

# RPG Combat Manager: app multiplatforma per la gestione dei combattimenti in D&D/Pathfinder

Francesco Gradi  
Matricola: 7017190  
francesco.gradi@stud.unifi.it

Federico Målato  
Matricola: 7017325  
federico.malato@stud.unifi.it

## Abstract

*"RPG Combat Manager" è un'app "da giocatori, per giocatori", nata per venire incontro alle esigenze degli appassionati di giochi di ruolo che da anni combattono contro ogni sorta di creatura mitologica e contro i dadi: una fase fondamentale del combattimento è infatti il calcolo dei danni, capace di raggiungere complessità ragguardevoli. Con questo progetto abbiamo tentato di migliorare l'esperienza del giocatore, sia esso novizio o esperto, automatizzando i calcoli necessari per lo svolgimento del turno. L'app è stata progettata con Ionic e Angular Material, testata su dispositivi mobili e tablets Android/iOS. I test svolti hanno indicato il raggiungimento di un risultato molto soddisfacente, sebbene ci sia ancora spazio per alcuni miglioramenti.*

## 1. Introduzione

### 1.1. Dungeons&Dragons

*Dungeons&Dragons* è un gioco di ruolo fantasy, da tavolo, che è stato ideato negli anni '70 del secolo scorso da Gary Gygax e Dave Arneson e ha conquistato i nerd di tutto il mondo, grazie all'enorme profondità e alle infinite possibilità di campagne leggendarie, tutte lasciate all'immaginazione dei giocatori [1].

Il gioco classico consiste nel ruolare personaggi immaginari, le cui sorti sono gestite non solo dalla bravura del giocatore, ma anche dalla fortuna dovuta al lancio di particolari dadi (il più iconico è quello a 20 facce, il d20). I personaggi hanno una determinata classe, che rappresenta le loro specifiche abilità, e sono caratterizzati da statistiche appuntate su una *scheda giocatore*, la quale contiene tutto il necessario per esplorare, combattere e effettuare incantesimi nel nostro mondo immaginario. Tale mondo è descritto e plasmato da un *Dungeon Master*, il quale si occupa di definire la storia, di interpretare gli NPC (i personaggi non giocanti) e, naturalmente, di gestire i combattimenti con i mostri

che il party di avventurieri incontrerà nel cammino verso la prossima ricompensa.

Si sono susseguite negli anni diverse edizioni, sempre più articolate e complesse, anche se l'ultima (5.0) ha cercato di semplificare una quantità di offerta davvero esagerata. Nonostante questo, l'edizione probabilmente più di successo e più giocata resta la 3.5, la quale garantisce un'esperienza appagante soprattutto per i giocatori più hardcore. Da una costola proprio di questa edizione, ma da un diverso editore, è nato l'universo Golarion con *Pathfinder*, che ha cercato di risolvere alcuni problemi patologici di tale edizione, senza stravolgere troppo le meccaniche di gioco [2].

La scheda giocatore è interamente gestita su carta e può essere davvero complicata, è necessario un po' di sforzo da parte del giocatore per ricordare tutte le regole (che sono moltissime) e per applicare correttamente i modificatori da applicare al dado per effettuare una determinata azione. Per esempio, se un Guerriero di livello 3 vuole colpire un Ogre con la propria spada dovrà tirare un *d20* e aggiungere al risultato un modificatore che dipende dal proprio punteggio di forza: soltanto se il risultato complessivo supera la classe armatura dell'Ogre, si procederà al calcolo dei danni. All'interno del gioco ci sono decine di modificatori che vengono applicati di volta in volta in base all'azione attuale, il gioco si intuisce subito che è complicato, ma è un male necessario per poter giocare (la curva di apprendimento è piuttosto ripida...).

Come se non bastasse, durante una battaglia, possono essere lanciati diversi incantesimi, utilizzati talenti oppure oggetti che cambiano le statistiche e i modificatori attuali dei personaggi: questi vengono chiamati *buff*. I *buff* possono consentire a un personaggio debole di affrontare nemici molto più forti di lui e in generale sono un espediente che rende le battaglie molto più interessanti, ma sono estremamente complicati, soprattutto perché il loro numero cresce esponenzialmente con il passare dei livelli.

