

**Virtual Training Coach:  
un'applicazione multiplatforma per i tuoi  
allenamenti a casa**

Human Computer Interaction

**Niccolò Guiducci & Giulia Bertazzini**

Corso di Laurea in Ingegneria Informatica

Università degli Studi di Firenze

## Abstract

*Nel mondo del fitness, la perfetta tecnica di esecuzione di un esercizio è una regola inviolabile, non solo al fine di ottenere una maggiore efficacia, ma anche per evitare movimenti scorretti che possono portare a pesanti infortuni.*

*Virtual Training Coach è un'applicazione che si pone come obiettivo quello di fornire supporto a persone che vogliono eseguire esercizi fisici, per permettere loro di valutarne la correttezza e tener traccia dei propri progressi e miglioramenti.*

## 1 Introduzione

Con l'avvento del COVID-19, molte persone per continuare a mantenersi in forma sono state costrette a spostare gli abituali allenamenti dalla palestra all'interno delle proprie mura domestiche; da qui nasce l'idea che ha dato vita a **Virtual Training Coach**. La nostra applicazione è pensata per tutte quelle persone, neofiti o veterani del fitness, che desiderano eseguire un buon allenamento anche senza la presenza di un personal trainer in carne ed ossa.

Virtual Training Coach, infatti, permette all'utente di caricare o registrare un video riguardante lo svolgimento di uno degli esercizi proposti all'interno dell'applicazione; dopodiché, attraverso l'uso dell'intelligenza artificiale e di metriche appositamente studiate, lo svolgimento dell'esercizio viene confrontato con quello di un vero personal trainer. In questo modo l'applicazione può fornire all'utente una percentuale di riuscita dell'esercizio e mostrargli gli errori più marcati durante lo svolgimento, così da permettergli di migliorarsi di volta in volta.

L'intelligenza artificiale e le metriche precedentemente citate sono state sviluppate nel corso di un nostro precedente progetto universitario, Guiducci e Bertazzini (*3D Virtual Training Coach*), e successivamente adattate al contesto dell'applicazione.



**Figura 1:** Homepage

## 2 Sviluppo dell'idea

Una volta messo in chiaro l'obiettivo da raggiungere con lo sviluppo dell'applicazione si è reso necessario procedere con l'attività di needfinding, così da consentire, successivamente, lo sviluppo di un mockup.

### 2.1 Needfinding

Come suggerisce il nome, il **needfinding** è un'attività della Human Computer Interaction finalizzata alla conoscenza dei bisogni dell'utente; permette infatti di individuare necessità, obiettivi, frustrazioni e problemi di colui che utilizzerà il prodotto sviluppato. Così facendo è possibile poi costruire delle Personas, degli Scenarios e dei Requirements da cui estrarre una prima idea del software.

#### 2.1.1 Interviste

Inizialmente abbiamo intervistato un gruppo di persone rivolgendo loro delle specifiche domande mediante l'uso di un questionario.

In particolare, ci siamo rivolti a un gruppo di 44 persone, con un range di età piuttosto ampio, che va dai 20 ai 64 anni; di queste persone, l'84% pratica una qualche attività fisica, che per la maggior parte delle volte si svolge in palestra, mentre il restante 16% ha uno stile di vita molto sedentario. Inoltre è emerso che, con il sopraggiungere della pandemia, i più motivati a mantenersi in forma sono stati costretti ad allenarsi in casa propria, e circa il 35% di questi ha deciso di tornare in palestra non appena possibile, mentre il 65% ha preferito rimanere tra le mura domestiche, avendo trovato l'esperienza di allenamento indoor più stimolate, soddisfacente, comoda e coinvolgente. La scelta di coloro che hanno deciso di tornare ad allenarsi in palestra è stata per lo più dettata dal fatto che a casa non è comune possedere l'attrezzatura necessaria a svolgere alcuni esercizi e dal fatto che mancasse una figura che ne valutasse la loro corretta esecuzione.

Durante le sessioni di allenamento in casa, dall'intervista è emerso che:

- circa il 34% delle persone ha scelto i propri esercizi seguendo dei video tutorial su Youtube o altre piattaforme, valutandone in maniera autonoma la corretta esecuzione;
- il 16% ha contattato telefonicamente il proprio personal trainer, concordando una scheda di esercizi personale e in alcuni casi facendo anche una videochiamata per farsi correggere durante l'allenamento;
- il 18% ha svolto esercizi in maniera del tutto casuale, senza prestare particolare attenzione alla correttezza.

Infine, l'88% degli intervistati è interessato particolarmente a tenere traccia dei propri progressi e miglioramenti ottenuti nei vari allenamenti.

### 2.1.2 Personas

Dalle interviste sono state quindi create tre tipologie di persone:

- **GIULIA:** età 23, studentessa universitaria con una vita attiva e piena di impegni. Le piacciono molto gli sport, è un habitué della palestra e si allena almeno 4 volte a settimana. Con l'avvento della pandemia GIULIA è stata costretta ad allenarsi a casa, dove ha svolto i suoi esercizi in maniera costante e motivata chiedendo consiglio al suo personal trainer. Nonostante GIULIA abbia trovato l'allenamento casalingo inaspettatamente efficace, appena le palestre sono state riaperte ha deciso di tornare a riprendere i suoi normali allenamenti, a causa della mancanza di attrezzatura specifica e di un confronto diretto con il proprio personal trainer.



