



Università Degli Studi “G. d’Annunzio”

Chieti –Pescara

Dipartimento di Economia

**Corso di Laurea Triennale in *Economia e Informatica per
l’Impresa***

Tesi di Laurea

Sviluppo e progettazione di un’app Android

Relatore

Prof.ssa Francesca Scozzari

Laureando

Vito Ruggiero

Matricola

3179136

ANNO ACCADEMICO 2019/2020

“It’s in your reach.
Concentrate.”

INDICE

INTRODUZIONE	1
Capitolo 1 – Il mondo delle app	2
1.1 – Le app nella vita quotidiana	2
1.2 – <i>Swordplay – Fencing Competitions</i> , com'è nata e perché	5
Capitolo 2 – Programmazione e sviluppo	9
2.1 – <i>Java</i> come linguaggio di programmazione	9
2.2 – Sviluppo attraverso il software <i>Android Studio</i>	11
Capitolo 3 – <i>Swordplay – Fencing Competitions</i>	16
3.1 – L'app e le sue funzionalità	16
3.2 – Beyond <i>Swordplay</i> : pubblicazione e sviluppi futuri	26
CONCLUSIONE	28
Sitografia	29
Ringraziamenti	32

INTRODUZIONE

Questo lavoro intende illustrare il processo di programmazione e sviluppo di un'app Android, sottolineando anche l'importanza che assumono le app nella vita quotidiana odierna e spiegando le ragioni che hanno condotto alla creazione di questa app in particolare, *Swordplay – Fencing Competitions*.

A tal proposito, una ricerca sul mondo dello sport e in particolar modo della scherma, ha permesso di rilevare problemi sul metodo di annotazione dei punti da parte degli arbitri, in quanto attualmente il punteggio viene appuntato a penna su foglietti che possono facilmente essere persi o messi in disordine, causando spesso malintesi e proteste da parte degli atleti in gara, per punti erroneamente assegnatigli.

La tesi si articola in tre capitoli. Il primo capitolo concerne il ruolo che hanno le app al giorno d'oggi nella vita quotidiana, includendo un piccolo excursus sulla storia della loro creazione e sulle caratteristiche imprescindibili per il loro successo; inoltre, vengono inserite delle delucidazioni sull'utilizzo che potrebbe essere fatto di questa app specifica, al fine di risolvere i problemi presenti attualmente nel mondo della scherma, qui spiegati in dettaglio. Il secondo capitolo invece riguarda la parte più tecnica, la programmazione e lo sviluppo dell'app, illustrandone il processo e specificando il linguaggio di programmazione e il software utilizzati. Infine, nel terzo capitolo, viene presentata l'app ideata, spiegandone le funzionalità ed estendendo il discorso ad una futura pubblicazione e ai vantaggi che la stessa potrebbe apportare in diversi settori.

Capitolo 1

Il mondo delle app

1.1 Le app nella vita quotidiana

Al giorno d'oggi, è quasi impossibile incontrare qualcuno che non sia in possesso di uno smartphone. Lo stile di vita delle persone è mutato completamente nel corso del tempo e, ad oggi, la giornata tipo dell'individuo è scandita dall'uso di questi dispositivi intelligenti che lo aiutano ad affrontare ogni situazione in modo semplice ed efficace. Esistono app che tengono il conto dei passi per monitorare l'attività fisica o che permettono di contabilizzare le calorie assunte in una giornata per portare avanti una dieta sana ed equilibrata; app che consentono di mantenersi in contatto con amici e parenti lontani o addirittura di conoscere persone nuove, come i social network, diventati imprescindibili specialmente in questo periodo storico, segnato da un'emergenza sanitaria che impedisce i contatti reali e ravvicinati; app per appuntare note, eventi, riunioni e viaggi; app per prenotare tavoli ai ristoranti o per evitare di dover passare giornate intere in coda in luoghi come uffici postali e banche; insomma, ormai esistono app di tutti i generi e che investono ogni aspetto della vita, tanto da non essere più solo materiale di interesse della popolazione più giovane, ma da comprendere tutte le fasce d'età.

Inoltre, le app permettono di fronteggiare diverse situazioni in modo innovativo: si pensi ai ristoranti provvisti di tablet per ordinare che, non solo velocizzano il servizio, eliminando il tempo di attesa del cameriere per ordinare i propri piatti, ma che permettono, in questo periodo di emergenza sanitaria, anche di ridurre al minimo indispensabile le interazioni tra persone sconosciute; o, sempre rimanendo nello stesso contesto, le app che sono state adottate dai supermercati per consentire alle persone scegliere i prodotti da comprare prima di arrivare in negozio, così che il consumatore potesse entrare nel supermercato semplicemente per pagare e ritirare la propria spesa, evitando assembramenti all'interno; o ancora, virando dalla pandemia alla salvaguardia ambientale, i locali che si sono dotati di codici QR da incollare sul tavolo, che permette ai clienti di eseguire una semplice scansione con il proprio smartphone, per leggere cosa

offre il locale, evitando un'enorme spreco di carta utilizzata per stampare i menù e per aggiornarli ogni volta che si aggiungano o eliminino piatti e prodotti vari dalla lista.

Prima di andare avanti con il discorso però, è opportuno dare una definizione al concetto di “app”: diminutivo di “application”, è delineata, da un punto di vista informatico, come quella parte di software che ha l'obiettivo di sfruttare e migliorare l'hardware e inquadrata, in modo più concreto, come un “programma” scaricabile sui dispositivi mobili come smartphone e tablet che, attraverso la sua fruibilità e velocità, agevola la vita dell'uomo.

Il primo telefono “smart” fu realizzato da IBM e implementava al suo interno 10 semplici applicazioni, come la rubrica, il calendario, la calcolatrice e l'orologio; Nokia, invece, fu la prima ad inserire un gioco per cellulare, il famoso *Snake*; ma il vero trampolino di lancio per le app, fu fornito da Apple nel 2008, con la versione 2.0 del suo sistema operativo che presentava l'inedito *App Store*. Nello stesso anno, Google commercializzò il primo smartphone dotato di sistema operativo Android, chiamato HTC Dream ma, perché Android raggiungesse lo stesso traguardo di Apple, ossia superare il miliardo di download, ci vollero due anni in più; questo gap, però, si assottigliò sempre di più, fino ad arrivare ai 50 miliardi di download per entrambi i sistemi operativi, nel 2013.¹

L'utilizzo delle app si è insediato così tanto nella vita di tutti i giorni da spingere esperti a condurre veri e propri studi sull'impatto che esse possono avere sulla personalità, sull'umore e addirittura sulle motivazioni che spingono gli utenti ad utilizzare un'app in un determinato momento piuttosto che un'altra. A tal proposito, di particolare rilevanza, è lo studio denominato “Apposphere”, condotto da Snapchat (un social media caratterizzato da filtri buffi per foto e video) in collaborazione con Murphy Research, ideato per comprendere quali sono le motivazioni che spingono le persone ad utilizzare le proprie applicazioni preferite. Per farlo, l'azienda ha analizzato il comportamento, le sensazioni, e le emozioni di 1.005 utenti di età compresa tra i 13 e i 44 anni, sottoposti a specifici questionari. Il test è riuscito a spiegare nel dettaglio non solo perché gli utenti scelgono di utilizzare una specifica applicazione, ma anche quando la consultano e che tipo di sensazioni derivano dal suo utilizzo.² In conclusione, lo studio ha dimostrato che

¹ *La nascita dello Smartphone e l'espansione delle mobile app*, L. Diana, HDblog, 26 mag, 2017.

² *Snapchat: come le app influenzano la vita quotidiana e le emozioni*, Skytg24, 11 gen. 2019.

Snapchat è l'applicazione che rende “più felici in assoluto” rispetto alle altre e che viene spesso utilizzata in compagnia di parenti o amici; altre app molto apprezzate sono YouTube, per approfondire argomenti ritenuti interessanti o conoscere meglio determinati prodotti, e Instagram, per “spiare” la vita di influencer e celebrità, entrambe utilizzate soprattutto per passare il tempo quando si è in casa da soli; e infine Facebook e Twitter, utilizzati per tenersi al passo con le novità e gli eventi e per conversare con amici e parenti, sono le app che generano più reazioni negative da parte degli utenti, provocando spesso in loro sensazioni di ansia e sopraffazione.

