

Relazione Finale Tirocinio

Carlo Ramponi, matricola 201720

carlo.ramponi@studenti.unitn.it, +39 3421741931

Dipartimento di ingegneria e scienza dell'informazione

Corso di Laurea in Informatica

Progetto: Training Stats

Soggetto ospitante: Fondazione Bruno Kessler

Tutor aziendale: Maurizio Napolitano

0. Presentazione soggetto ospitante	2
1. Premessa	2
2. Requisiti	3
2.1 Gestione esercizi	3
2.2 Gestione allenamenti (istanza di un esercizio)	3
2.3 Dati raccolti	4
2.4 Eventuali evoluzioni	4
3. Progettazione	4
3.1 Mockup interfaccia raccolta dati	4
3.2 Schema database	5
4. Sviluppo	6
4.1 Premessa	6
4.2 Metodologia	6
4.3 Licenza	7
4.3 Framework	7
5. Risultato	8
5.1 Video	8
5.2 Screenshots	8
5.2.1 Gestione giocatori e squadre	8
5.2.2 Impostazioni, navigazione e informazioni	9
5.2.3 Modalità segnapunti	12
5.2.4 Modalità "Simple scout":	13
5.2.4.1 Raccolta dati	13
5.2.4.2 Report	14
5.2.6 Modalità "Video scout":	16
6. Deployment	16
7. Considerazioni finali	17

0. Presentazione soggetto ospitante

FBK è la Fondazione di ricerca al top in Italia, classificata al 1° posto per l'eccellenza scientifica in 3 diverse aree tematiche e per l'impatto economico e sociale secondo i risultati dell'ultima valutazione ANVUR¹ sulla qualità della ricerca.

1. Premessa

Il progetto è frutto del tirocinio di Carlo Ramponi in FBK per lo sviluppo di una applicazione di supporto all'allenamento della pallavolo in collaborazione con la Trentino Volley

Solitamente, negli sport di squadra dove si fa uso di una palla, vengono utilizzati software di scouting (= [raccolta dati durante le partite](#)²).

Nonostante ciascuno sport abbia delle caratteristiche proprie, è possibile individuare delle funzioni di base di raccolta dei dati che individuano almeno queste tre variabili che avvengono durante una gara:

- Giocatore che ha toccato palla
- Tipo di fondamentale utilizzato
- Valutazione dell'azione appena effettuata

Si è pensato quindi di concentrare lo sviluppo su una applicazione che, anche se nasce nel contesto della pallavolo, può essere utilizzata anche in altri sport di squadra e quindi favorire meglio la sua diffusione.

L'entrare in merito ad ulteriori variabili che si possono raccogliere nella pallavolo come le direzioni in entrata ed in uscita o la stima della velocità di palla (nella pallavolo si parla di attacchi di primo, secondo e terzo tempo) potrebbero richiedere una revisione completa dell'interfaccia creando una applicazione molto specifica per un solo sport (questione che si potrebbe sviluppare in futuro).

L'introduzione dell'uso di software di raccolta dei dati durante le gare non è una novità (datavolley - il prodotto italiano più famoso a livello internazionale è nato negli anni '90 quando la pallavolo italiana ha fatto un salto di qualità a livello internazionale) ed ha creato anche una nuova figura (= lo scoutman) il cui compito è occuparsi solo di quello e, spesso, anche durante la sedute di allenamento.

Si tratta comunque di una risorsa che si possono permettere solo le squadre che lavorano nelle massime serie dove lo sport è diventato una professione.

Scendendo di livello la situazione è più precaria: gli staff tecnici sono composti da una o due persone (primo allenatore e secondo o primo allenatore e dirigente), inoltre cambiano le questioni legate alla gestione del tempo (durata delle sedute di allenamento e disponibilità dei giorni durante la settimana) e, in particolare lavorando con i giovani, anche la questione delle metodologie di allenamento utilizzate dove, non necessariamente, le sequenze specifiche del gioco vengono sviluppate.

¹ <https://www.anvur.it/>

² [https://en.wikipedia.org/wiki/Scout_\(sport\)](https://en.wikipedia.org/wiki/Scout_(sport))

Grazie quindi al dialogo con i tecnici incontrati durante il periodo di tirocinio è nata l'esigenza di sviluppare l'applicazione "Training Stats" il cui compito principale è quello di permettere una soluzione versatile e riutilizzabile anche in altri sport (ed anche non necessariamente di squadra) per il monitoraggio degli allenamenti - anche in contesti amatoriali - basata sulla raccolta delle variabili sopra indicate.

2. Requisiti

Questi requisiti sono stati definiti all'inizio del tirocinio e sono frutto di numerose riflessioni, meeting tra me, il mio tutor (anche lui allenatore e docente FIPAV) e, il direttore sportivo del giovanile della Trentino Volley, docente FIPAV e head coach della squadra UniTrento di Trentino Volley (campionato nazionale di A3) Francesco Conci.

Nei frequenti confronti con questo gruppo di lavoro abbiamo preso in considerazione più volte i requisiti applicativi, creato mockup e rilasciato versioni avvicinandoci molto alla [metodologia Agile](https://it.wikipedia.org/wiki/Metodologia_agile)³.

Siamo così arrivati ad individuare le seguenti funzionalità.

2.1 Gestione esercizi

Focalizzando l'uso dell'applicazione alla gestione degli allenamenti e considerando la difficoltà che uno staff tecnico composto da una o due persone non può immaginare di allocare il tempo totalmente alla raccolta dei dati, si è proposto di avere una gestione degli esercizi per permettere quindi di focalizzarsi su alcuni frangenti dell'allenamento a sua scelta partendo quindi dagli esercizi (= una delle componenti fondamentali di una seduta di allenamento).

- Creazione esercizio
- Eliminazione esercizio
- Modifica esercizio

2.2 Gestione allenamenti (istanza di un esercizio)

Dalla creazione degli esercizi poi si passa alla composizione di un allenamento che è composto anche dai suoi protagonisti: gli atleti.