**Figura 2:** Giulia

- **NICCOLÒ:** età 25, neo-laureato, ha appena iniziato a lavorare. È pieno di energie ma a causa del nuovo lavoro non ha più molto tempo libero; oltre a qualche partita a tennis nei fine settimana, si allena quando può e va in palestra 1 o 2 volte a settimana. Con la pandemia anche NICCOLÒ, deciso a non demordere, ha cominciato a fare qualche esercizio casalingo, nonostante lo trovasse abbastanza noioso. Quando le palestre sono state riaperte, NICCOLÒ ha ritrovato la sua motivazione, ma per mancanza di tempo si è ritrovato ad andarci molto più sporadicamente, continuando invece a fare esercizi in casa.



**Figura 3:** Niccolò

- **GIANMARCO:** età 36, è un informatico a cui non piace molto l'attività fisica. Non si allena praticamente mai e ovviamente non ha mai messo piede in una palestra. Nonostante non si sia quasi reso conto del passaggio della pandemia, navigando in rete si è trovato sommerso di pubblicità sul fitness e incuriosito ha deciso di provare qualche home workout, naturalmente attraverso l'aiuto di una qualche app sullo smartphone e di qualche video su YouTube. Terminato lo stato d'emergenza GIANMARCO ci ha preso gusto e ogni tanto, anche se non troppo spesso e davanti al televisore, continua ad allenarsi.



**Figura 4:** Gianmarco

### 2.1.3 Scenari

Al fine di ricavare successivamente i requisiti necessari per l'applicazione, sono stati definiti i seguenti scenari:

- GIULIA, per continuare a mantenersi in forma durante il lockdown e prendersi una pausa dal troppo studio, decide di chiamare il suo personal trainer per farsi consigliare qualche esercizio da svolgere a casa propria. Essendo una veterana del fitness conosce a menadito tutti gli esercizi che le sono stati proposti e sa che la scheda deve essere eseguita in un certo tempo, ma vorrebbe comunque qualcuno che la correggesse in caso di errori. Decide quindi di riprendersi con lo smartphone e sottoporre i suoi video all'analisi di Virtual Training Coach.
- NICCOLÒ è riuscito ad andare in palestra ma è affollatissima e il personal trainer non può dedicargli troppe attenzioni, in quanto è occupato con altre persone; NICCOLÒ è impaziente di sapere come fare gli esercizi e decide di prendere il suo smartphone e utilizzare Virtual Training Coach come supporto nell'esecuzione di ogni esercizio.
- GIANMARCO, fortemente spinto dalla curiosità, decide di spostarsi dalla sua postazione di gaming e provare a fare qualche esercizio per perdere qualche chilo. Da buon informatico qual è, scarica l'applicazione Virtual Training Coach e, intrigato dal nome, decide di svolgere un mountain climber, senza però avere la minima idea di quale sia la sua utilità e di come si svolga. Una volta visto il video tutorial sull'applicazione e lette le informazioni correlate all'esercizio, decide di riprendersi mentre prova a fare l'esercizio, per poi farsi giudicare da VTC.
- GIANMARCO, che dopo mesi di utilizzo di Virtual Training Coach ha finalmente iniziato a prenderci gusto, decide di provare nuovamente a eseguire un esercizio di mountain climber, convinto di riuscire a far meglio delle prime volte. Terminato l'esercizio vuole quindi paragonare il risultato appena ottenuto con i risultati dei mesi precedenti e, in particolare, con quello del primo mountain climber svolto.

### 2.1.4 Requisiti

Dall'analisi degli scenari descritti nella precedente sezione 2.1.3, sono stati ricavati i principali requisiti dell'applicazione:

- l'utente deve avere la possibilità di **caricare un video** presente nella galleria del proprio telefono;
- l'utente deve poter **registrare un video** da analizzare direttamente dall'applicazione;
- l'applicazione deve mettere a disposizione dell'utente una **sezione di cronologia**, dove sono salvati tutti i risultati ottenuti;
- l'applicazione deve permettere di **filtrare i risultati** mostrati nella cronologia per data, percentuale di riuscita e tipologia dell'esercizio;
- l'utente deve poter **osservare un video tutorial** e **leggere informazioni** in merito a un esercizio mai svolto;
- l'applicazione deve **mostrare visivamente** all'utente dove sono stati compiuti gli errori.