Fino ad adesso, è stato affrontato l'argomento sempre prendendo in analisi gli utenti e la loro esperienza ma, al fine di contestualizzare al meglio il prossimo paragrafo, che illustrerà la nascita dell'app *Swordplay, Fencing Competitions*, realizzata appositamente per questa tesi di laurea, è necessario prendere in esame il punto di vista degli sviluppatori. Ideare un'app non è semplice, o meglio, ideare un'app di successo non è semplice. Per prima cosa bisogna chiedersi se ci sia il bisogno dell'app in questione sul mercato, se non sia già presente qualcosa di simile e perché l'utente dovrebbe scegliere questa app in particolare piuttosto che un'altra. Se l'app ideata è molto innovativa si è avvantaggiati, specialmente se è gratuita, perché gli utenti saranno più propensi a scaricarla. Se invece l'app ha già dei concorrenti sul mercato, bisogna concentrarsi su altri aspetti per far sì che venga scelta. Un dettaglio che viene spesso trascurato nelle app gratuite, ma che è uno di quelli che incide di più sulla scelta dell'app, è la grafica, correlata dall'intuitività dei comandi; la relazione con l'utente avviene attraverso l'interfaccia e si sviluppa attraverso i percorsi cognitivi che mette in atto. Il neuroscienziato Antonio Damasio sottolinea come il cervello attivi segnali inconsci legati a stimoli sensoriali, definendo gli esseri umani come “macchine emotive dotate di pensiero”. Inoltre, uno studio di Lucia Carriero, ricercatrice di Neuroset, ha evidenziato quanto sia indispensabile il pulsante *back* perché l'utente non si “senta in gabbia”, ma anche come sia necessario posizionarlo a sinistra e possibilmente in alto, in quanto in occidente si è abituati a leggere da sinistra a destra e dall'alto verso il basso. I colori dei pulsanti, poi, devono essere distinti per funzione e la forma deve essere preferibilmente tondeggiante o avere un effetto tridimensionale, in modo da dare l'idea che si stia manipolando un oggetto concreto e stimolare così la corteccia laterale premotoria del cervello che è la sezione che permette al corpo di compiere le azioni. L'app deve essere intuitiva e i comandi devono essere semplici e

pochi, quasi obbligati, in modo tale da non causare confusione e conseguente frustrazione nell'utente che non riesce a capire come usarla. In questa fase di progettazione, il team o il programmatore deve pensare fuori dagli schemi, immedesimarsi nella clientela, per cercare di capire e prevedere quali problemi, sensazioni o stati d'animo l'utente possa provare nell'utilizzo dell'applicazione. Bisogna quindi, in sostanza, migliorare sempre di più l'esperienza dell'utente tenendo presente che il cervello è un sistema complesso.

Un altro elemento fondamentale a cui dedicare particolare attenzione è il nome dell'app, che serve a comunicare l'identità del prodotto e a cosa serve: in una o due parole devono essere sintetizzati lo scopo e l'utilità, che devono poi anche essere coerenti con il logo, il quale, a sua volta, deve fornire delle informazioni immediate sull'app³ (come per esempio in questo caso, il logo dell'app *Swordplay* è un ovale contenente le tre tipologie di armi utilizzate nelle varie competizioni schermistiche).

Ricapitolando quindi, un'app, per avere successo, deve essere esteticamente bella, facile da utilizzare, con pulsanti tridimensionali, distinti per colore e per funzione, unica e riconoscibile, dotata di un nome efficace e un logo che sintetizzi la sua utilità; ma soprattutto, è necessario che ci sia bisogno della suddetta applicazione sul mercato: si tenterà appunto di applicare tutte le seguenti caratteristiche nell'app sviluppata per questa tesi di laurea.

1.2 *Swordplay – Fencing Competitions*: com'è nata e perché

Riprendendo il discorso sull'efficacia del nome, l'oggetto di cui si occupa l'app è subito riscontrabile nel titolo, ossia *Swordplay*, traducibile con “gioco di spada”, ma è ancor di più specificato nel sottotitolo esplicativo *Fencing Competitions*, ovvero “competizioni schermistiche”.

La necessità di ideare quest'app è nata da un discorso, affrontato con un arbitro, sui problemi presenti nei campionati regionali, in questo caso specifico della regione Puglia, nell'annotare i punteggi degli atleti, in quanto, attualmente, è in uso un sistema obsoleto di annotazione dei punti a mano, con carta e penna, da parte degli arbitri stessi, che nel frattempo non devono perdere di vista gli sviluppi successivi del match; da ciò, è facilmente intuibile quanto sia difficile, per queste figure professionali, annotare il

³ 7 regole per realizzare un'app di successo, D. Mangini, Wired.it, 24 mar. 2017.

punteggio giusto, continuare a seguire il duello e successivamente stilare un riepilogo preciso e completo di tutti gli assalti dei vari gironi, riordinando da sé tutti i fogli compilati durante la giornata. Successivamente, questi fogli chiamati *cartellini* in cui l'arbitro annota i punteggi, vengono consegnati ad un'altra figura, il *computerista*, che deve assicurarsi che il riepilogo stilato dall'arbitro sia corretto, controllando ogni singolo cartellino e riannotando ogni punteggio in forma digitale sul computer. Non c'è da sorprendersi, infatti, che possa capitare di sbagliare nell'assegnazione dei punti, perché questo processo macchinoso, caratterizzato da tante fasi di ricopiatura dei punteggi, dura generalmente un'ora e mezza e, nei vari passaggi, non è difficile commettere errori. Di conseguenza, queste sviste generano inevitabilmente proteste da parte degli atleti che hanno ricevuto un punteggio non congruo alle loro prestazioni e tutto ciò comporta una grandissima perdita di tempo e di energie.

Prima di spiegare concretamente le funzionalità dell'app, è opportuno introdurre lo sport della scherma, in modo tale da permettere al lettore che non ha familiarità con l'argomento, di comprendere come un arbitro possa utilizzarla e quali benefici possa trarne.

Dal punto di vista etimologico, la parola "scherma" deriva dal verbo "schermire" che indica l'atto del ripararsi e del difendersi, ma molti studiosi guardano molto più indietro nella storia per cercarne l'origine, trovandola nel termine "skirmen" in tedesco arcaico, che vuol dire proteggere e, per sinonimia, difendersi. L'Accademia della Crusca definisce "schermire" come "riparare con arte il colpo che tira il nemico e cercare di offenderlo sempre", facendo riferimento, appunto, all'arte della scherma. Tale arte, nata nel XIII secolo, si evolverà nel corso del tempo, così come si evolveranno le sue armi; il duello, inizialmente limitato alla scherma militare, si estenderà all'uso civile come "salvaguardia dell'onore e dei principi morali", fino a passare da mero concetto di arte a vero e proprio sport da combattimento con le olimpiadi del 1913 e la contemporanea nascita della Federazione Internazionale di Scherma (FIE, dal francese *Fédération Internationale d'Escrime*)⁴.

La scherma si divide in tre competizioni, in base al tipo di arma usata; esse sono:

- Fioretto: è l'arma che può colpire solo di punta ed è l'arma con cui generalmente i più piccoli si avvicinano allo sport perché considerata più adatta per imparare

⁴ Federazione Italiana Scherma, federscherma.it, La scherma, *La Storia*.

le azioni fondamentali. L'unico bersaglio valido è il tronco dell'atleta, non lo sono, invece, testa, braccia e gambe, anche se è stata introdotto, come bersaglio valido, parte della gorgiera della maschera dal 1° gennaio del 2009. Il punto viene assegnato al duellante che attacca per primo, che para e risponde o che ha l'arma "in linea" (braccio disteso e punta che minaccia il bersaglio valido) prima dell'inizio dell'attacco dell'avversario;

- Sciabola: è l'arma più dinamica e veloce, generalmente scelta dagli atleti con un temperamento più aggressivo e vivace. Si può colpire con tutta la lama, quindi di punta, di taglio e controtaglio e il bersaglio valido, a differenza del fioretto, è più ampio perché interessa tutta la parte superiore del corpo dell'avversario. L'assegnazione dei punti segue le medesime regole del fioretto;
- Spada: quest'arma può colpire solo di punta ma il bersaglio valido è esteso a tutto il corpo. È la specialità più matura e meditata e l'assegnazione dei punti va a chi semplicemente colpisce per primo, senza seguire le regole più ferree delle altre due competizioni. È anche possibile che il punto venga assegnato ad entrambi i duellanti nel caso in cui avvenga un colpo doppio nell'arco di 40-50 millisecondi.⁵

Precedentemente, all'inizio di questo paragrafo, sono stati utilizzati termini sportivi impropriamente, per permettere al lettore non informato sullo sport in questione, di comprendere al meglio l'argomento. Dando uno sguardo al regolamento regionale, è possibile adesso fare chiarezza sulla terminologia specifica, per permettere una lettura più precisa e consapevole delle funzionalità dell'app che verranno illustrate a breve. È definito *assalto* il combattimento cortese tra due schermatori e il risultato di questo combattimento viene indicato con la parola *match*. La scherma è caratterizzata non solo da prove individuali, ma anche da prove a squadre ed è designato con il termine *incontro*, l'insieme dei match tra schermatori di due squadre diverse. La *competizione* invece, è l'insieme dei match o degli incontri necessari per designare il vincitore della gara. Ogni gara ha determinate caratteristiche che la distinguono dalle altre, come il sesso degli schermatori, l'età, e il fatto che le competizioni possano essere a squadre o individuali.⁶ Ogni incontro di scherma in una competizione ufficiale della FIE è

⁵ Federazione Italiana Scherma, federscherma.it, La scherma, *Le Armi*.