Il problema di base fondamentale identificato e il

motivo di questo progetto risiede proprio qui: la difficoltà del calcolo dei buff temporanei e non, a *run time*, durante un combattimento. Un'applicazione che aiuti un giocatore a gestire in tempo reale i vari buff attivi avrebbe vari vantaggi, come velocizzare il calcolo durante i combattimenti, evitare errori e consentire ai giocatori concentrarsi più sul combattimento che sul calcolo dei modificatori attuali.

I giocatori più tradizionalisti disdegneranno l'uso dello smartphone, ma l'idea dell'applicazione è quella di fornire soltanto un supporto per il combattimento, non ha lo scopo un po' blasfemo di sostituire e digitalizzare la scheda giocatore tradizionale.

## 1.2. Utilizzo Mobile

La gestione durante una fase di combattimento può essere naturalmente effettuata tramite uno smartphone o al più un tablet, non certo su un computer. Per questo lo sviluppo dell'app è stato studiato appositamente per dispositivi mobili; inoltre, la caratterizzazione dell'applicazione prevede uno sviluppo più improntato alla parte *front-end* rispetto al *back-end*, per questo il linguaggio comunemente più usato è il *javascript*, anche nelle sue varianti a seconda dei vari framework, specie se, come in questo caso, è previsto un supporto multiplatforma.

## 1.3. Material Design

Per quanto riguarda l'interfaccia grafica vera e propria, l'idea di base è stata quella di cercare di utilizzare le direttive di *Material Design*, secondo il paradigma creato da Google, al fine di ottenere una grafica pulita e moderna in grado di garantire un'esperienza d'uso gradevole [3]. Le applicazioni di Google sono uno degli esempi migliori di interfaccia grafica che abbiamo attualmente, la loro semplicità e usabilità garantiscono un'esperienza sempre soddisfacente soprattutto grazie all'integrazione perfetta con i moderni smartphone.

Una GUI in *Material Design* si basa sulla metafora del foglio di carta, mediante superfici che sembrano tangibili (e in qualche modo lo sono, attraverso il touch screen del nostro smartphone) e forniscono all'utente indicazioni ulteriori che riguardano la semantica dell'applicazione stessa. Le ombre e l'effetto di profondità garantisce un modello mentale dell'applicazione ancora più intuitivo per dell'utente.

## 1.4. Contenuto del Progetto

Il progetto si sviluppa in più fasi, ciascuna volta a raggiungere determinati obiettivi parziali:

- **Needfinding:** sono individuate le necessità degli utenti riguardo l'argomento, mediante interviste

ad hoc con specifiche domande aperte. A seguito sono definite alcune *personas*, cioè personaggi tipici a cui è destinato l'utilizzo dell'applicazione finale. Le *personas* create sono inserite in scenari tipici per mettere a fuoco i diversi casi d'uso possibili. Un semplice *mockup* dell'app ha, in seguito, definito alcune caratteristiche principali volute.

- **Sviluppo dell'applicazione:** si crea l'app da zero a partire dalle tecnologie e frameworks sopra citati.
- **Test di usabilità:** una volta terminata l'iterazione di sviluppo, si cerca di verificare l'usabilità dell'applicazione mediante un test con cui è richiesto di effettuare alcune operazioni possibili, seguito da un breve questionario a 7 livelli di gradimento.

## 2. Needfinding

Nel processo di sviluppo di un'applicazione il primo passo è sicuramente rappresentato dal cosiddetto *Needfinding*: durante questo step l'intenzione degli sviluppatori è riuscire a comprendere quali sono le tipologie di utenti (definite in linguaggio tecnico *Personas*) da cui poter estrapolare una primitiva idea di progettazione dell'applicazione, oltre ad una serie di scenari e casi d'uso possibili.

Nel nostro progetto questa fase di progettazione è stata completata usando delle interviste in due differenti modalità: la prima è stata intervistare alcuni individui mediante un colloquio orale, mentre in un secondo momento è stato posto un questionario agli utenti di alcuni gruppi social.