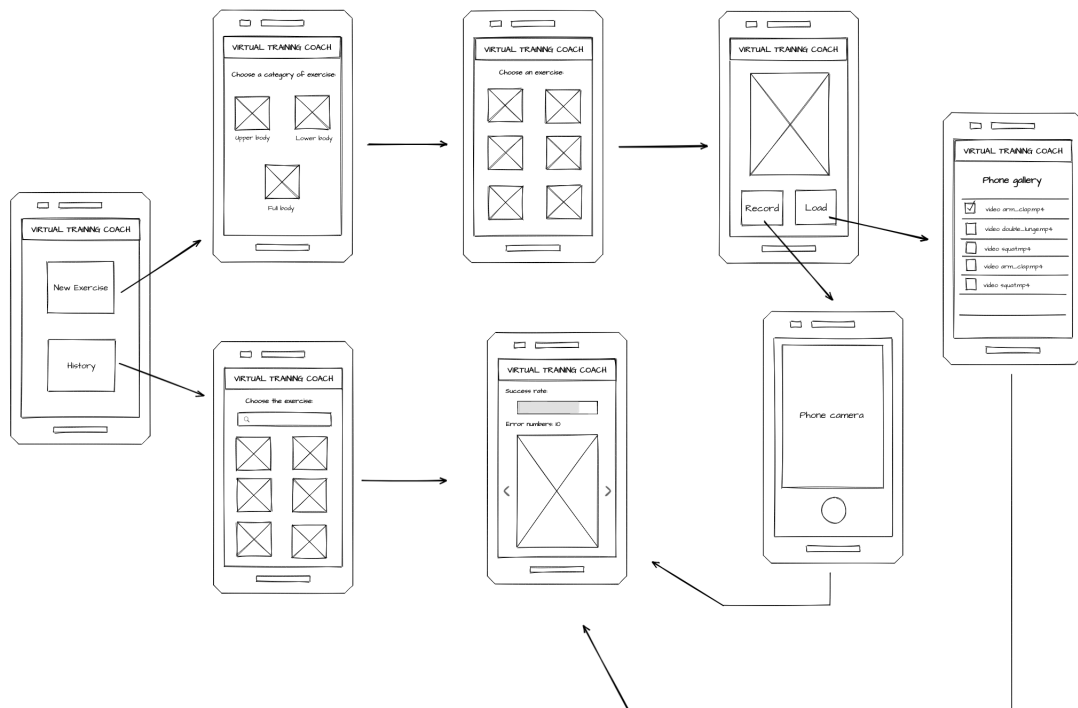


Figura 5: Mockup dell'applicazione.

## 2.2 Mockup

Terminata la fase di ricerca dei requisiti, siamo passati allo sviluppo del **mockup**, ovvero la realizzazione di uno schema a scopo illustrativo per semplificare il processo di sviluppo del software. In Figura 5 è riportato il primo modello sul quale ci siamo basati per implementare l'applicazione, al quale sono state poi apportate leggere modifiche in corso d'opera (come, ad esempio, la pagina di cronologia degli esercizi e l'aggiunta di un homepage [Figura 1] all'avvio dell'applicazione).

## 3 Struttura del progetto e strumenti utilizzati

Una volta delineata la strada da seguire e l'obiettivo da raggiungere, si è reso necessario suddividere il progetto in due parti principali: una parte di **backend**, che si occupa di gestire l'elaborazione e l'analisi dei video inviati dall'utente, e una parte di **frontend**, relativa all'interfaccia che l'applicazione dovrà assumere, in accordo a quanto descritto in precedenza (vedi 2.2).

### 3.1 Backend

Per quanto riguarda la parte di backend, i principali strumenti di sviluppo utilizzati sono stati l'IDE **PyCharm** e il **VTC Toolbox** da noi in precedenza sviluppato (vedi Guiducci e Bertazzini, *3D Virtual Training Coach*).

Per poter controllare la comunicazione tra l'applicazione e il VTC Toolbox, si è optato per

la creazione di un server basato su Web Socket, implementato mediante il modulo **Socket** della libreria standard di Python, grazie al quale è possibile gestire più richieste contemporaneamente.

Al fine di valutare la corretta esecuzione di un esercizio di fitness, lo strumento di VTC Toolbox permette l'analisi di sequenze di fotogrammi, che avviene grazie al confronto tra la posizione delle articolazioni degli skeleton di un personal trainer e di un generico utente; le articolazioni degli skeleton sono usate come punti di riferimento per rappresentare la posa di una persona nella sequenza temporale, così da individuare quali parti dell'esercizio sono state sbagliate. Il processo di analisi si svolge seguendo una sequenza di tre parti: un primo pre-processing dei fotogrammi (cropping e clipping), la successiva estrazione degli skeleton per ogni fotogramma grazie a una rete neurale, e, infine, il confronto tra le sequenze di skeleton del personal trainer e dell'utente attraverso una delle tre metriche definite. Nel contesto dell'applicazione VTC si è scelto di usare la metrica Combined Distance, che combina distanza tra angoli e distanza euclidea, in quanto è quella che ha prodotto i risultati migliori nello studio Guiducci e Bertazzini, *3D Virtual Training Coach*.

## 3.2 Frontend

Per lo sviluppo della parte di frontend, si è scelto di utilizzare il framework **Cordova**, il quale comprende l'uso di **HTML**, **CSS** e **JavaScript**. L'utilizzo di Cordova ha semplificato l'implementazione dell'applicazione per le varie piattaforme mobile (es. Android, iOS, ecc.) fornendo delle API universali che permettono l'accesso alla fotocamera, l'accesso in scrittura e lettura della memoria, plugin APIs e altro ancora, attraverso comandi indipendenti dal sistema operativo in uso. La stesura del codice di frontend è avvenuta sull'IDE **WebStorm** mentre, per il runtime testing sono stati utilizzati l'IDE **Android Studio**, **Android SDK** e un emulatore Android.

# 4 Implementazione

Per quanto riguarda l'implementazione, l'intero progetto è suddivisibile in due macro-aree: la parte di WebApp, le cui funzionalità sono scritte principalmente in JavaScript, e la parte Server, in Python.

## 4.1 WebApp

All'interno della parte di WebApp si possono individuare alcuni moduli principali, ognuno dei quali implementa una specifica funzionalità:

### 4.1.1 ClientVTC

In questo modulo sono contenute tutte quelle funzioni che permettono la gestione della connessione con il server e quelle adibite al pre-processing dei video e delle immagini da inviare al server. Più in particolare, sono presenti:

- una serie di funzioni event-driven dedicate alla gestione della connessione e della comunicazione, secondo lo standard Web Socket, tra l'applicazione (Client) e il VTC

Toolbox (Server); queste funzioni, oltre a fornire un controllo sulla connessione, forniscono anche un feedback all'utente nel caso in cui il collegamento al server non fosse avvenuto con successo;

- un `WebSocket` object per la costituzione del collegamento tra applicazione e server;
- la funzione `recordVideo()` che permette la registrazione di un video attraverso l'uso della fotocamera dello smartphone;
- la funzione `loadVideo()` che permette, invece, il caricamento di un video direttamente dalla memoria interna dello smartphone;
- la funzione `extractFramesFromVideo()` che si occupa del pre-processing dei video e delle immagini, ovvero dell'estrazione dei singoli fotogrammi dal video che, successivamente, vengono scalati e ridotti di qualità per poi essere inviati al server;
- le funzioni `sendImageFromStorage()` e `sendImageFromURI()` che si occupano della scomposizione in pacchetti e dell'invio di un'immagine al server, rispettivamente operando direttamente dalla memoria del dispositivo o utilizzando un URI.