⁶ Regolamento tecnico per le gare, Capitolo 2, *Terminologia*. Per mancato reperimento del regolamento della regione Puglia, è stato utilizzato qui, come fonte, il regolamento reperito sul sito schermalazio.it, con la sicurezza che i regolamenti siano i medesimi in ogni regione.

diretto da un arbitro che ha diversi compiti, tra cui quello dell'assegnazione delle *stoccate* (ossia, colpo rapido di punta⁷) e quindi dei punti. Ed è proprio a loro che è rivolta quest'app, agli arbitri, per agevolare il loro lavoro nell'annotazione dei punti. L'app, infatti, dopo avergli dato il benvenuto, chiede all'utente di scegliere il tipo di arma e successivamente di inserire informazioni come la fascia d'età degli atleti, il sesso, il numero dei partecipanti, il girone, la pedana e il nome dell'arbitro stesso. Al completamento della compilazione, l'app genererà una schermata in cui compariranno tutti gli assalti e l'utente potrà autonomamente aprirli, inserire il nome degli atleti e assegnare i punti e, conseguentemente, scaricare un documento che faccia da riepilogo di tutta la competizione avvenuta nella giornata, riducendo considerevolmente gli sforzi e il tempo necessari per redarlo e agevolando così il lavoro dell'arbitro che, grazie alla velocità e all'intuitività dell'app, sarà molto probabilmente anche più preciso nell'annotazione. L'applicazione sarà poi di ausilio anche al computerista, in quanto il documento generato alla fine non conterrà errori di copiatura e non sarà quindi necessario stilarlo nuovamente per controllarlo. Tutto ciò, però, verrà illustrato in dettaglio nel terzo capitolo, in cui saranno illustrate le varie schermate con cui l'utente avrà modo di interagire, al fine di presentare al meglio l'app ideata.

⁷ Treccani, vocabolario online.

Capitolo 2

Programmazione e sviluppo

1.1 Java come linguaggio di programmazione

L'app *Swordplay* è stata sviluppata utilizzando il linguaggio di programmazione *Java*.

Java è stato sviluppato nel 1995 da *Sun Microsystem*, un'azienda statunitense produttrice di software e semiconduttori, con sede nella Silicon Valley. La sopracitata società è stata poi acquistata nel 2010 da Oracle Corporation, che tutt'oggi continua ad aggiornare il linguaggio di programmazione in questione.

Le ragioni alla base della scelta di questo linguaggio di programmazioni sono molteplici: prima di tutto *Java* è ormai onnipresente, dai portatili ai datacenter, dalle console per videogiochi ai computer altamente scientifici, ai telefoni cellulari e a internet, in quanto esiste un numero considerevole e sempre in aumento di applicazioni e siti web che funzionano esclusivamente se viene installato Java sul dispositivo in questione.⁸ Ci sono diversi dati che confermano quanto appena detto, tra cui il numero di sviluppatori, che ha raggiunto i 9 milioni in tutto il mondo, o ancora che il 97% dei computer desktop aziendali utilizza Java, così come 3 miliardi di telefoni cellulari e 125 milioni di dispositivi TV. Java, infatti, è stato progettato per consentire lo sviluppo delle più disparate applicazioni, utilizzabili su un ambiente eterogeneo, costituito da un'ampia gamma di piattaforme di elaborazione. Esso è perciò diventato insostituibile per la maggior parte dei programmatori, consentendo loro di scrivere software su una piattaforma specifica per poi eseguirli su ogni altro tipo di piattaforma; sviluppare applicazioni lato server per negozi, sondaggi ed elaborazione di form; creare programmi che possono essere eseguiti all'interno di un browser web e accedere ai servizi web disponibili; scrivere applicazioni potenti ed efficienti per dispositivi elettronici di ogni genere e molto altro ancora.⁹

Inoltre, l'indice di TIOBE, nell'ottobre 2019, vedeva Java al primo posto in classifica, posizione mantenuta fino al 1° aprile dell'anno corrente, seguita da linguaggio di

⁸ *Che cos'è la tecnologia Java e a cosa serve?*, Java.it.

⁹ *Informazioni sulla tecnologia Java*, Java.it.

programmazione *C*, che lo ha attualmente superato scambiando le due posizioni, ma con uno scarto minimo; nonostante ciò, Java non è mai sceso al di sotto della seconda posizione ed è stato addirittura eletto “language of the year” per ben 10 anni di seguito, dal 2005 al 2015.¹⁰ TIOBE è un indice di popolarità dei linguaggi di programmazione, aggiornato mensilmente con valutazioni fornite da un gran numero di ingegneri qualificati provenienti da tutto il mondo. L’acronimo TIOBE sta per “The Importance of Being Earnest” e fa riferimento al titolo di una commedia di Oscar Wild del 1895, in cui è presente un gioco di parole tra il nome proprio del protagonista *Ernest* e la parola omofona *earnest* che in inglese significa “sincero” (per intenderci, in italiano lo stesso gioco di parole potrebbe essere reso con l’omografia e omofonia tra il nome proprio di persona *Franco* e l’aggettivo *franco* derivante dal sostantivo “franchezza”, ossia sincerità). Il titolo, infatti, allude all’atteggiamento di sincerità e professionalità dei fondatori nei confronti dei consumatori, dei fornitori e dei colleghi.¹¹ L’indice TIOBE però, non ha l’obiettivo di sancire la superiorità di un linguaggio di programmazione rispetto ad un altro, ma piuttosto consente al programmatore di verificare che le sue conoscenze siano al passo con i tempi e, in caso contrario, di aggiornarle e per prendere decisioni strategiche su quale linguaggio sia più opportuno utilizzare per iniziare un progetto.

Sebbene i linguaggi di programmazione *Java* e *C* si contendano le prime due posizioni da anni, ultimamente un altro linguaggio di programmazione denominato *Python*, ha scalato la classifica, posizionandosi al terzo posto e avvicinandosi sempre di più a Java, tanto che il divario tra i due raggiunge, attualmente, appena l’1,3%; se la sua popolarità dovesse aumentare, potrebbe superare entrambi i linguaggi di programmazione, ribaltando una classifica ormai ferma a questi risultati dai primi anni 2000.

Infine, nonostante Java sia disponibile da tanti anni, è ancora uno dei linguaggi di programmazione più popolari al mondo e le abilità di programmazione Java sono tra le più richieste dai datori di lavoro per la creazione di software per le aziende private e della pubblica amministrazione; inoltre, gli sviluppatori che hanno buona conoscenza di questo linguaggio di programmazione sono tra i più pagati nel settore.¹²

¹⁰ *The Java Programming Language*, tiobe.com.

¹¹ “The name TIOBE stands for “The Importance Of Being Earnest”. This is also the name of a comedy play written by Oscar Wilde at the end of the Nineteenth Century. By choosing this name, the founders of TIOBE Software emphasize their sincere and professional attitude towards customers, suppliers and colleagues.”, about company, tiobe.com.

¹² *5 motivi per apprendere il linguaggio Java*, fiorerosalba.com.

2.2 Sviluppo attraverso il software *Android Studio*

L'app *Swordplay – Fencing Competitions*, presentata in questa tesi di laurea, è stata sviluppata in Android attraverso il software *Android Studio*.

Il software *Android Studio* è definito, come riportato nel sito ufficiale, “ambiente di sviluppo (IDE) ufficiale per le app android” basato su *IntelliJ IDEA*¹³. IntelliJ IDEA è un IDE indirizzato specialmente alla stesura e gestione di codici Java, vincitore di un Award e creato dalla compagnia *JetBrains*, azienda produttrice di software a livello globale, con sede a Praga, specializzata nella produzione di strumenti intelligenti dedicati a singoli sviluppatori o team.¹⁴ Il software sopracitato consente al programmatore di sviluppare app Android potendo usufruire anche di un simulatore (che permette di visualizzare un’anteprima di come si presenterà visivamente l’app su un dispositivo mobile) veloce e ricco di funzionalità, ma soprattutto fornisce un ambiente unificato, permettendogli di sviluppare app utilizzabili su qualunque tipo di dispositivo Android.

Android nasce nel 2003 come sistema operativo per fotocamere digitali, per permettere agli utenti di poter aggiungere nuove potenzialità ai propri dispositivi. I quattro fondatori dell’*Android Inc.* (A. Rubin, R. Miner, N. Sears e C. White) però, decisero di virare l’attenzione sui “cellulari intelligenti” con la convinzione, rivelatasi poi vincente, che ben presto avrebbero soppiantato le fotocamere. Da allora, si iniziarono a produrre migliaia di dispositivi di tutti i generi con sistema operativo Android, in particolare smartphone, da sempre in competizione con i dispositivi Apple aventi un sistema operativo denominato *iOS*. I vantaggi di scegliere un dispositivo Android però, sono molteplici: primo tra tutti è il prezzo, perché è possibile acquistare dispositivi Android, qualitativamente al pari dei dispositivi Apple (se non con specifiche addirittura superiori), a costi nettamente inferiori. Questo è dipeso dal fatto che Android è un sistema operativo “open source” che può essere scaricato gratuitamente dai produttori di dispositivi.¹⁵ Il fatto che sia “open source” indica che il codice sorgente¹⁶ può essere visualizzato,

¹³ “Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on IntelliJ IDEA”, *Meet Android Studio*, User Guide, developer.android.com.