Queste le funzioni

- Creazione allenamento:
 - Giocatori coinvolti nell'allenamento
 - Fondamentali utilizzati durante l'allenamento
- Raccolta dati durante l'allenamento:
 - Per ogni azione / tocco palla:
 - Giocatore
 - Fondamentale
 - Valutazione
 - Timestamp

³ https://it.wikipedia.org/wiki/Metodologia_agile

2.3 Dati raccolti

L'idea dell'applicazione è quella di essere in grado poi di fornire da subito un output di quello che si è raccolto al termine di ogni esercizio o allenamento e di poterlo poi condividere con altri per ulteriori analisi o per fornire un feedback.

- Riepilogo a fine allenamento con semplici statistiche.
- Esportazione dati raccolti durante gli allenamenti

2.4 Eventuali evoluzioni

Una delle caratteristiche dell'apprendimento motorio è anche quello di fornire un feedback visivo su come un gesto o un'azione di gioco è avvenuta.

Pertanto, dopo che l'applicazione ha raggiunto il suo primo ciclo di rilascio, si è deciso di concentrare lo sviluppo su una funzione che permetta di associare i dati raccolti anche ad una raccolta video che possa essere fatta direttamente dallo smartphone o tablet.

Da qui quindi si sono sviluppate queste funzioni

- Associazione dati esportati con video per estrapolare le clip delle singole azioni (implementata)
- Possibilità di personalizzare squadra e fondamentali (per portarla in altri sport) (implementata)
- Intelligenza artificiale per facilitare la raccolta dei dati (se ciò non comporta una riduzione dell'usabilità dell'applicazione) (non implementata)

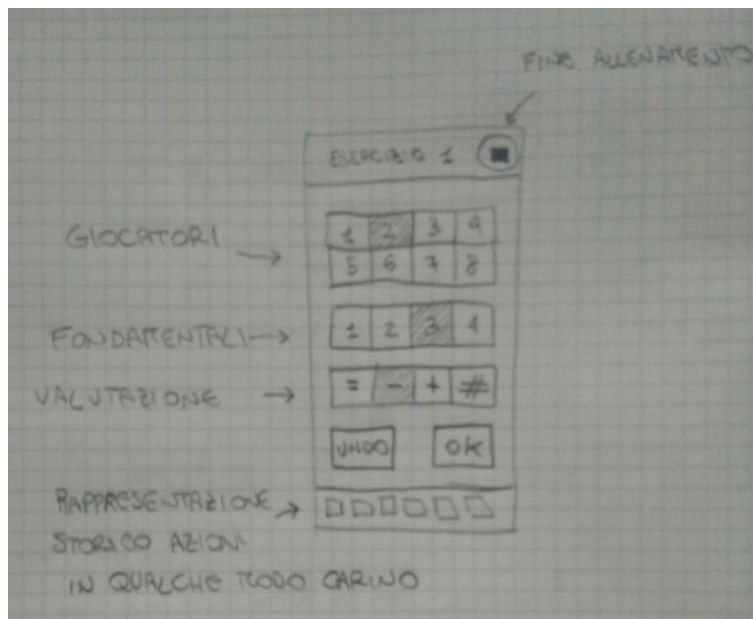
3. Progettazione

Come precedentemente riportato la fase di progettazione è stata fatta con diversi confronti con i diretti interessati (= i clienti) al prodotto.

Il primo passaggio necessario è quindi stato quello di creare per loro un mockup attraverso cui poter ragionare per avere un prodotto in grado di mostrare le sue funzionalità da subito.

3.1 Mockup interfaccia raccolta dati

Il primo passaggio quindi è stato un semplice disegno (come riportato qui sotto) di come l'interfaccia si poteva presentare



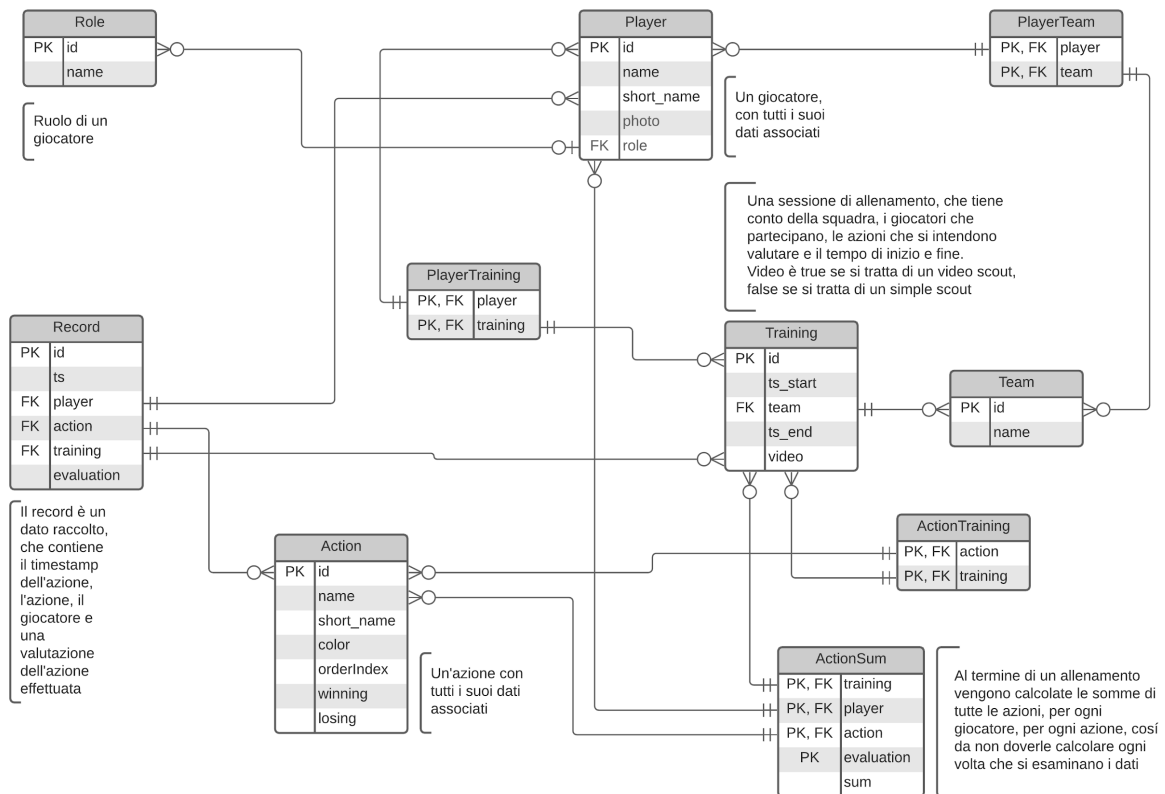
questo ha permesso di avere da subito le idee chiare tant'è che il risultato finale di questa interfaccia non si discosta molto dal prodotto finale.