#### 4.1.2 FileHandler

Come suggerisce il nome, tale modulo prevede varie funzioni per la gestione dei file:

- la funzione `createDatabaseStructure()` che viene eseguita solamente al primo avvio dell'applicazione per la creazione della struttura del database locale all'interno della memoria di archiviazione del dispositivo, nel quale saranno contenuti tutti i dati riguardanti i risultati degli esercizi e i relativi fotogrammi di errore;
- la funzione `writeResults()` che si occupa della scrittura nel database locale dei risultati, che verranno salvati in formato txt;
- la funzione `readResults()` per la lettura dei risultati dal database locale;
- la funzione `deleteResult()` per la cancellazione dei risultati dal database locale;
- tre funzioni per leggere l'intero database, restituendo però soltanto determinati risultati sulla base di un filtro selezionato dall'utente (vedi 5.2);
- la funzione `storeErrorFrame()` per il salvataggio nel database dei fotogrammi di errore;
- la funzione `storeTemporaryFrame()` per la scrittura in memoria temporanea dei fotogrammi inviati al server. Questo salvataggio di dati è necessario per assicurare la sincronizzazione dei fotogrammi di errore tra applicazione e server.

#### 4.1.3 HistoryFunctions

In questo modulo sono contenuti tutti quei metodi necessari alla creazione, all'ordinamento, alla visualizzazione e, in generale, alla gestione della cronologia degli esercizi svolti dall'utente.



#### 4.1.4 ResultsLoader

Questo modulo, infine, contiene l'insieme delle funzioni necessarie alla lettura e visualizzazione dei risultati; contiene infatti una funzione per mostrare graficamente i risultati all'utente, dopo che questi sono stati letti dalla memoria, e una funzione per recuperare i fotogrammi contenenti gli errori dal database e segnare mediante delle "X" le articolazioni più sbagliate.

### 4.2 Server

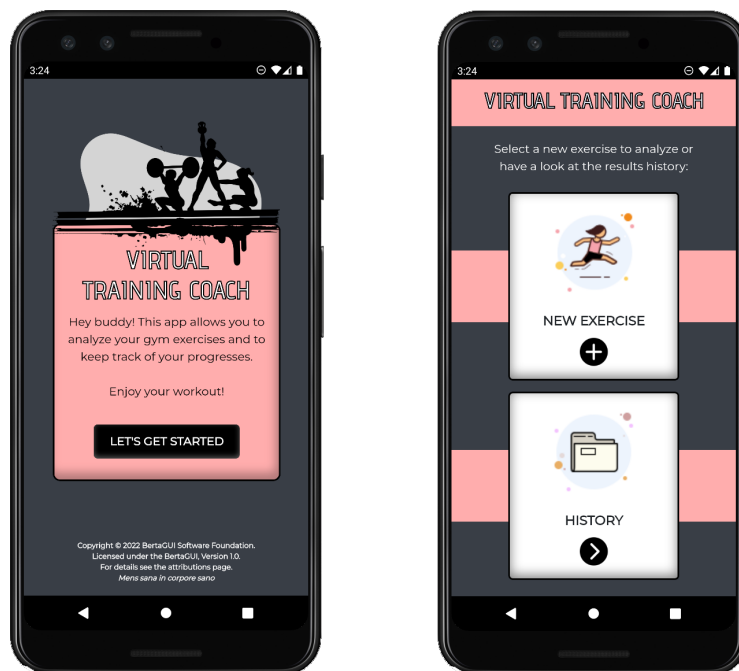
Per l'implementazione della parte Server si è resa necessaria principalmente l'integrazione del VTC Toolbox all'interno di un ambiente che potesse fornire le funzionalità che un server dovrebbe avere. Dovendo interagire con un client scritto in Javascript, non è stato possibile comunicare utilizzando direttamente le librerie socket standard; si è reso quindi necessario re-implementare lo standard di comunicazione WebSocket, gestendo direttamente i byte dei pacchetti in arrivo.

Le funzioni che permettono la gestione ed il mantenimento del server si trovano all'interno del modulo **server.py** e le principali sono:

- la funzione **start()** che inizializza i parametri del server (IP, porta, ecc.) e lo mette in ascolto per eventuali connessioni; nel caso in cui arrivi una richiesta di nuova connessione, la funzione si occupa di lanciare un nuovo thread per la gestione parallela della comunicazione con il nuovo client;
- **communication\_pipeline()** e **handshake()**, all'avvio di una nuova connessione, si occupano di gestire la fase di handshake tra server e client, per permettere l'instaurazione della comunicazione WebSocket; nel caso in cui la procedura non vada a buon fine, la connessione viene rifiutata e chiusa;
- la funzione **handle\_client()** si occupa di mantenere attiva la connessione dopo che questa è stata instaurata, gestendo quindi le comunicazioni ed i comandi provenienti dal client;
- la funzione **decode\_first\_frame()** permette la decodifica del primo pacchetto di un messaggio: le informazioni che si ricavano da questa operazione (chiave di decrittazione, lunghezza, ecc.) permettono il proseguimento della ricezione del messaggio;
- **receive\_message()** e **receive\_image()** sono le funzioni adibite alla ricezione, rispettivamente, di un messaggio generico e di un messaggio che contiene la codifica di un'immagine;
- **send\_message()** è la funzione che si utilizza quando si vuole inviare un messaggio al client; prima di essere spedito, il messaggio viene incapsulato nell'header standard di WebSocket.