¹⁴ *JetBrains Corporate Overview* (sezione “About”).

¹⁵ Android.com, Facts, Fatto #2: “Ogni nuovo aggiornamento di Android è un sistema operativo per dispositivi mobili open source, completamente funzionale e gratuito, che può essere utilizzato dai produttori per realizzare dispositivi.”

¹⁶ Versione di un algoritmo scritta in un linguaggio di programmazione ad alto livello (ossia più vicino al linguaggio umano, tipicamente in pseudo inglese), le cui istruzioni sono poi eseguite dalla macchina

scaricato e anche migliorato gratuitamente da chiunque.¹⁷ Altri vantaggi del sistema operativo Android sono: la sua grande possibilità di personalizzazione, che permette ad ogni utente di cambiare completamente i connotati del proprio dispositivo fino a renderlo irriconoscibile; la condivisione, praticamente ogni app è dotata del tasto “condividi” che permette di inoltrare a chiunque, ogni genere di file, dalle foto ai documenti e così via; la semplicità nel fare i *backup*, perché a differenza di iOS, Android permette di salvare in maniera selettiva o completa, l'intero contenuto di un dispositivo su qualunque supporto di archiviazione esterna; il trasferimento dati attuabile in modo semplice e veloce, con l'utilizzo di un cavo USB o del Bluetooth, verso qualunque altro tipo di dispositivo, senza complicazioni e limiti; e infine l'opportunità di utilizzare non solo un unico Store per le app, come l'App Store predefinito di Apple, ma anche Store alternativi, per poter scaricare applicazioni non presenti nello store ufficiale.¹⁸

Dopo questa breve introduzione generale su questo sistema operativo, si può ora passare all'argomentazione degli aspetti tecnici riguardanti lo sviluppo di un app Android. Programmare un'app richiede una quantità di tempo non indifferente e, per costruire da zero il codice su cui essa sarà basata, lo sviluppatore deve necessariamente avere già ben in mente come l'app sarà strutturata e come l'utente dovrà e potrà interagire con essa. Perché l'app possa essere sviluppata, bisogna partire da tre elementi di base:

- il file *AndroidManifest.xml*;
- la cartella *src*;
- la cartella *res*.

Vediamo ora nel dettaglio cosa sono questi elementi.

Il primo è un file in formato *xml* in cui sono contenute le informazioni di base sull'app e il suo comportamento, i permessi che l'app dovrà richiedere alle varie risorse del dispositivo (come l'accesso alla memoria interna, l'utilizzo della fotocamera, l'accesso ad Internet, ecc.) e le caratteristiche hardware e software che i device

mediante appositi programmi (compilatori, assembleri o interpreti). [...] (Treccani, enciclopedia online).

¹⁷ Android.com, Facts, Fatto #1: “Poiché si tratta di una piattaforma open source, il codice sorgente di Android può essere visualizzato, scaricato, modificato, migliorato e ridistribuito da chiunque, senza dover pagare commissioni, royalty o altri tipi di costi. Questo è il contrario di ciò che accade con i software closed source/proprietary, che non rendono mai pubblico il proprio codice sorgente e proibiscono tassativamente qualsiasi modifica.”

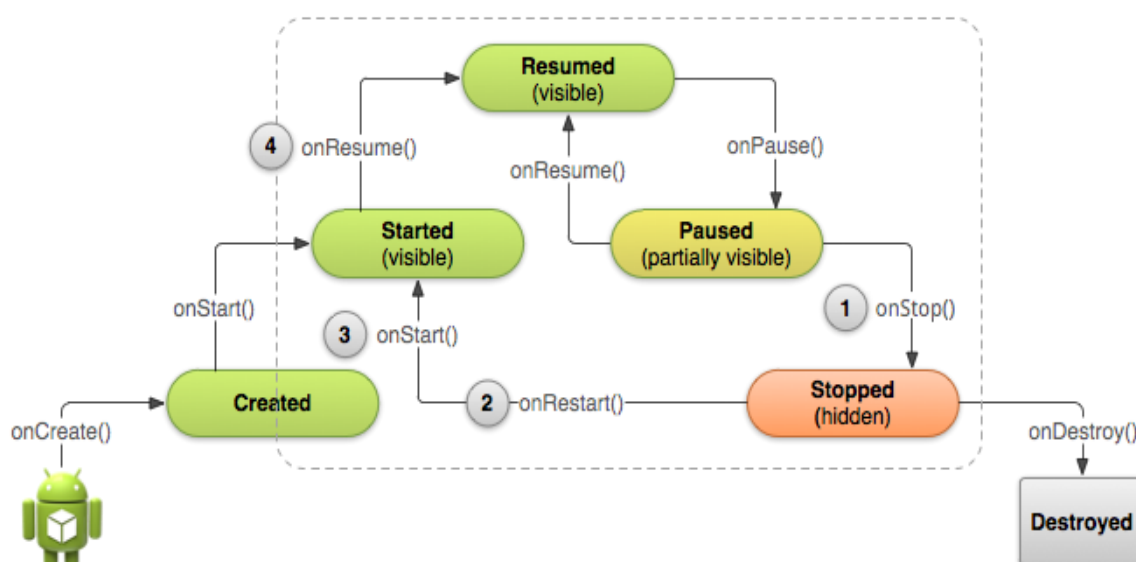
¹⁸ *Perché scegliere uno smartphone Android: 8 validi motivi*, redazione Focustech, 31 mag. 2019, focustech.it

dovranno possedere perché sia possibile scaricare l'app da uno store¹⁹. Nell'app *Swordplay*, per esempio, il Manifest richiede al dispositivo i permessi per accedere alla memoria interna in cui poi verrà scaricato il riepilogo della competizione in formato PDF; il comando attraverso il cui la richiesta viene inoltrata al dispositivo in questione, è contenuto nelle seguenti stringhe di codice:

```
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