3.2 Schema database

Definita l'idea di come i dati venivano raccolti e quindi capito esattamente cosa il "cliente volesse" si è passati alla creazione dello schema del database su cui l'applicazione si appoggia.

La scelta tecnologica è stata quella di un database relazionale utilizzando sqlite.

Ad ogni entità forte dello schema corrisponde una classe nel codice, che fornisce tutti i metodi per accedere a quel tipo di tabella e quelle ad essa collegate all'interno del database.



4. Sviluppo

4.1 Premessa

Lo sviluppo si è svolto in quasi completa autonomia, svolgendo meeting settimanali con il tutor per aggiornarsi sulla situazione.

Il tutor e l'allenatore Conci venivano comunque aggiornati più spesso sullo sviluppo tramite degli screenshots e video dell'applicazione e, non appena una prima versione funzionante è stata sviluppata, hanno potuto provare l'applicazione attraverso le funzionalità di test interni di google play store e apple app store.

Grazie a questa frequente interazione ho potuto ricevere molti feedback sin da subito per dirigermi nella direzione giusta.

4.2 Metodologia

Essendo stato l'unico membro del team di sviluppo ho potuto decidere quale metodologia di sviluppo/progettazione adottare, come già ho anticipato ho scelto una metodologia Agile: Scrum.

La pubblicazione del versionamento del codice è stata fatta attraverso la piattaforma [GitHub](https://github.com/)⁴, utilizzando il programma [git](https://git-scm.com/)⁵.

In questo modo il tutor ha potuto prendere visione dello sviluppo in maniera asincrona e indipendente e fornire, attraverso la creazione di “issues” segnalazioni di bug o suggerimenti di modifiche.

4.3 Licenza

È stata scelta una licenza di tipo open-source così che il codice sia disponibile a tutti e che tutti possano contribuire al miglioramento dell’app anche in futuro.

Non avevo molta esperienza con le varie licenze, ora conosco le principali differenze tra le più famose licenze open-source come [Apache License 2.0](https://opensource.org/licenses/Apache-2.0)⁶, [GNU General Public License](https://www.gnu.org/licenses/gpl-3.0.html)⁷, [GNU Lesser General Public License](https://opensource.org/licenses/lgpl-license)⁸ e [MIT License](https://opensource.org/licenses/MIT)⁹

La mia scelta è stata verso la licenza GPL 2.0 in quanto l’obbligo della condivisione allo stesso modo (share alike) diventa lo strumento attraverso cui il progetto continuerà a rimanere aperto fino che gli autori (= al momento sono io) non vorranno cambiare idea con una nuova versione.

Nello sviluppo ho seguito alla regola il metodo con cui la licenza deve essere esposta nel codice e sono stato attento a possibili conflitti di licenze con altre componenti software.

4.3 Framework

Come framework di sviluppo è stato scelto [Flutter](https://flutter.dev/)¹⁰, un recente framework open-source sviluppato da Google che permette di sviluppare applicazioni che vengono compilate **nativamente** per Android e IOS.

Il framework utilizza il linguaggio [Dart](https://www.dartlang.org/)¹¹, molto interessante perchè è un linguaggio che può essere sia interpretato che compilato che “tradotto” in javascript; infatti una nuova feature di flutter è proprio flutter web che permette di portare le proprie applicazioni flutter mobile sul web senza dover modificare il codice!

Flutter utilizza il paradigma “**Declarative programming**” che lo rende un po’ difficile da comprendere se si ha un background da sviluppatore di app con l’SDK google (Java, Kotlin) oppure IOS (Swift, Objective-C).

È inoltre possibile integrare del codice nativo Android o IOS in flutter nel caso in cui si debbano fare delle parti strettamente platform-specific, attraverso dei canali di comunicazione chiamati “Flutter channels”

⁴ <https://github.com/>

⁵ <https://git-scm.com/>

⁶ <https://opensource.org/licenses/Apache-2.0>

⁷ <https://opensource.org/licenses/gpl-license>

⁸ <https://opensource.org/licenses/lgpl-license>

⁹ <https://opensource.org/licenses/MIT>

¹⁰ <https://flutter.dev/>

¹¹ <https://www.dartlang.org/>

5. Risultato

Il risultato finale è un'applicazione funzionante che può essere utilizzata per monitorare gli allenamenti dello sport della pallavolo e riutilizzabile anche ad altri sport di squadra.

Lo sviluppo mi ha comunque appassionato e alcune nuove funzionalità che mi sono state richieste ho intenzione di inserirle nelle prossime versioni usando il mio tempo libero.

La versione attuale soddisfa tutti i requisiti ed è stata realizzata nei tempi del tirocinio senza chiedere estensioni.

La prima funzionalità che vorrei portare nella prossima versione è il riconoscimento del giocatore che sta compiendo l'azione, in base al numero di maglia o alla faccia, se le prestazioni di uno smartphone lo permettono...

Attualmente Training Stats è disponibile sugli store di [Google](#)¹² e di [Apple](#)¹³.

Seguono qui una serie di video e screenshot che mostrano lo stato attuale di Training Stats.

5.1 Video

Per venire incontro alle esigenze dei miei referenti sullo sviluppo dell'applicazione ho fornito loro dei video dimostrativi.

A questo indirizzo se ne trova uno della pre-release finale

<https://drive.google.com/file/d/1uXxyh5Br1Ld7iuHYm-X4YKT2bbVwEbSQ/view?usp=sharing>

Il video si è dimostrato utilissimo per raccogliere velocemente i feedback ed ha dimostrato subito che il percorso fatto era coerente.

