient Boosting*. URL: <https://datascience.eu/it/apprendimento-automatico/gradient-boosting-cosa-bisogna-sapere/>.
- [11] Data Science. *Tecniche di apprendimento*. URL: <https://datascience.eu/it/apprendimento-automatico/top-6-tecniche-di-apprendimento-a-macchina/>.

ELENCO DELLE FIGURE

1.1	Tecnologie e framework utilizzati per il progetto	6
2.1	Challenge Based Learning	13
2.2	Views esercizio	21
2.3	Views dettagli esercizio	22
3.1	ContentView	26
3.2	Aggiunta Workout	27
3.3	Workout View	28
3.4	Play View	29
3.5	WorkoutAward View	31
3.6	Exercise	32
3.7	Profile View	33
3.8	Analytics View	34
3.9	Modello	37
3.10	Set dati	39
3.11	Valutazione	40
3.12	Prediction View	41
4.1	Progress View	44
4.2	Chat View	45

RINGRAZIAMENTI

A conclusione di questo elaborato, desidero esprimere la mia sincera gratitudine a tutte le persone che hanno contribuito a questo lavoro di tesi.

Ringrazio il mio relatore Luca Greco per il grande supporto in questi mesi di lavoro.
Ringrazio di cuore i miei genitori, mia sorella Miriam e mio fratello Mario per avermi sempre sostenuto.

Ringrazio la mia ragazza Lucrezia per aver sempre creduto in me.
Infine, desidero ringraziare i miei compagni di corso , in particolare Mattia De Bartolomeis, Mario Della Corte e Christian De Angelis. La nostra collaborazione e i momenti condivisi durante lo studio e la realizzazione dei progetti sono stati di inestimabile valore.