### 2.1. Colloqui

I colloqui hanno riguardato potenziali utenti di età compresa tra i 22 e i 28 anni, tutti giocatori attivi di *Dungeons&Dragons* e/o *Pathfinder*, e hanno avuto come obiettivo la definizione chiara e completa degli aspetti fondamentali di un'applicazione di tal genere. Le risposte sono state soddisfacenti ed in alcuni casi molto dettagliate, fornendoci l'opportunità di meditare a lungo e soprattutto in maniera profonda sull'identificazione di quelle che sarebbero dovute essere le fondamenta dell'applicazione. Dai colloqui abbiamo appreso che l'idea di semplificare i calcoli necessari per lo svolgimento del turno ha riscosso molto successo tra tutti i candidati e che perciò fosse nostro preciso compito semplificare il più possibile tale procedura, senza però perdere in efficacia o chiarezza (è infatti molto semplice confondersi e non identificare la provenienza di un certo buff); inoltre, abbiamo capito

che i giocatori alle prime armi o comunque con meno esperienza desiderassero uno strumento intuitivo che li aiutasse ad organizzare i numerosi concetti da tenere a mente, o che per lo meno alleggerisse il carico di informazioni da tener presenti durante i combattimenti.

## 2.2. Questionario

In un secondo momento, usando Google Forms [4] abbiamo cercato di ampliare e raffinare il nostro punto di vista e le nostre idee offrendo la possibilità di rispondere un questionario articolato in sette domande aperte agli utenti di alcuni gruppi a tema Dungeons&Dragons su Facebook [5]. È stato molto interessante vedere come, sebbene le domande poste agli utenti fossero pressoché le stesse domande che erano state poste agli intervistati durante i colloqui, le risposte fossero molto differenti: in particolare, le risposte forniteci dagli utenti dei gruppi Facebook risultavano in generale più concise e generiche. Tali risposte ci hanno comunque consentito sia di rafforzare la nostra idea di quali punti fossero veramente fondamentali per l'applicazione, sia di definire altre importanti funzionalità secondarie. Inoltre, non avendo trovato gruppi che trattassero di una specifica versione del gioco, abbiamo notato che gli utenti che giocano alla versione 3.5 o precedente di D&D sono in media più interessati al concept dell'applicativo rispetto ai giocatori delle versioni 4.0 o 5.0, proprio in virtù del fatto che tali versioni hanno introdotto numerose semplificazioni nel gioco.

## 2.3. Personas

Dai dati raccolti abbiamo potuto identificare due Personas, sulle quali si è basata la progettazione:

- **Arturo** (26-28 anni) è il giocatore d'esperienza, che gioca dalla versione 1.0 del gioco per cui la sessione settimanale rappresenta la messa, da attendere con la massima devozione e rispetto e per cui il Master, oltre a moderare il gioco, è anche una guida che attinge la sua infinita conoscenza direttamente dalle Sacre Scritture, ossia i Manuali di Gioco.
- **Bernardo** (22-24 anni) ha una viscerale passione per tutto ciò che riguarda i GDR ma, complici la maggior parte dei suoi amici che hanno altri interessi e la mancanza di tempo propria dello studente universitario-lavoratore, da poco si è affacciato al mondo di D&D/Pathfinder. Proprio questa mancanza di tempo rendono la sua scalata alla conoscenza completa del gioco lenta e complessa, per cui cerca di ovviarvi con gli espedienti più vari e fantasiosi che comprendono

(senza pretesa di esaustività) fogli di carta, post-it appesi sulla scheda del giocatore, memo sullo smartphone, svariati documenti salvati su servizi di storage online e una discreta quantità di segnalibri sul browser.

Il concept dell'applicazione è dunque il seguente: se da un lato essa vuole rendere agevole il turno di combattimento al giocatore che sa quel che fa, dall'altro non vuole trascurare l'avventuriero novizio, ma anzi aiutarlo nella comprensione del gioco.