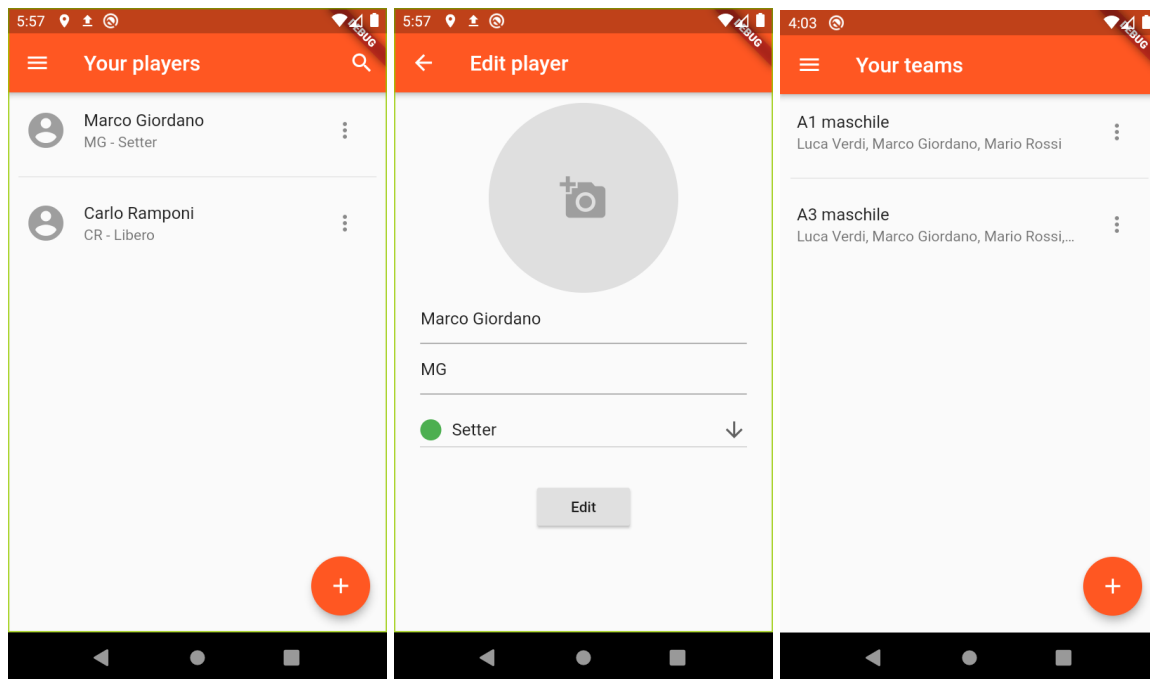
5.2 Screenshots

5.2.1 Gestione giocatori e squadre

All'interno dell'app è possibile creare le proprie squadre di allenamento che contengono i giocatori, identificati dal loro nome, una foto, una sigla (numero di maglia, iniziali) e il ruolo all'interno della squadra.