```
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
```

Il secondo elemento necessario alla creazione di un app Android è la cartella *src*, dove *src* è l'abbreviazione di *source*, ossia “fonte, sorgente”. Questa cartella, infatti, contiene il codice sorgente dell'app; esso è composto da classi di *Activity*. Un'*Activity* è il punto di contatto tra l'utente e l'app, in quanto genera delle “finestre” che funzionano da interfaccia e permettono all'user di interagire con essa.²⁰ Ogni app può contenere un numero diverso di Activity e ognuna di esse ha una funzione ben precisa che corrisponde alle diverse interazioni che si possono avere con l'utente. Il funzionamento della Activity segue un processo articolato in diverse fasi, come illustrato nel seguente schema.



¹⁹ *App Manifest Overview*, Guides, developer.android.com

²⁰ *The concept of activities*, Introduction to Activities, Guides, developer.android.com.

Quando un'Activity va in esecuzione, si dice, in gergo informatico, che viene “lanciata” e può quindi interagire con l'utente. In questo esatto momento vengono invocati tre metodi definiti *callback*:

- onCreate() : con questo metodo l'Activity viene creata; attraverso di esso il programmatore potrà definire il layout;
- onStart() : attraverso questo metodo l'Activity diventa visibile; l'utente può quindi interagire con essa;
- onResume() : le interazioni dell'utente con l'activity vengono immagazzinate attraverso questo metodo e l'activity stessa diventa, quindi, destinataria di tutti gli input ricevuti dall'user.

Se l'utente cessa l'interazione con l'Activity sopracitata, per esempio rispondendo ad una telefonata, aprendo un'altra applicazione o semplicemente spostando l'attenzione su una funzione diversa della stessa applicazione, si attiva un processo opposto al precedente, composto da altrettanti metodi di callback:

- onPause() : questo metodo è opposto al precedente onResume(), in quanto informa il sistema della cessata interazione con l'utente;
- onStop() : tale metodo svolge una funzione contraria a quella di onStart() perché, se con esso l'Activity diventava visibile, onStop() ne annulla la visibilità;
- onDestroy() : questo metodo viene applicato nel caso in cui l'utente chiudesse definitivamente l'app di sua spontanea volontà oppure a causa di insufficienza di memoria da parte del dispositivo.²¹

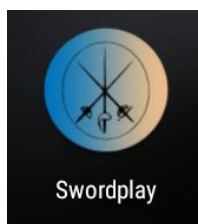
Ogni app è composta da più Activity e, per passare da un'Activity a un'altra, viene utilizzato l'*Intent*. Esso è un modo per richiedere l'esecuzione di un'azione da una componente all'altra di un'app. Sebbene l'Intent faciliti la comunicazione tra le varie componenti in diversi modi, è utilizzato principalmente in tre casi: l'avvio di un'Activity, l'avvio di un servizio e l'invio di un messaggio broadcast che può essere recepito da ogni tipo di applicazione. Inoltre, un'Intent può essere di due tipi, esplicito, quando specifica quale componente deve essere attivata, utilizzato

²¹ *Il ciclo di un'activity*, G. Maggi, html.it, 22 giu. 2020.

soprattutto al momento dell'apertura di una nuova Activity; e implicito, quando indica solo quale azione debba essere svolta, senza specificare le componenti.²²

L'ultimo elemento di base necessario allo sviluppo di un'app è la cartella *res*. Ogni applicazione è caratterizzata da diverse schermate che, la maggior parte delle volte, si presentano in modo graficamente diverso l'una dall'altra; tutti i file e le informazioni riguardanti la parte grafica sono contenuti nella sopracitata cartella, il cui nome, ossia *res*, non è altro che l'abbreviazione di *resource* (risorsa). Essa è suddivisa a sua volta in sottocartelle, che prendono il nome dalle informazioni che contengono. Le sottocartelle più comuni sono:

- *layout*: questa cartella contiene l'architettura grafica dell'interfaccia utente in formato xml;
- *values*: come dice il nome stesso, in questa cartella sono racchiuse diverse tipologie di valori come stringhe, colori, dimensioni e altro ancora; per esempio, facendo riferimento alle stringhe, nella cartella *values* dell'app *Swordplay* è presente il file *strings.xml* che contiene tutta la parte scritta dell'app. Questo file non è indispensabile ai fini dello sviluppo dell'app, ma è altamente consigliato in quanto permetterebbe di tradurre l'intera app in più lingue, semplicemente traducendone il contenuto;
- *drawable*: essa contiene tutte le immagini che verranno visualizzate nel layout, come le immagini di background per esempio;
- *mipmap*: questa cartella, un tempo parte integrante della precedente e successivamente divisa, contiene l'icona che l'utente visualizza nel menù del dispositivo e su cui clicca per aprire l'app²³. Sempre facendo riferimento a *Swordplay*, nella cartella *mipmap* del progetto è contenuto il logo di seguito illustrato.



²² *Intents and Intent Filters*, Guides, developer.android.com.

²³ *Gestire le risorse e gli asset*, G. Maggi, html.it, 1 lug. 2020.

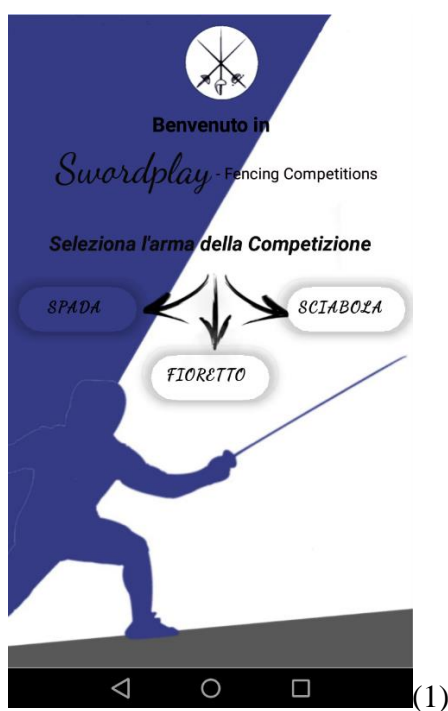
Capitolo 3

Swordplay – Fencing Competitions

3.1 L'app e le sue funzionalità

In questo capitolo verrà illustrata l'app in ogni sua parte, commentando i codici che la compongono e corredando il discorso con gli screenshot di tutte le schermate con cui l'utente potrà interagire.

All'avvio, l'user si troverà la seguente schermata iniziale, che gli darà il benvenuto e gli permetterà di scegliere la competizione in base all'arma che verrà utilizzata dagli atleti.



Da questa schermata, si intuisce l'intenzione dello sviluppatore di presentare un'app semplice e intuitiva, di facile utilizzo. Il nome accattivante dà già un'idea della funzione dell'app, che è poi esplicitata dal sottotitolo e ulteriormente confermata dall'immagine di sfondo e dal logo che rappresenta le tre armi simbolo delle competizioni schermistiche.

Le azioni che l'utente può compiere su questa schermata (ossia la possibilità di cliccare su uno dei tre pulsanti disponibili per scegliere la competizione) sono descritte dal seguente codice, situato all'interno della classe *MainActivity*.

```
public class MainActivity extends AppCompatActivity {
    public TextView textBenvenuto, textSelezione, textTitolo, textSubtitolo;
    public ExtendedFloatingActionButton exFabSpada, exFabSciabola, exFabFioretto;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        textBenvenuto=findViewById(R.id.textBenvenuto);
        textTitolo=findViewById(R.id.textTitolo);
        textSubtitolo=findViewById(R.id.textSubtitolo);
        textSelezione=findViewById(R.id.textSelezione);
        exFabSpada=findViewById(R.id.exFabSpada);
        exFabSciabola=findViewById(R.id.exFabSciabola);
        exFabFioretto=findViewById(R.id.exFabFioretto);
        exFabSpada.setOnClickListener((view) -> {
            Intent i = new Intent(view.getContext(), ActivityCompetizione.class);
            i.putExtra( name: "nomeCompetizione", value: "Spada");
            startActivity(i);
        });
        exFabSciabola.setOnClickListener((view) -> {
            Intent i = new Intent(view.getContext(), ActivityCompetizione.class);
            i.putExtra( name: "nomeCompetizione", value: "Sciabola");
            startActivity(i);
        });
        exFabFioretto.setOnClickListener((view) -> {
            Intent i = new Intent(view.getContext(), ActivityCompetizione.class);
            i.putExtra( name: "nomeCompetizione", value: "Fioretto");
            startActivity(i);
        });
    }
}
```

Come si evince dal codice, l'interazione con l'utente è gestita dai metodi *setOnClickListener()* che vengono invocati ogniqualvolta l'user interagisce con il layout, cioè clicca uno dei pulsanti disponibili. A questo punto, l'app passerà alla seconda Activity, denominata *ActivityCompetizione*, che riporterà il nome della competizione scelta (fioretto, spada o sciabola). Questa Activity corrisponderà visivamente al layout della seconda schermata dell'app riportata qui di seguito.



Questa schermata permette all'utente, in questo caso l'arbitro, di inserire ulteriori informazioni sulla competizione. I primi due campi corrispondono all'età e al sesso degli atleti e sono facilmente compilabili attraverso un metodo di inserimento dei dati denominato *RadioButton*, che permette all'utente di cliccare sul pallino posto accanto alla categoria selezionata. Il terzo e ultimo campo invece, richiede informazioni che devono essere necessariamente inserite manualmente dall'utente, perché di non facile previsione e riguardano il nome dell'arbitro, la pedana, il numero degli atleti e il girone. Tale modalità di inserimento viene chiamata *EditText* e, come mostra l'immagine riportata in alto, le righe da compilare contengono i suggerimenti delle informazioni da inserire, che scompariranno non appena l'utente inizierà a digitare sulla propria tastiera. La scelta di questo specifico layout non è casuale, anzi, la sua intuitività è volta a guidare l'utente nell'inserimento dei dati, impedendogli di sentirsi disorientato. Il pulsante "crea competizione" non ha solo lo scopo di avviare un'altra Activity, ma ha anche la funzione di controllare che tutti i dati siano stati inseriti e che gli stessi rispettino i requisiti imposti dal programmatore. Solo quando questo controllo avrà ottenuto un esito positivo, sarà possibile passare all'Activity successiva; in caso contrario, apparirà in sovrimpressione un messaggio che informerà l'utente della mancata o errata compilazione dei vari campi di inserimento.

I dati inseriti, non verranno poi persi, ma bensì immagazzinati in modo tale da poter essere riutilizzati in seguito. Il processo attraverso il quale i dati vengono immagazzinati è descritto nel codice seguente.