## 5 Interfacce e funzionalità

La prima schermata che si presenta all'utente all'avvio dell'applicazione, inizialmente non prevista nel mockup (5), è una pagina contenente una breve descrizione dell'applicazione, come si vede in Figura 6. Premendo sul bottone con etichetta “**LET’S GET STAR-**



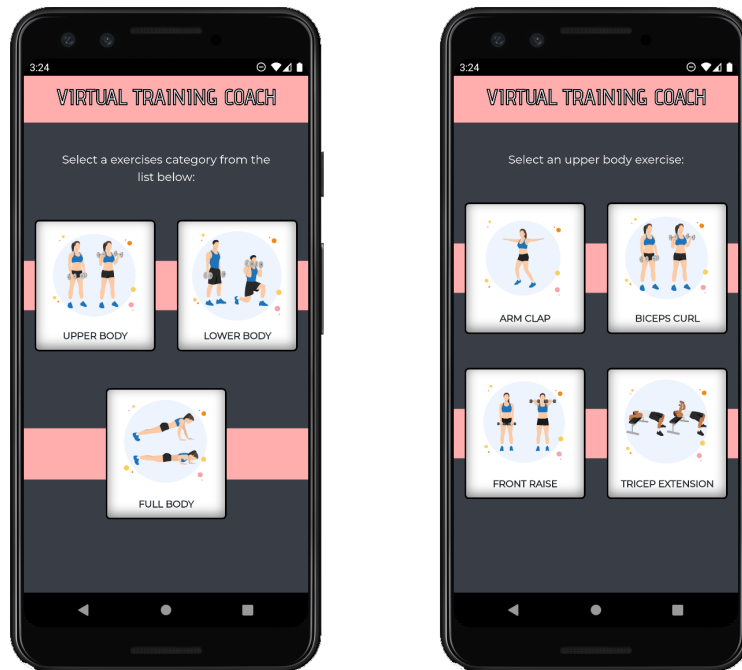
**Figura 6:** Interfaccia corrispondente alla homepage (sinistra) e alla successiva pagina (destra)‘.

**TED**”, l’utente viene portato alla pagina successiva, a destra nella Figura 6, in cui l’utente può decidere se far analizzare un nuovo esercizio o accedere alla cronologia dei risultati precedentemente ottenuti.

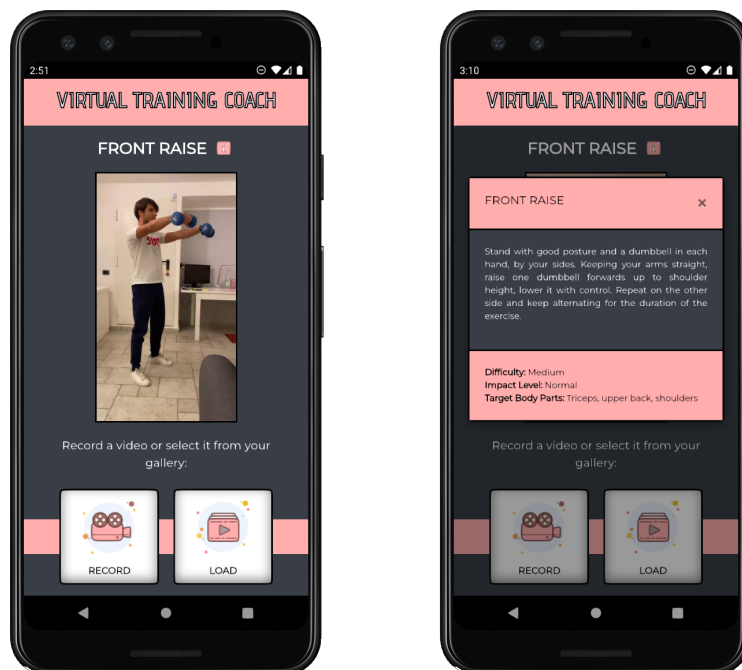
### 5.1 Analisi di un nuovo esercizio

Nel caso in cui si decida di cliccare sul bottone con etichetta “**NEW EXERCISE**”, l’utente verrà guidato dall’applicazione nella scelta della tipologia di esercizio da analizzare. Ai fini di semplificarne la ricerca, l’applicazione prevede una suddivisione degli esercizi in tre categorie, quali **UPPER BODY** (contenente gli esercizi che coinvolgono prevalentemente la parte superiore del corpo), **LOWER BODY** (per gli esercizi che interessano principalmente la parte inferiore del corpo) e **FULL BODY** (per gli esercizi in cui sono coinvolte tutte le parti del corpo, sia superiori che inferiori). Tale suddivisione, inoltre, rispecchia quella prevista dal VTC Toolbox, in quanto, a seconda della categoria di esercizio, vengono considerate soltanto le opportune articolazioni dello skeleton per effettuare i confronti tra trainer e utente.

Come si può vedere dalla Figura 7, una volta scelta la categoria (nell’esempio in questione è stata selezionata la categoria **UPPER BODY**), l’utente viene portato alla pagina successiva per selezionare l’esercizio da analizzare.



**Figura 7:** Interfaccia relativa alla scelta della categoria dell'esercizio (sinistra) e alla scelta della tipologia di esercizio appartenente alla corrispondente categoria, in questo caso UPPER BODY (destra).