¹² <https://play.google.com/store/apps/details?id=com.drenlab.trainingstats>

¹³ <https://apps.apple.com/app/id1529717431>

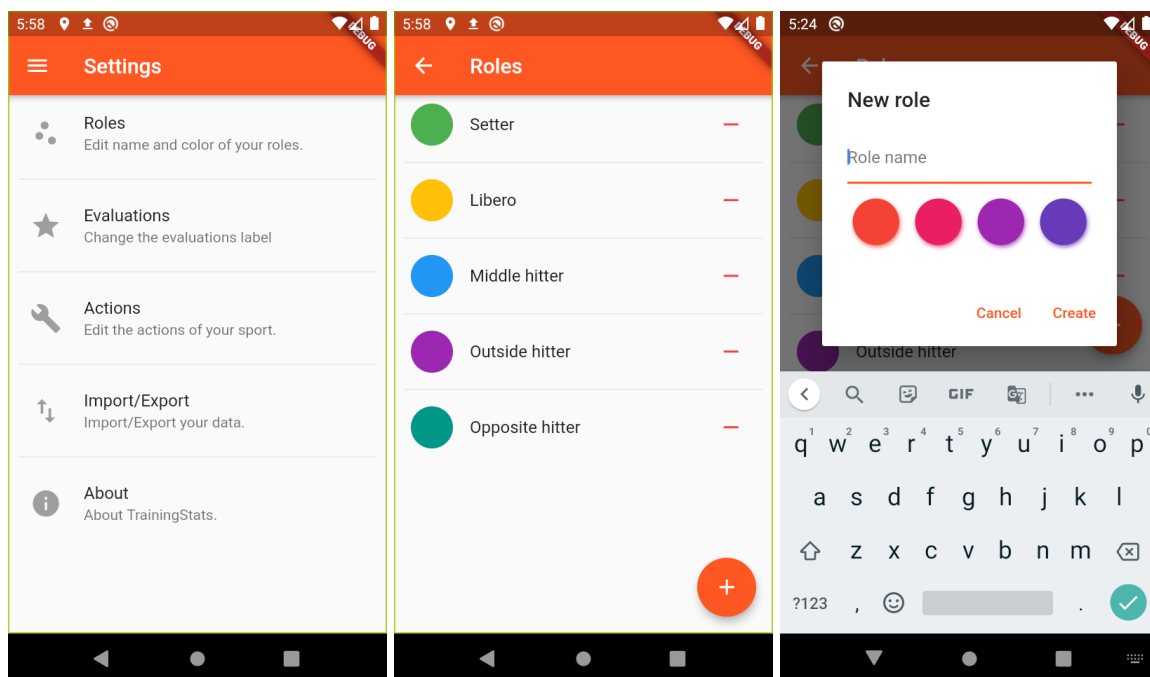


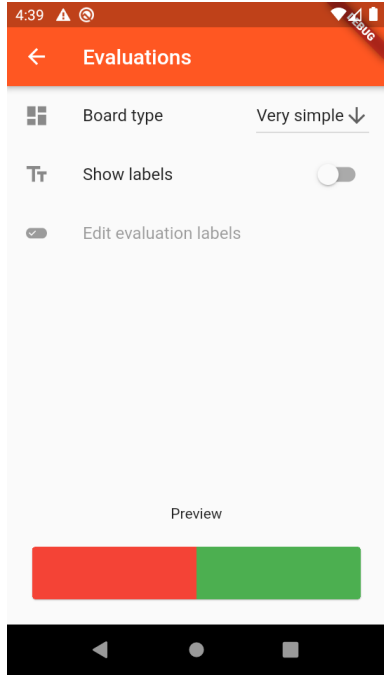
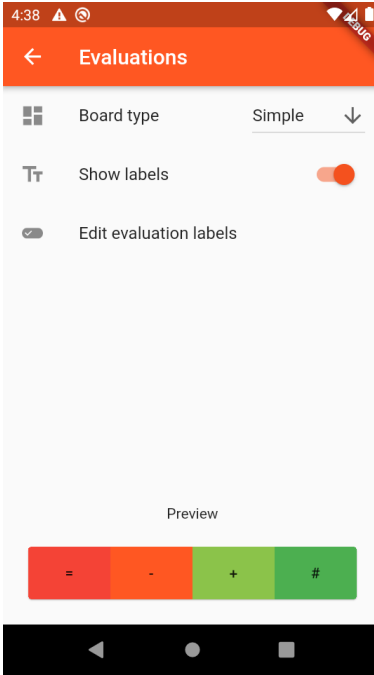
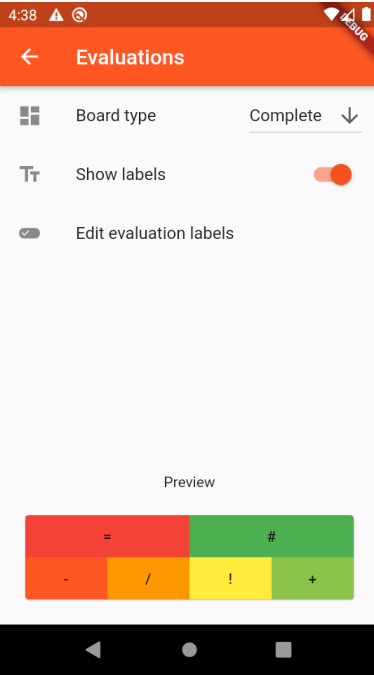
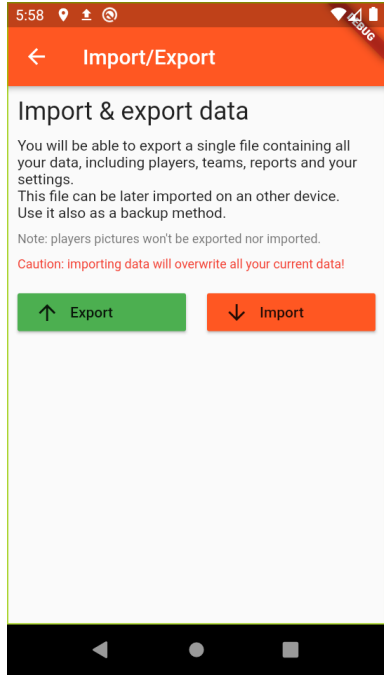
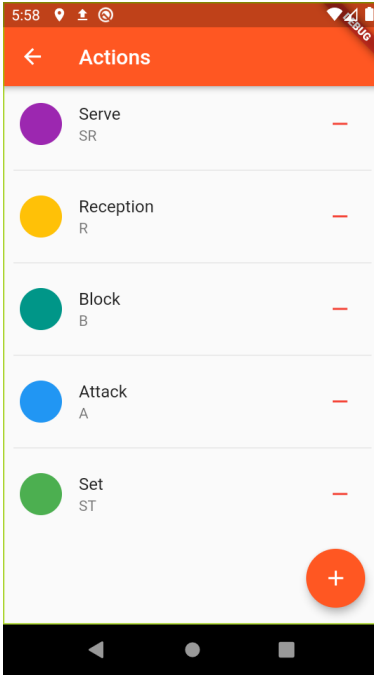
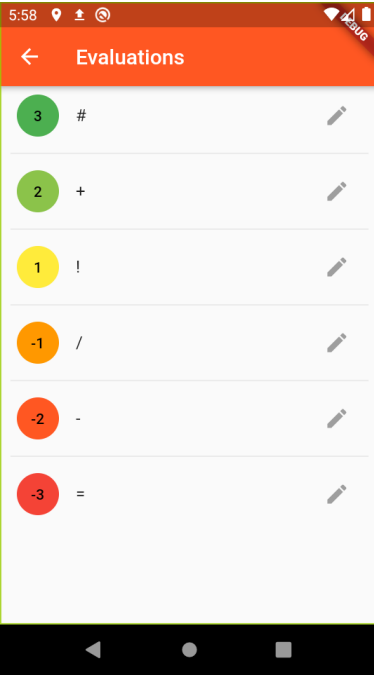
5.2.2 Impostazioni, navigazione e informazioni

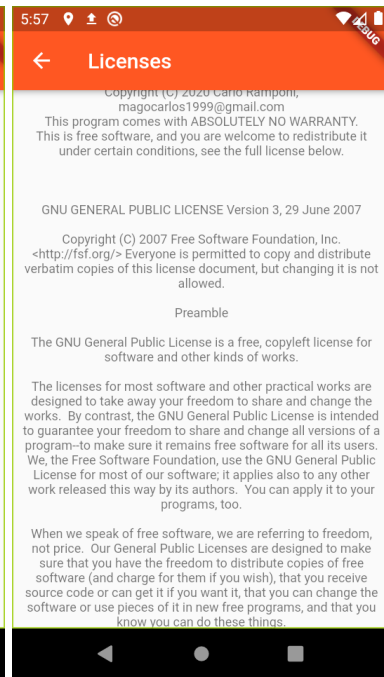
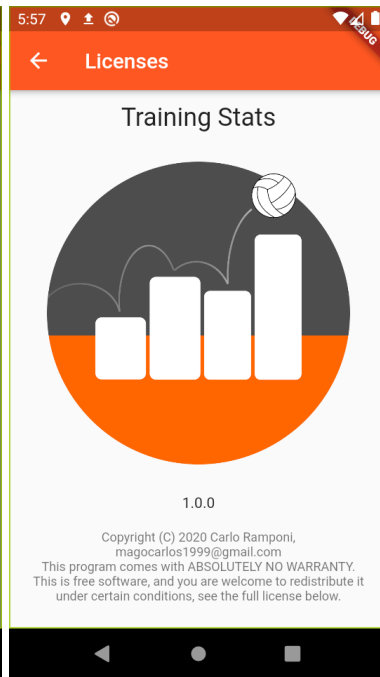
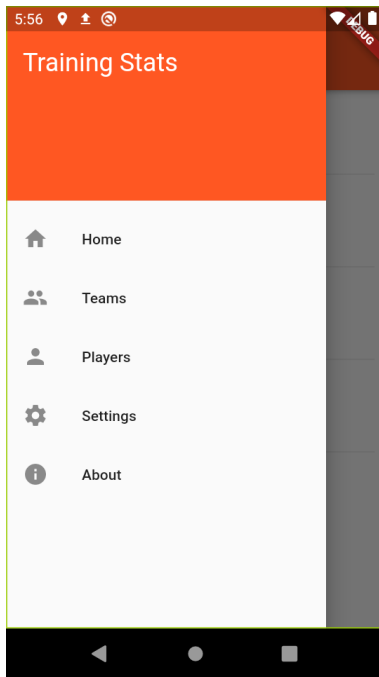
Al fine di permettere il riuso in più scenari (es. altri sport), l'applicazione è disegnata in maniera completamente configurabile, pertanto:

tutto quello che riguarda le informazioni che vengono raccolte durante una seduta di allenamento come azioni, ruoli e valutazioni possono essere configurate dall'utente.

Le configurazioni create sono esportabili in un file json che può essere poi importato su altri dispositivi e condiviso con altri utenti.

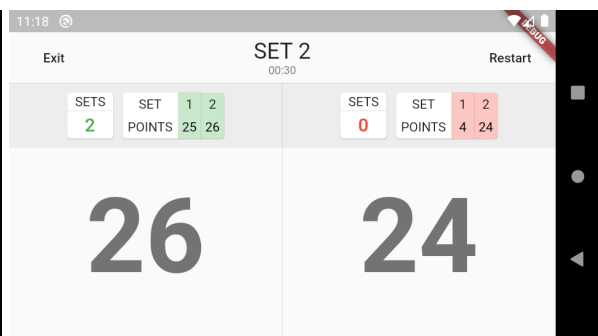
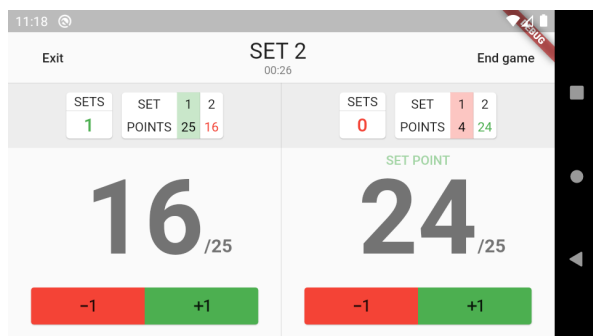
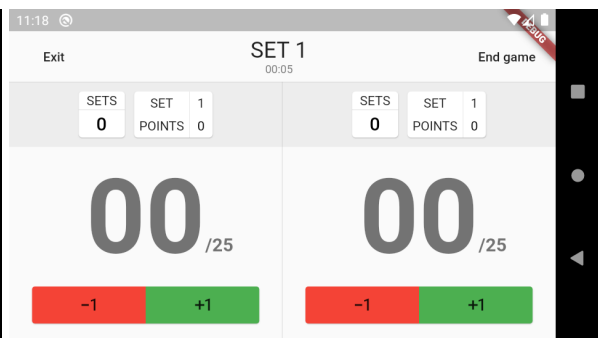
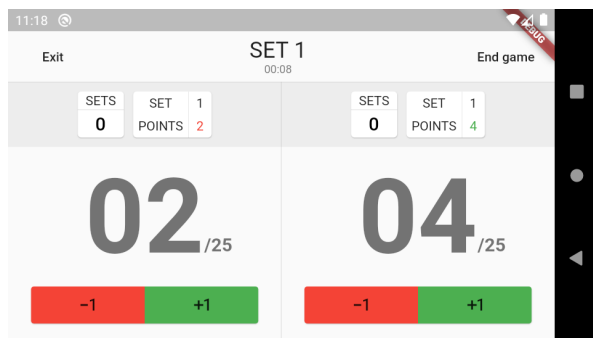
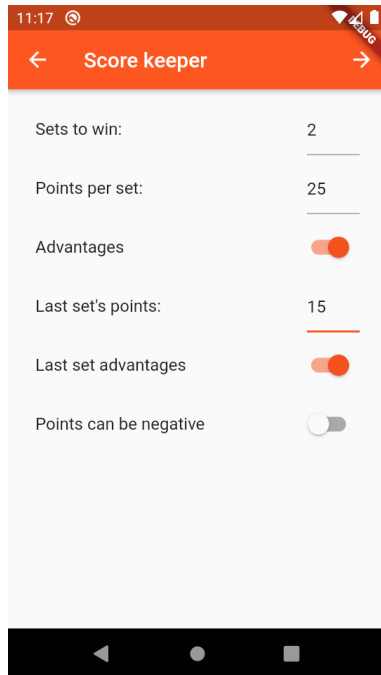






5.2.3 Modalità segnapunti

In questa modalità l'app terrà traccia dei punti durante una partita, con punteggio, sets e vantaggi configurabili a inizio partita. Questa modalità è completamente separata dal resto dell'applicazione, è solo una feature che ritenevo giusto inserire in un app come questa.