```
exFab2.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if (controllo()==false){
            Toast.makeText(view.getContext(), "ATTENZIONE: dati Mancanti o errati", Toast.LENGTH_SHORT).show();
        }else{
            numAtleti= Integer.parseInt(ETatleti.getText().toString());
            numAssalti=controllonumAssalti(numAtleti);
            Intent i= new Intent(view.getContext(), ActivityGirone.class);
            i.putExtra( name: "numeroAssalti", numAssalti);
            i.putExtra( name: "Girone", ETgirone.getText().toString());
            startActivity(i);
        }
    }
});
}

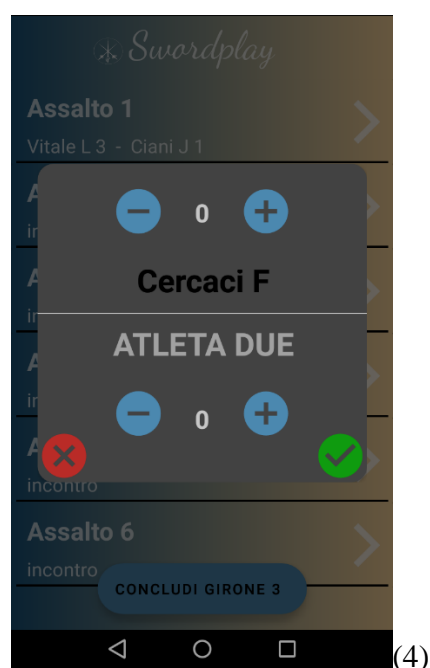
public boolean controllo(){
    competizione=getIntent().getStringExtra( name: "nomeCompetizione");
    boolean controllo;
    if (ETatleti.length()==0 || ETnomeArbitro.length()==0 || ETgirone.length()==0 || ETpedana.length()==0 ||
        radioSesso.getCheckedRadioButtonId()==-1 || radioCategoria.getCheckedRadioButtonId()==-1){
        controllo=false;
    }else{
        risultato= "Ripilogo della Competizione " + competizione +
            "\n\nCATEGORIA: "+rCategoria.getText()+
            "\n\nSESSO: "+rSesso.getText().toString()+
            "\n\nNUMERO ATLETI: "+ETatleti.getText().toString()+
            "\n\nARBITRO: "+ETnomeArbitro.getText().toString()+
            "\n\nGIRONE: "+ETgirone.getText().toString()+
    }
```

Uno dei più significativi tra i dati precedentemente inseriti, è il numero degli atleti, in quanto viene gestito attraverso un metodo a sé, dal nome *controllonumAssalti()*. Esso ha lo scopo di generare il numero degli assalti (ossia gli incontri tra due atleti) che verranno visualizzati nella schermata successiva. Per fare ciò, deve prima di tutto controllare che il numero degli atleti sia superiore a 1 e successivamente calcolare il numero degli assalti utilizzando la formula:

$$num. assalti = \frac{num. atleti * (num. atleti - 1)}{2}$$

Prima di passare all'Activity successiva, tutti questi dati vengono salvati in una struttura chiamata *ArrayList* dichiarata in modo statico, ovvero visibile a tutte le Activity.

La schermata successiva sarà appunto quella che contiene i vari assalti, i quali potranno essere selezionati e compilati.



A questo punto, man mano che si susseguiranno gli assalti, l'arbitro cliccherà su ognuno di essi (3) per poter inserire i nomi degli atleti e i punti assegnatigli. A tale scopo, una volta cliccato su un assalto, si aprirà una *DialogBox*, come illustrato nell'immagine (4). Il metodo di inserimento dei nomi degli atleti è lo stesso della seconda schermata (ossia *EditText*) e anche qui è presente il suggerimento "atleta 1/2" che scomparirà una volta inserito il nome. Per assegnare i punti, l'arbitro potrà cliccare sui tasti "+" ogni volta che un'atleta colpirà un bersaglio valido e sul tasto "-" nel caso in cui venisse erroneamente assegnato un punto. Al termine dell'incontro, l'arbitro potrà cliccare sul tasto *check* in verde, posizionato in basso a destra, e la *DialogBox* scomparirà, riportando alla schermata che contiene la lista degli assalti. A questo punto, i dati precedentemente inseriti nella *DialogBox*, verranno visualizzati al posto della parola "incontro" al di sotto dell'assalto e l'utente potrà passare a quello successivo. Alla base di questa schermata c'è l'*ActivityGirone*, fase centrale e sicuramente più impegnativa nella programmazione di *Swordplay*. Il suo layout è esteticamente lineare e riconoscibile, in quanto si tratta di una *RecyclerView*, riscontrabile anche nelle app più in voga del momento, come YouTube, Instagram, Gmail, Amazon e tante altre. L'implementazione di questa struttura è una delle

principali difficoltà che il programmatore si trova ad affrontare. La RecyclerView è stata implementata per la prima volta nel 2014 con l'uscita della versione *Lollipop* di Android ed è ideale per mostrare all'utente una lista di elementi correlati da informazioni. Con il metodo precedente, denominato *ListView*, nonostante l'utente riuscisse a vedere solo un numero limitato di view, (ossia di layout che contengono gli elementi e le informazioni a riguardo) nella sua schermata, l'app doveva creare e salvare in memoria anche i layout di tutti gli elementi non visibili; se vi era un numero elevato di elementi, però, si rischiava di saturare in poco tempo la memoria del dispositivo e ciò si rifletteva negativamente sulle prestazioni dell'applicazione. La RecyclerView, invece, fornisce un nuovo approccio al problema della creazione delle liste, in quanto permette di “riciclare” il layout di ogni view, cambiando semplicemente le informazioni presenti al suo interno.²⁴ Una parte dell'implementazione della RecyclerView è situata all'interno dell'ActivityGirone.

```
createExampleList();
buildRecyclerView();
onClickGirone(exFab);
}

private void createExampleList() {
    int nAssalti=getIntent().getIntExtra( name: "numeroAssalti", defaultValue: 0);
    nameModels = new ArrayList<NameModel>(nAssalti);
    for(int i=1; i<=nAssalti; i++){
        nameModels.add(new NameModel(R.drawable.ic_baseline_arrow_forward_ios_24,
            textAssalto: getString(R.string.assalto) + i,getString(R.string.incontro)));
    }
}

private void buildRecyclerView() {
    mRelativeLayout=findViewById(R.id.ll);
    mRecyclerView=findViewById(R.id.recyclerview);
    mRecyclerView.setHasFixedSize(true);
    mLayoutManager= new LinearLayoutManager( context: this);
    mAdapter=new ModelAdapter(nameModels);
    mRecyclerView.setLayoutManager(mLayoutManager);
    mRecyclerView.setAdapter(mAdapter);
    mAdapter.setOnItemClickListener((position, a) -> {
        changeItem(position,a);
    });
}

public void changeItem(int position,String text){
    nameModels.get(position).changeText(text);
    mAdapter.notifyItemChanged(position);
}

public void onClickGirone(ExtendedFloatingActionButton exFab){
```

²⁴ RecyclerView:dietro le quinte, A. Tedeschi, html.it, 27 dic. 2018.

La struttura della *RecyclerView* è composta da tre classi Java, rispettivamente *ActivityGirone*, *ModelAdapter* e *NameModel*, e da due file di layout, ovvero *activity_girone* e *layout_row*. Ogni singola classe è determinante e ha un compito ben preciso, ma in questo contesto prenderemo in esame solamente l'*ActivityGirone*, in quanto costituita da elementi che verranno richiamati nelle altre due classi. Essa è formata da tre metodi principali, quali:

- *createExampleList()*: che crea le *CardView* (Assalto 1, Assalto 2...) sulla base del risultato della formula precedentemente menzionata;²⁵
- *buildRecyclerView()*: il quale crea la struttura della *RecyclerView*, richiamando la classe *ModelAdapter*, la quale contiene anche la funzione che permette la comparsa della *DialogBox*.
- *onClickGirone()*: che segna la conclusione del girone e contiene l'*Intent* che consente il passaggio all'*Activity* successiva.

L'ultimo metodo descritto, si attiva quando l'utente clicca sul pulsante "concludi girone". Questa tipologia di pulsante viene denominata *Floating Action Button*, in quanto, visivamente, non appare "fisso" su una superficie ma sembra "galleggiare" sospeso nell'aria; la motivazione che ha spinto all'utilizzo di questo *material design* concerne i concetti argomentati nel primo capitolo, riguardanti gli elementi che contribuiscono a stimolare la corteccia laterale premotoria del cervello.²⁶

Una volta cliccato su questo pulsante, l'app condurrà l'utente all'ultima schermata, che contiene il riepilogo dei dati inseriti in fase di inizio e dei risultati dei vari assalti. Di seguito, un esempio di come potrebbe presentarsi la suddetta schermata, con un inserimento casuale dei dati a fini dimostrativi.

²⁵ Vedere p. 19.

²⁶ Vedere cap. 1, par. 1.1, p. 4



Nel riepilogo, subito dopo il nome e il logo dell'app che accompagnano l'utente in ogni schermata, è riportato, nella prima riga, il nome della competizione scelta (in questo caso Fioretto) e di seguito le informazioni sui dati inseriti nella prima schermata di compilazione, ovvero la categoria (che corrisponde all'età degli atleti, in questo caso Giovani dai 17 ai 20 anni), il sesso degli atleti, il loro numero, il nome dell'arbitro, il girone e la pedana. Ad essi, seguono i nomi degli schermidori con i rispettivi punteggi, inseriti dall'arbitro nella DialogBox derivante dalla terza schermata (immagine n.4). Anche qui è presente un Floating Action Button, posizionato nella parte inferiore della schermata, ma la sua funzione verrà spiegata successivamente. Prima, è opportuno illustrare l'Activity alla base di questa schermata, il cui nome è inequivocabilmente *ActivityRiepilogo*.