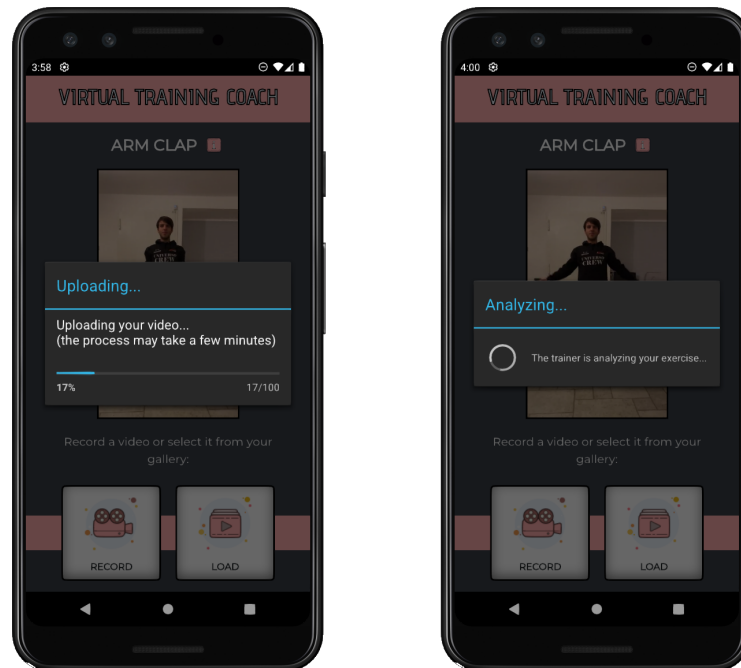
**Figura 8:** Interfacce raffiguranti la pagina dedicata al caricamento del video (a sinistra) e popup di informazioni sull'esercizio.

Cliccando su uno degli esercizi mostrati, per esempio FRONT RAISE, l'applicazione mostrerà all'utente la schermata riportata a sinistra in Figura 8. Oltre a un video dimostrativo

dell'esercizio (in loop e autoplay), in alto è presente un pulsante informazioni: cliccandovi si aprirà un popup (a destra nella figura) contenente una panoramica dell'esercizio, composta da una breve spiegazione sul corretto svolgimento e delle indicazioni riguardo la difficoltà, il livello di impatto e le parti del corpo prevalentemente coinvolte.

In fondo alla schermata sono presenti due pulsanti, **RECORD** e **LOAD**, che implementano rispettivamente la funzionalità di registrazione di un video direttamente dall'applicazione e di caricamento di un video dalla galleria del telefono.

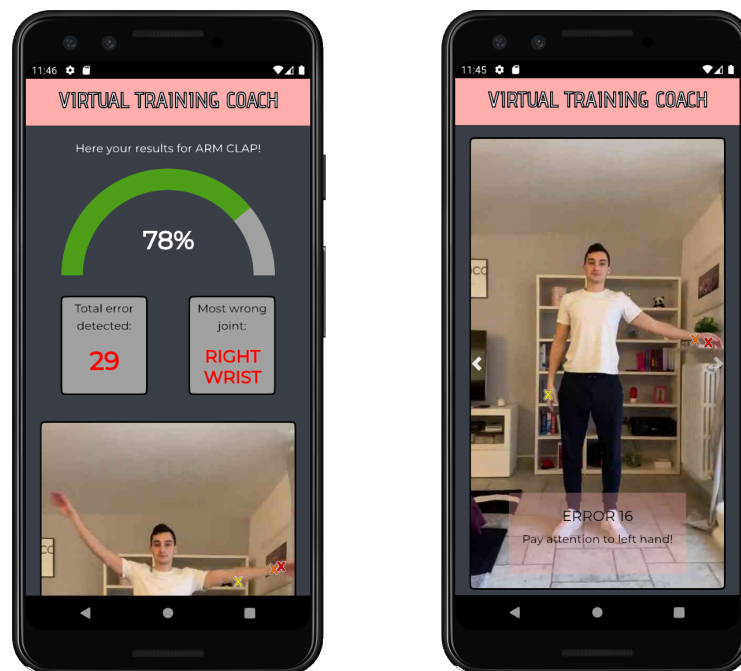
Indipendentemente dalla scelta compiuta dall'utente, una volta che l'applicazione ha otte-



**Figura 9:** Popup per lo stato di avanzamento del caricamento e dell'analisi del video caricato.

nuto l'accesso al video, verranno mostrati due popup in sequenza: il primo mostra lo stato di avanzamento dell'uploading del video sul server (a sinistra in Figura 9), una volta terminato, lascerà spazio al secondo che, invece, notifica all'utente che VTC sta analizzando l'esercizio, come mostrato in Figura 9.

Terminato poi il caricamento e l'analisi del video, l'utente viene automaticamente reindirizzato alla pagina dei risultati (Figura 10). In questa schermata l'applicazione mostra la percentuale di riuscita dell'esercizio, accompagnata dal numero totale di errori e dall'articolazione che è stata maggiormente sbagliata nel corso dello svolgimento delle varie ripetizioni. Inoltre, ai fini di aiutare l'utente a comprendere quali sono stati gli errori compiuti, vengono presentati i fotogrammi relativi agli errori individuati all'interno di uno slideshow, dove per ogni fotogramma vengono indicate all'utente le articolazioni a cui prestare maggiormente attenzione.