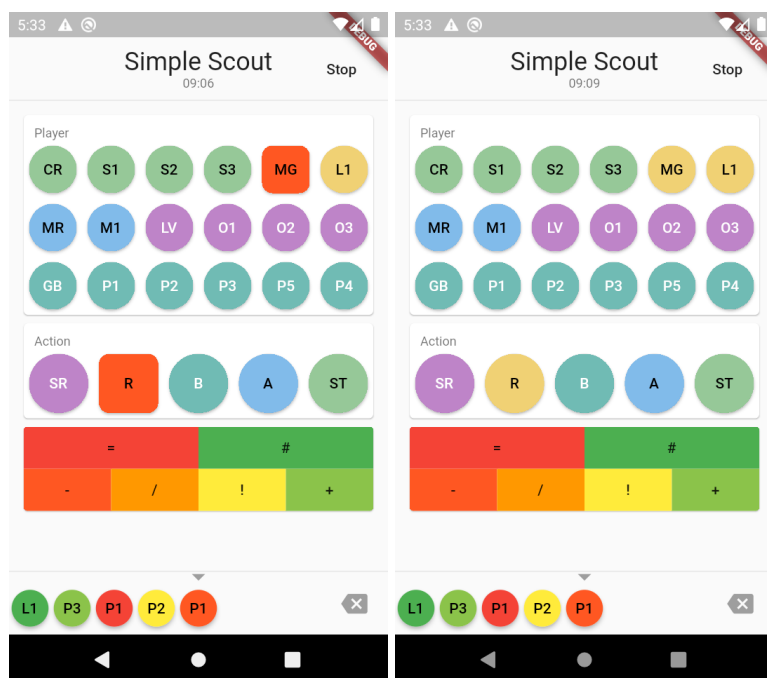
5.2.4 Modalità “Simple scout”:

5.2.4.1 Raccolta dati

E, finalmente, la modalità di raccolta dati principale.

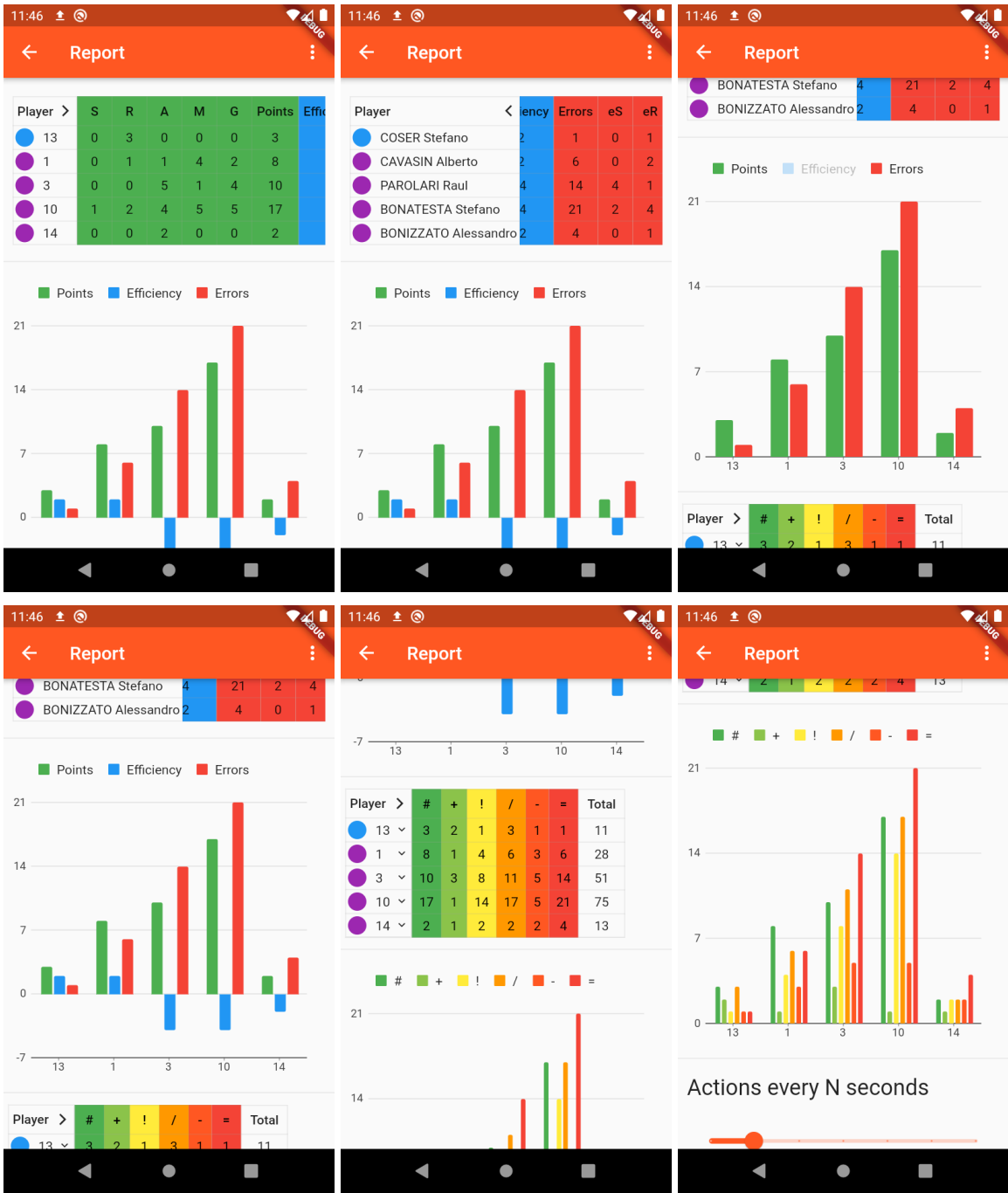
Qui i giocatori sono raggruppati per ruolo, e il colore di sfondo corrisponde proprio al ruolo del giocatore (colore personalizzabile nelle impostazioni).

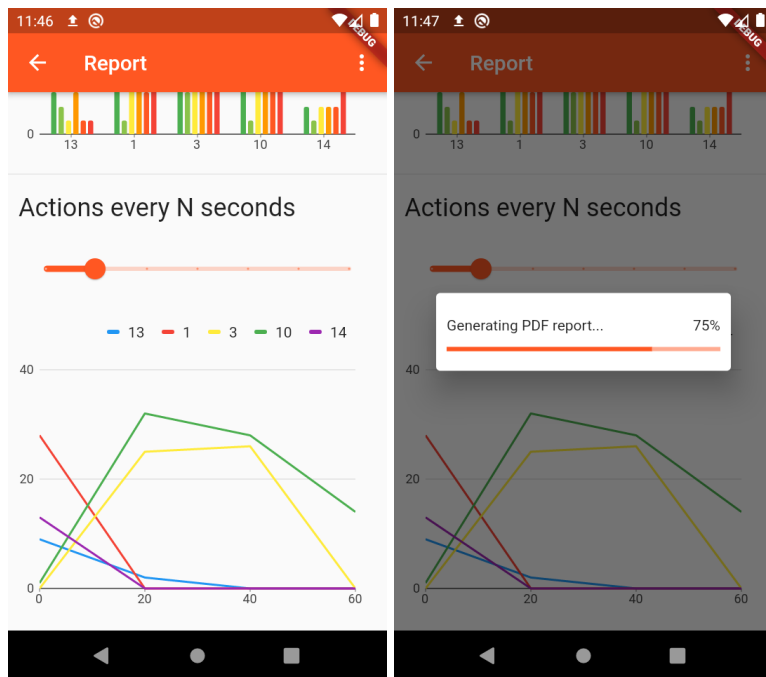
Anche le azioni hanno un colore (personalizzabile) per facilitare il tap una volta abituati al colore dell’azione.