```

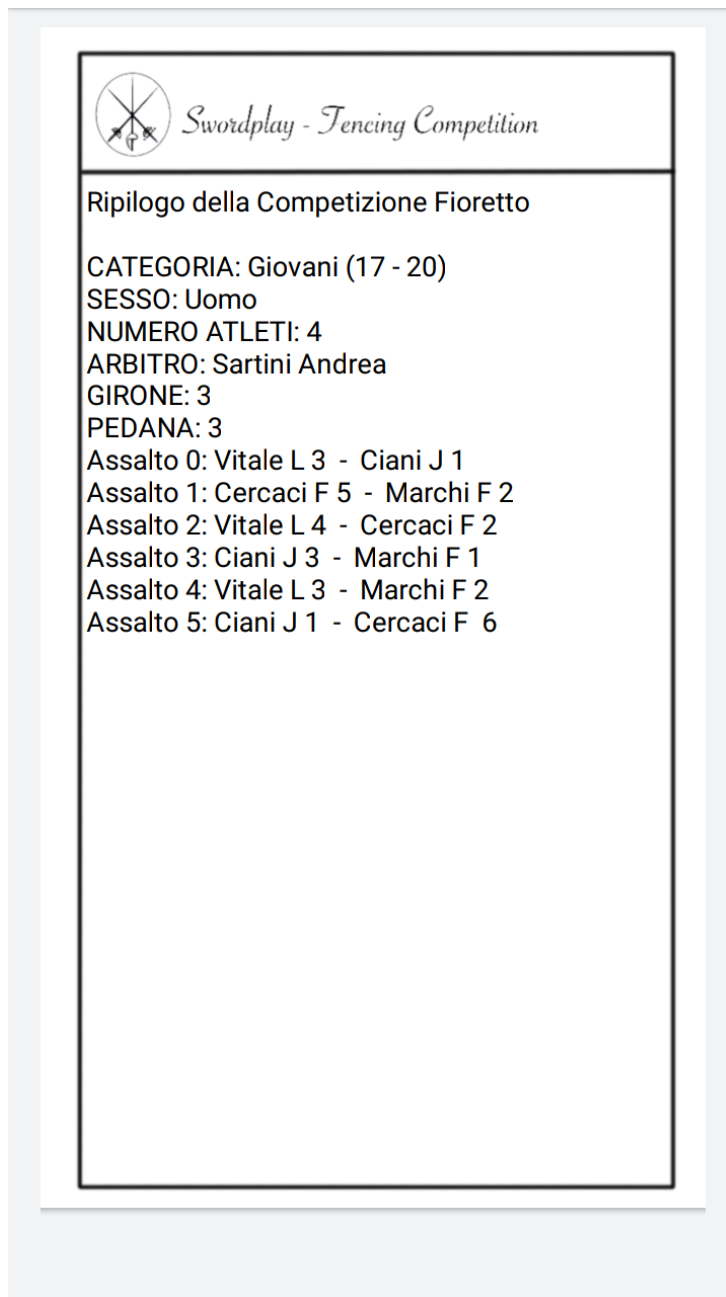
public void createPDF() {
    File f=null;
    int c = 0;
    boolean creato = false;
    while (!creato) {
        String myFilePath = String.format(Environment.getExternalStorageDirectory().getPath() + "/swordplay" + c + ".pdf");
        try {
            f=new File(myFilePath);
            creato=f.createNewFile();
            c++;
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
    PdfDocument mypdfDocument = new PdfDocument();
    PdfDocument.PageInfo mypageInfo = new PdfDocument.PageInfo.Builder( 720, 1280, 1).create()
    PdfDocument.Page mypage = mypdfDocument.startPage(mypageInfo);
    Paint mypaint = new Paint();
    btm = BitmapFactory.decodeResource(getResources(), R.drawable.pdf);
    scaledbmp = Bitmap.createScaledBitmap(btm, dstWidth: 720, dstHeight: 1280, filter: false);
    mypage.getCanvas().drawBitmap(scaledbmp, left: 0, top: 0, mypaint);
    //String myString = "questa è la \n nuova stringa";
    String myString = "" + leggi(info);
    mypaint.setTextSize(30);
    int x = 50, y = 200;
    for (String line : myString.split( regex: "\n")) {
        mypage.getCanvas().drawText(line, x, y, mypaint);
        y += mypaint.descent() - mypaint.ascent();
    }
}

```

In questa classe viene richiamato l'*ArrayList*, denominato in questo caso *Info*, grazie al quale è possibile ripescare tutte le informazioni inserite finora, affinché possano essere mostrate all'utente attraverso la schermata di Riepilogo. Se l'app venisse chiusa in questo momento, però, tutti i dati inseriti andrebbero persi e il suo utilizzo non avrebbe alcuna utilità; a tal proposito, è indispensabile che lo sviluppatore inserisca nell'app un metodo per far sì che l'user possa in qualche modo scaricare i dati fino a quel momento raccolti; ed è proprio qui che entra in gioco il Floating Action Button visibile nella parte inferiore della schermata di riepilogo.



L'immagine all'interno del bottone è la tipica icona riscontrabile in tutte le app per indicare la funzione "scarica" e, infatti, permette di scaricare un file in formato PDF che, non solo contenga tutti i dati, ma che sia anche curato esteticamente, senza tralasciare la specificazione dell'app di provenienza. Di seguito, un esempio di documento scaricabile dall'app.



Prima di scaricare il riepilogo in PDF utilizzando il metodo *createPDF* (visibile nella prima riga dell'immagine), una volta cliccato il bottone, l'app chiederà all'utente i permessi per leggere e scrivere sulla memoria interna del dispositivo; questi permessi, come spiegato precedentemente, sono stati inseriti nel Manifest agli inizi della programmazione e vengono qui richiamati attraverso il metodo *requestStoragePermission()*. Il file verrà scaricato (e quindi il metodo *createPDF()* eseguito) solo ed esclusivamente se l'utente accetterà di dare i permessi, che gli verranno richiesti con la comparsa di una *AlertBox*.

Infine, l'app ritornerà alla schermata iniziale, comunicando all'utente l'avvenuto salvataggio del file attraverso un *Toast*, una sorta di “messaggio in sovraimpressione” che specifica il nome del file e il posto in cui è stato salvato.



Per evitare che i nuovi file vengano sovrascritti ai precedenti o che siano di difficile reperimento, ogni file salvato avrà nel nome un numero che seguirà l'ordine di download (per esempio, il primo file sarà denominato *swordplay.Pdf* come riportato nell'immagine, il secondo *swordplay1.Pdf*, il terzo *swordplay2.Pdf*, e così via).

2.2 Beyond *Swordplay*: pubblicazione e sviluppi futuri

Nonostante l'app sia stata ideata e sviluppata principalmente ai fini di questa tesi di laurea, non mancano i progetti futuri. I problemi, citati all'inizio del lavoro, nell'annotazione dei punteggi, sono effettivamente esistenti e spinosi e quest'app potrebbe davvero rappresentare una risoluzione pratica e concreta. La collaborazione con un arbitro

ufficiale di scherma permetterebbe di effettuare test sul campo e rilevare feedback che potrebbero condurre all'analisi statistica dei dati registrati, per valutare vantaggi e svantaggi e ispirare possibili miglioramenti.

L'utilizzo concreto dell'app nelle gare spingerebbe inevitabilmente lo sviluppatore alla pubblicazione della stessa sullo store ufficiale di Android, ovvero il *Google Play Store*, come in tutti gli altri store alternativi.

La programmazione di quest'app non ha solo a che fare con l'informatica, perché i suoi sviluppi potrebbero favorire una sinergia tra diversi ambiti, tra cui quello della traduzione, in quanto basterebbe tradurre la parte scritta contenuta nel file *strings.xml* (contenuto nei *values* della cartella *res*) in diverse lingue, affinché possa essere usata anche nelle competizioni internazionali. In più, degli esperti di marketing potrebbero curarne gli aspetti inerenti alla sponsorizzazione e a ciò che potrebbe renderla più accattivante per il pubblico.

Inoltre, un obiettivo ancora più alto sarebbe quello di presentare l'iniziativa alla Federazione Italiana di Scherma (FIS), non solo illustrando le funzionalità dell'app ma anche redigendo un report in cui documentare i dati ottenuti dalla sperimentazione sul campo e i feedback degli utilizzatori.

I progetti per il futuro potrebbero essere tanti, alcuni più difficilmente realizzabili di altri, specialmente in questa situazione di emergenza sanitaria che ha messo in pausa il mondo dello sport come tanti altri; ma uno dei primi step da cui partire potrebbe essere appunto la traduzione in inglese, considerato lingua franca e quindi comprensibile da parlanti delle più svariate nazionalità.

In ogni caso, il punto focale di questo paragrafo è dimostrare, con piccoli spunti di riflessione, come questa semplice app potrebbe costituire una buona base di partenza per un progetto ben più ampio e di larghe vedute.

CONCLUSIONE

Questo lavoro ha avuto lo scopo di presentare il mondo delle app nel suo complesso, partendo dalla loro storia, da come e quando sono state inventate e seguendone lo sviluppo fino al giorno d'oggi, in cui sono diventate praticamente imprescindibili nella vita di tutti i giorni.

Ha voluto, inoltre, illustrare le caratteristiche che un'applicazione dovrebbe avere affinché abbia successo, argomentandone le motivazioni e mettendole in pratica sperimentalmente con l'app ideata ai fini di questa tesi di laurea.

La progettazione e lo sviluppo dell'app in questione, denominata *Swordplay – Fencing Competitions*, sono stati illustrati e commentati passo per passo, documentando poi le funzionalità dell'app conclusa, attraverso gli screenshot delle schermate con cui l'utente si trova ad interagire e dei codici che sottendono a tali visualizzazioni; per di più, non mancano le motivazioni che hanno condotto all'ideazione di quest'app specifica, dedicata all'annotazione dei punti nelle competizioni schermistiche, che viene proposta come risoluzione alle problematiche legate alle metodologie utilizzate a tal proposito, citando e chiarendo anche le regole principali di tale sport, affinché anche il lettore meno esperto nell'argomento possa vederne limpidamente i vantaggi.