**Figura 10:** Schermata dei risultati ottenuti per un esercizio di arm clap.

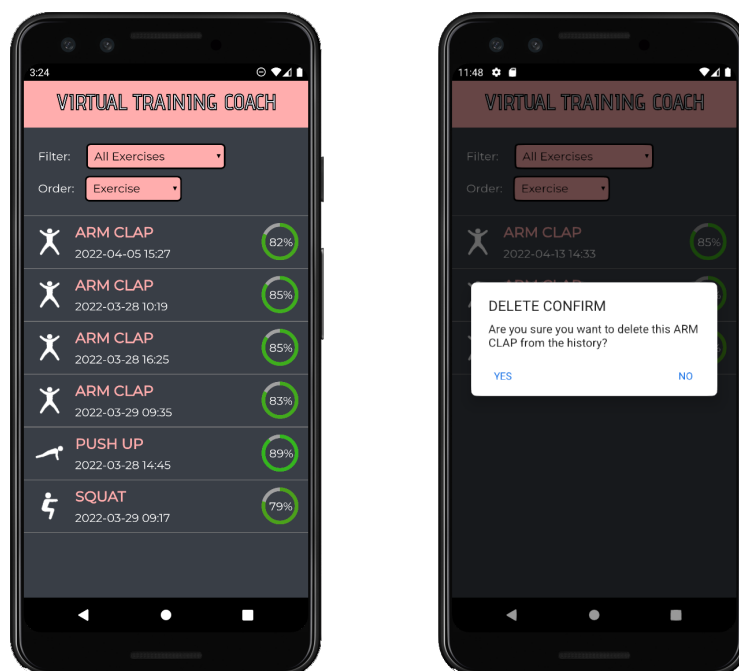
## 5.2 Cronologia dei risultati

Se si decide invece di aprire la cronologia, l'applicazione mostrerà all'utente la schermata in Figura 11. In questa pagina sono riportati, sotto forma di lista, tutti i risultati ottenuti dall'utente dal primo utilizzo dell'applicazione.

Originariamente, nella fase di mockup, avevamo previsto un'interfaccia di tale pagina simile a quella per la scelta dell'esercizio (Figura 7); tuttavia, durante lo sviluppo dell'applicazione, abbiamo ritenuto più opportuna un'interfaccia come quella riportata in Figura 11, in quanto si presuppone che, a lungo andare, l'utente avrà innumerevoli risultati all'interno della propria cronologia. Per questo motivo si è ritenuto migliore rappresentare i risultati sotto forma di lista, dove per ogni elemento della lista è indicato il nome dell'esercizio, la data in cui è stato svolto e la percentuale di successo. Nella parte alta della pagina sono presenti due pulsanti, che implementano due diverse funzionalità:

- **Filter** consente di scegliere la tipologia di filtraggio dei risultati; è possibile infatti scegliere di visualizzare i risultati ottenuti per una tipologia di esercizio (per esempio mostrando soltanto gli esercizi di arm clap) oppure per data (mostrando soltanto gli esercizi svolti in un giorno specifico).
- **Order** permette invece di ordinare i risultati sulla base del nome dell'esercizio, della data o della percentuale di successo.

Infine un'ulteriore funzionalità è quella di eliminare un risultato dalla cronologia: tenendo premuto su un elemento della lista compare l'icona di un cestino e cliccandovi si permette all'utente di eliminare un risultato, dopo aver richiesto conferma attraverso un message box.



**Figura 11:** Interfaccia della cronologia dei risultati.

## 6 Test di usabilità

Al fine di osservare al meglio l'interazione tra l'utente e applicazione, sono stati infine svolti dei test di usabilità. Tali test vengono svolti assegnando all'utente uno o più attività da svolgere e analizzando il suo comportamento nel portarli a termine.

Il test è stato strutturato come un ibrido tra un test "al naturale" ed un test guidato da un moderatore: all'utente viene fornito un dispositivo in cui si trova un'installazione dell'applicazione dotata già dei dati riguardanti alcune sessioni di allenamento; in questo modo è come se l'utente avesse già usufruito dell'applicazione e può dunque esplorare tutte le funzionalità che VTC ha da offrire. Dopo una breve spiegazione dello scenario per cui l'utente si ritrova per le mani VTC e una piccolissima presentazione delle principali funzionalità, il soggetto del test è lasciato libero di esplorare l'applicazione. Passato qualche minuto in cui il moderatore ha osservato l'utente, questo procede a supportare il soggetto ponendogli dei quesiti come: "E se fossi interessato a capire se sono migliorato nell'arco di un mese?", così da capire, oltre la qualità della funzionalità stessa, anche la sua rilevabilità.

Una volta conclusa la parte operativa del test, al soggetto è stato sottoposto un questionario formato da domande a risposta multipla e SEQs, ovvero delle "single ease questions" riguardanti la difficoltà di compiere un determinato compito o la difficoltà di utilizzo di una funzionalità. SEQs sono domande a cui si può rispondere con un numero intero in un intervallo tra 1 e 7 e che vanno a misurare l'esperienza di utilizzo (*user experience*).

I soggetti che hanno preso parte al test sono stati 7 in totale e appartengono a varie fasce di età (dai 15 ai 65 anni); inoltre, la metà dei partecipanti è stata scelta tra i soggetti intervistati nella fase preliminare di needfinding (vedi sezione 2.1.1).