Al termine dell’allenamento i dati raccolti verranno salvati e verrà proposto il report, che potrà essere esportato in vari formati.

5.2.4.2 Report





Il report può essere esportato in due formati diversi:

- PDF per una condivisione rapida dell'allenamento con i giocatori. In questo report ci saranno i grafici e le tabelle presenti nel report dell'app.

Es.

<https://drive.google.com/file/d/14jqkU1vpay5BYxHOMvWxrfdwvxv20NJal/view?usp=sharing>

- CSV per un'elaborazione ulteriore dei dati raccolti, infatti questo formato è facilmente importabile in altri programmi come Excel.

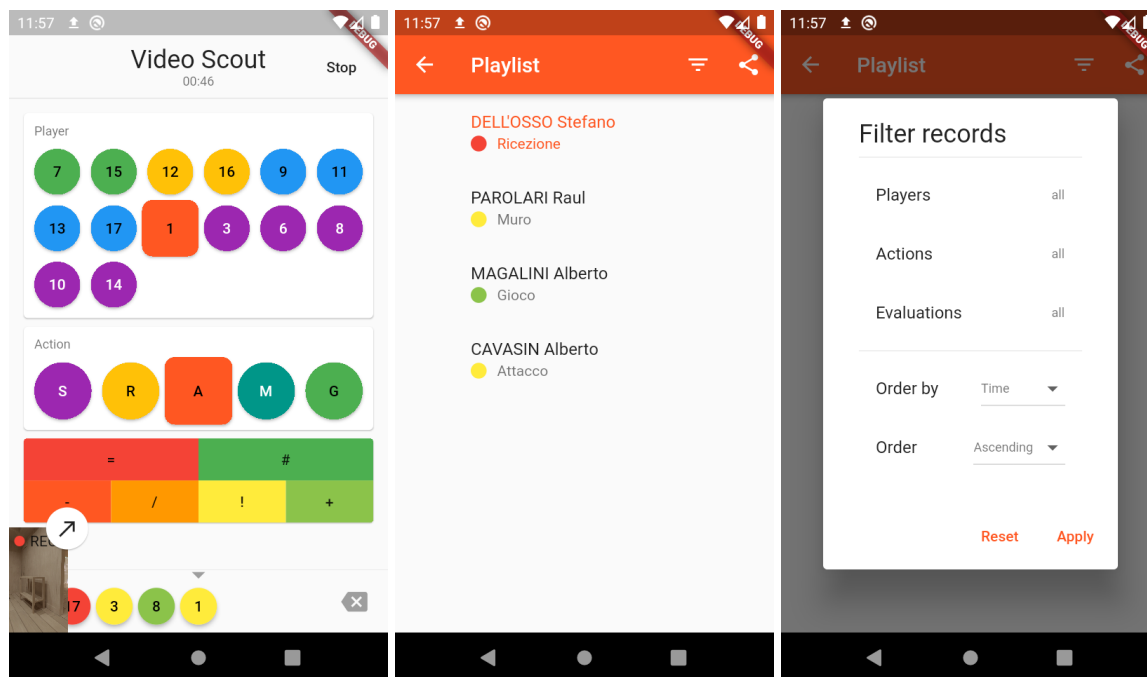
Verranno esportate le tabelle contenute nel report dell'app e i dati raccolti senza essere elaborati (ogni singola azione registrata)

Es,

<https://drive.google.com/file/d/14jUemdJEJMm5XeUua1eokwfcv9Phw2M3/view?usp=sharing>

5.2.6 Modalità “Video scout”:

In questa modalità viene associato al “simple scout” anche un video registrato con lo stesso dispositivo, da quale verranno poi estratte le clips di ogni singola azione, così da analizzarle in un secondo momento oppure esportarle ed inviarle ai giocatori.



6. Deployment

L'applicazione è open-source, quindi il codice sorgente è disponibile a tutti su [GitHub](https://github.com/CarloRamponi/TrainingStats)¹⁴, con anche lo storico di tutte le versioni.

Ho però anche caricato sugli store la versione compilata utilizzando l'account della mia startup ([Drenlab](https://www.drenlab.com/)¹⁵), ecco i link:

- [Google play store](https://play.google.com/store/apps/details?id=com.drenlab.trainingstats)¹⁶
- [Apple app store](https://apps.apple.com/app/id1529717431)¹⁷

¹⁴ <https://github.com/CarloRamponi/TrainingStats>

¹⁵ <https://www.drenlab.com/>

¹⁶ <https://play.google.com/store/apps/details?id=com.drenlab.trainingstats>

¹⁷ <https://apps.apple.com/app/id1529717431>

7. Considerazioni finali

L'esperienza è stata nel complesso molto positiva.

Ho potuto raffinare ulteriormente le mie competenze nello sviluppo di applicazioni mobile, un po' meno quelle nel team working, visto che ero solo nello sviluppo. Questo è stato però anche un bene perchè ho dovuto affrontare tutte le difficoltà implementative in autonomia. Un vero peccato non aver potuto entrare fisicamente in FBK causa pandemia, ma questo non ha compromesso in alcun modo il risultato del tirocinio.

I goal che ci eravamo fissati a inizio tirocinio, con la stesura del progetto, sono stati completamente raggiunti, non si è riuscito ad aggiungere un po' di intelligenza artificiale per mancanza di tempo, ma nulla mi impedisce di continuare lo sviluppo!

*Firmato digitalmente,
Carlo Ramponi*