Infine, si coglie l'occasione per utilizzare questo progetto come punto di partenza per piani futuri, argomentando i possibili risvolti che un'ipotetica pubblicazione potrebbe generare e gli ambiti che essa potrebbe coinvolgere.

Sitografia

Snapchat: come le app influenzano la vita quotidiana e le emozioni, Skytg24,
<https://tg24.sky.it/tecnologia/2019/01/11/snapchat-app-influenzano-vita-quotidiana-emozioni>

La nascita dello Smartphone e l'espansione delle mobile app, L. Diana, HDblog.it
mobile <https://mobile.hdblog.it/2017/05/26/nascita-smartphone-espansione-mobile-app/>

7 regole per realizzare un'app di successo, D. Mangini, Wired.it,
https://www.wired.it/economia/business/2017/03/24/7-regole-app-successo/?refresh_ce=

Federazione Italiana Scherma, La scherma, *La Storia*, Federscherma.it,
<https://www.federscherma.it/homepage/la-scherma/storia.html>

Federazione Italiana Scherma, La scherma, *Le Armi*, Federcherma.it,
<https://www.federscherma.it/homepage/la-scherma/le-armi.html>

Regolamento tecnico per le gare di scherma, FIE, Gruppo Schermistico Arbitrale,
Schermalazio.it <http://www.schermalazio.it/Comunicati/Comunicati%202018-2019/Formazione/Regolamento%20Tecnico%20Italiano%20Dicembre%202017.pdf>

Che cos'è la tecnologia Java e a cosa serve?, Java.it,
https://www.java.com/it/download/help/whatis_java.html

Informazioni sulla tecnologia Java, Java.it, <https://www.java.com/it/about/>

The Java Programming Language, tiobe.com, <https://www.tiobe.com/tiobe-index/java/>

About Company, tiobe.com, <https://tiobe.com/company/about/>

5 motivi per apprendere il linguaggio Java, R. Fiore, fiorerosalba.com
<https://www.fiorerosalba.com/5-motivi-per-apprendere-linguaggio-java/>

10 motivi per cui una certificazione Java è importante per trovare lavoro, M. Masumeci, Nextre Engineering, 3 luglio 2018, <https://www.nextre.it/certificazione-java-per-lavoro/>

Meet Android Studio, User Guide, developer.android.com,
<https://developer.android.com/studio/intro>

IntelliJ IDEA, download.html.it, <https://download.html.it/software/intellij-idea/>

JetBrains Corporate Overview, sezione “About”,
https://resources.jetbrains.com/storage/products/jetbrains/docs/corporate-overview/en-us/jetbrains_corporate_overview.pdf?_gl=1*_pepu9r*_ga*MTE1MjU4NjQ4Ny4xNjA0NDg5NjU3*_ga_V0XZL7QHEB*MTYwNDQ4OTY1Ny4xLjEuMTYwNDQ4OTY3NC4w&_ga=2.77761373.898717074.1604489657-1152586487.1604489657

Enciclopedia della Scienza e della Tecnica, Treccani online, *codice sorgente*,
https://www.treccani.it/enciclopedia/codice-sorgente_%28Enciclopedia-della-Scienza-e-della-Tecnica%29/

Informazioni su Android, *Fatti*, android.com,
https://www.android.com/intl/it_it/everyone/facts/

Perché scegliere uno smartphone Android: 8 validi motivi, redazione Focustech, 31 mag. 2019, focustech.it, <https://focustech.it/2019/05/31/perche-scegliere-uno-smartphone-android-8-validi-motivi-246566>

App Manifest Overview, Guides, developer.android.com
<https://developer.android.com/guide/topics/manifest/manifest-intro>

The concept of activities, Introduction to Activities, Guides, developer.android.com,
<https://developer.android.com/guide/components/activities/intro-activities>

Il ciclo di un'activity, G. Maggi, html.it, 22 giu. 2020 <https://www.html.it/pag/48652/il-ciclo-di-vita-di-unactivity/>

Intents and Intent Filters, Guides, developer.android.com
<https://developer.android.com/guide/components/intents-filters>

Gestire le risorse e gli asset, G. Maggi, html.it, 1 lug. 2020,
<https://www.html.it/pag/48658/gestire-le-risorse-e-gli-asset/>

RecyclerView:dietro le quinte, A. Tedeschi, html.it, 27 dic. 2018,
<https://www.html.it/articoli/recyclerview-dietro-le-quinte/>

Ringraziamenti

Eccomi quindi giunto alla fine di questo inaspettato percorso, non lo avrei mai immaginato e tutt'ora fatico a crederci.

Prima di tutto, vorrei dedicare questo grande traguardo ai membri della mia famiglia, colonna portante di ciò che sono e che sarò, coloro che mi hanno sempre sostenuto qualsiasi decisione io prendessi e che mi hanno sempre accompagnato in tutte le mie folli avventure; grazie per aver sempre avuto fiducia in me.

A mia madre, la donna più forte che io conosca, che con la sua travolgente forza e una dolcezza che solo una mamma può avere, mi ha insegnato ad amare e rispettare la persona che avrò al mio fianco e ad affrontare il mondo nel modo giusto, un passo alla volta e senza avere paura di niente e di nessuno; a te devo la mia determinazione, sarai per sempre la mia forza e il mio dolce rifugio.

A mio padre, l'uomo che spero di diventare, sempre pronto ad essere al mio fianco per accompagnarmi in qualsiasi parte del mondo, che cerca in ogni occasione d'infondermi la calma e la tranquillità di cui ho bisogno. Tu mi insegni che nella vita non si smette mai d'imparare e che con l'impegno e il duro lavoro si può raggiungere qualunque obiettivo; a te devo la mia calma e il mio essere riflessivo, per me sarai sempre quel sospiro di sollievo che mi porta serenità.

A mia sorella, la donna che ammiro, nonché la mia più grande certezza. Sapere di poter contare in qualunque momento su di te è una fortuna inspiegabile, per te sarò sempre quel piccolo Vito da coccolare e strapazzare. Sei sempre stata presente nei miei pensieri, anche quando non lo avresti mai immaginato. Attraverso il tuo coraggio e la tua caparbia m'insegni che la vista mozzafiato che troverò sulla vetta della montagna vale tutti i sacrifici e le difficoltà che si incontrano scalandola. È grazie a te se credo in me stesso e a te devo il mio coraggio e la mia intraprendenza.

Inoltre, vorrei ringraziare tutti coloro che mi sono stati accanto durante questi anni.

Ringrazio Francesco e Pasquale, che considero più come fratelli che come amici; accanto a me, sono stati pronti a partire, spinti dalla voglia di cercare se stessi e hanno reso questa esperienza un'avventura e "casa di Nadia" indimenticabile.

Ringrazio Domenico, Marco, Roberto, Alessia e Ciro. Ognuno di voi, nel suo piccolo, è stato determinante e mi ha dato la spinta necessaria per raggiungere tale obiettivo; anche se lontani fisicamente, mi siete stati più vicini di quanto possiate immaginare e non mi avete mai fatto sentire solo nelle difficoltà.

Infine, un ringraziamento speciale è doveroso all'unica persona che è stata in grado di sopportarmi per questi anni senza mai stancarsi, ad una persona che è riuscita a trasformare i miei difetti in pregi, una persona che ha creduto in me fin dall'inizio e non ha mai smesso di farlo, l'unica persona con cui so di poter essere me stesso e che si è rivelata essere un'amica, una confidente, una coinquilina, una cantante, una correlatrice, oltre che la mia ragazza. Federica, hai dato a tutto una forma ed un colore meraviglioso, grazie.