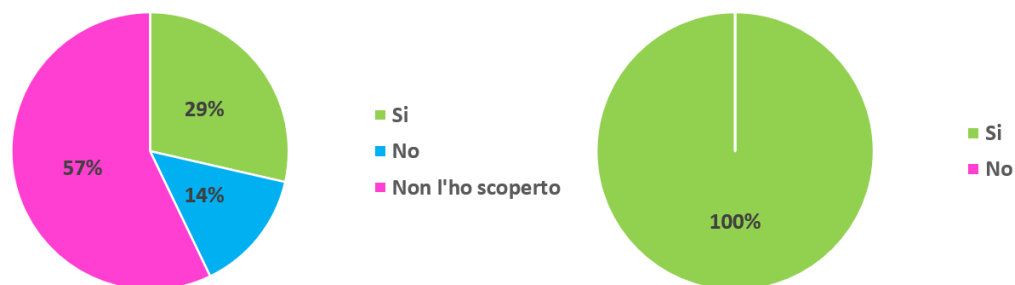
## 6.1 Risultati ottenuti dai test

Nella Tabella 1 sono riportati i risultati ottenuti dai test di usabilità precedentemente descritti.

N	Domanda	$\mu$	$\sigma$
1	Hai trovato semplice l'uso dell'applicazione?	4.67	1.21
2	Lo scopo dell'applicazione è risultato chiaro fin dall'inizio?	6	1
3	Reputi facilmente leggibile la schermata della cronologia?	5.86	1.68
4	È stato difficile usare i filtri e/o l'ordinamento per la visualizzazione delle sessioni di allenamento?	4.86	1.57
5	Le informazioni relative a una sessione di allenamento sono di facile lettura?	5.57	1.13
6	Reputi semplice la navigazione tra le varie categorie di esercizi?	5.86	0.69
7	Quanto facilmente hai trovato l'esercizio di tuo interesse tra quelli disponibili?	5.71	1.11
8	Quanto facilmente hai trovato le informazioni relative all'esercizio?	4	2.31
9	Quanto hai trovato utili le informazioni sull'esercizio?	5	1.41
10	Trovi che i tempi di caricamento del video siano adeguati?	2.29	1.25
11	Quanto reputi chiara la schermata dei risultati?	5.86	2.19
12	Hai trovato facile navigare tra gli errori commessi?	4.86	1.68

**Tabella 1:** Tabella dei risultati ottenuti dai test di usabilità, dove  $\mu$  è la media e  $\sigma$  la deviazione standard.

Inoltre sono state formulate due ulteriori domande a risposta multipla, di cui si riportano i risultati ottenuti sotto forma di grafici a torta (Figura 12); in particolare il grafico a sinistra è relativo alla domanda “Hai scoperto facilmente che si possono eliminare le sessioni di allenamento?”, mentre quello a destra a “Hai trovato difficoltà nella lettura dei testi?”.



**Figura 12:** Grafici a torta per test a risposta multipla: quello a sinistra riguarda la facilità di scoperta della possibilità di eliminare sessioni di allenamento; quello a destra la facilità di lettura dei testi presenti nell'applicazione.

## 7 Conclusioni e miglioramenti

Dai i risultati ottenuti dai test di usabilità, emerge, senza sorpresa, la criticità più evidente dell'applicazione: i tempi di caricamento dei video sul server. Tale criticità non è legata effettivamente all'interfaccia, ma sicuramente rischia di rendere la user experience dell'applicazione quasi frustrante. Tempi di caricamento così lunghi non sono dovuti tanto a un errore di implementazione, ma quanto ai limiti software imposti dal framework Cordova e dal linguaggio JavaScript; infatti per assicurare la sincronizzazione tra frame di errore elaborati dal client e dal server e risparmiare banda, il video dell'esercizio non viene inviato

integralmente, ma a singoli frame estratti a intervalli regolari. Facendo infatti alcuni test temporali, si è notato che l'operazione in questione è quella che richiede maggior tempo di esecuzione (circa 98%), rendendo in questo modo l'intero processo di caricamento e analisi di un esercizio abbastanza lungo. Possibili soluzioni al problema potrebbero essere quelle di non inviare singoli frame ma i video integrali, cercando qualche soluzione implementativa lato server che permetta la minimizzazione degli errori dovuti alla sincronizzazione con il client, oppure cercare soluzioni alternative di pre-processing lato JavaScript (attualmente si utilizzano delle strutture *canvas* per l'estrazione dei frame).

Nonostante le difficoltà nei primi minuti di utilizzo legate alla non conoscenza dell'applicazione, gli utenti che hanno preso parte ai test hanno ritenuto l'utilizzo di VTC semplice e intuitivo, grazie anche alle interfacce non troppo complesse. Inoltre, gli stessi utenti hanno espresso anche il loro apprezzamento per l'estetica dell'intera applicazione.

Un altro punto debole, che non è stato rilevato dal questionario ma in fase di test dal moderatore, è la difficoltà di selezionare un video da analizzare dall'archivio del dispositivo usato per fare i test, aspetto che però dipende in larga parte da come la navigazione nella memoria è implementata nei vari sistemi operativi e dalle abitudini dei soggetti dei test; per venire incontro all'utente finale sarebbe necessario implementare un sistema di gestione file interno all'applicazione stessa.

Per quanto riguarda i possibili sviluppi futuri, VTC, per come è stata implementata, permette di inserire facilmente anche nuove tipologie di esercizio, per una facile scalabilità; inoltre l'utilizzo di Cordova permette una facile distribuzione di massa potendo generare gli eseguibili dell'applicazione per diverse piattaforme, sia mobile che non.

## Riferimenti bibliografici

- [1] Apache Software Foundation. *Apache Cordova Documentation*. 2022. URL: <https://cordova.apache.org/docs/en/latest/>.
- [2] Niccolò Guiducci e Giulia Bertazzini. *3D Virtual Training Coach*. 2021. URL: <https://github.com/Nick22ll/3D-Virtual-Training-Coach>.
- [3] *Il Modulo Socket*. URL: <https://docs.python.org/3/library/socket.html>.