## 2.4. Scenari e Requisiti

Grazie alle interviste condotte e alle risposte ricevute, abbiamo potuto identificare alcuni scenari riferiti alle Persona individuate. Tali scenari ci hanno poi aiutato molto nel definire i requisiti dell'applicazione. Alcuni tra gli scenari trovati sono:

- Arturo controlla un personaggio di livello medio/alto, con un'inclinazione verso il ruolo di supporto per il party e, nel momento in cui inizia il combattimento, inizia a "buffare" gli alleati e "debuffare" i nemici, generare condizioni atmosferiche favorevoli per il combattimento, applicare effetti ed aure a coloro che gli stanno intorno e così via. Il suo party, dopo aver ricevuto il quindicesimo buff, inizia a rallentare il ritmo di gioco a causa dei numerosi calcoli da svolgere per completare il proprio turno d'attacco. A quel punto si comincia a pensare che avere uno strumento automatico che sommi i vari effetti potrebbe ridurre la durata del combattimento da quattro sessioni a un'ora;
- Bernardo, a metà del combattimento, si rende conto di non aver sommato i buffs del bardo del suo party, mancando ogni volta il bersaglio al tiro per colpire. Dopo aver avvisato i suoi compagni (generando bonaria ilarità) si rende conto che avrebbe proprio bisogno di uno strumento che lo aiuti a ricordare tutto ciò che c'è da sommare ai suoi tiri ad ogni turno, perché quei calcoli proprio non gli entrano in testa;
- Bernardo, che ha voluto provare una classe magica per la prima volta, ha una lunga serie di buffs che il suo personaggio può lanciare ed usare durante il gioco ma non si ricorda metà di questi e dell'altra metà non ricorda gli effetti, fatta eccezione per quei 3-4 incantesimi che vengono usati con costanza. Per migliorare l'esperienza di gioco sua e del party nelle varie situazioni, vorrebbe avere una panoramica schematica e intuitiva dei

suoi buffs, senza dover andare ogni volta a ricercare gli effetti nei manuali e facendo perciò perdere tempo a tutto il party;

- Arturo segue più campagne contemporaneamente e vorrebbe avere un modo immediato per ottenere una panoramica di ciascun suo personaggio in modo intuitivo e veloce, senza perdersi nei dettagli della scheda del giocatore, che lo possa aiutare durante le sessioni con i vari gruppi di gioco.

Dati questi (e tanti altri) scenari, abbiamo ritenuto che i cardini della nostra applicazione dovessero essere i seguenti:

- L'applicazione deve fornire una panoramica completa ed intuitiva di tutti i valori necessari allo svolgimento di un combattimento;
- L'utente deve poter aggiungere ed eliminare rapidamente un buff o un pezzo di equipaggiamento;
- Si deve lasciare la possibilità agli utenti di attivare e disattivare i buffs (o i pezzi d'equipaggiamento) tramite la semplice pressione di un tasto;
- Gli utenti devono poter gestire più personaggi in maniera completamente separata, senza intaccare la semplicità dell'interfaccia e fornendo statistiche personalizzate per ciascun personaggio salvato;
- È necessario dare la possibilità all'utente di consultare in maniera semplice ogni buff presente su un determinato personaggio.
- Molti giocatori possiedono più personaggi, è emersa la necessità di offrire la memorizzazione di più characters.

Data inoltre la grande quantità di modelli e marche di dispositivi mobili disponibili, abbiamo puntato su degli strumenti che ci permettessero di utilizzare l'applicazione sia su dispositivi Android che Apple e che funzionasse sia su smartphone che su tablet.

## 2.5. Mock-up

L'individuazione dei principi sopra descritti ha portato in maniera quasi naturale ad un primo sketch (o mock-up) dell'applicazione, che tenesse conto di quanto stabilito e che enfatizzasse il concetto di semplicità ed utilità dello strumento, anche attraverso uno stile minimale ed ordinato dell'interfaccia. La Figura 1 rappresenta il primo concept grafico della nostra applicazione, da cui è possibile vedere anche le differenze rispetto al prodotto finale.

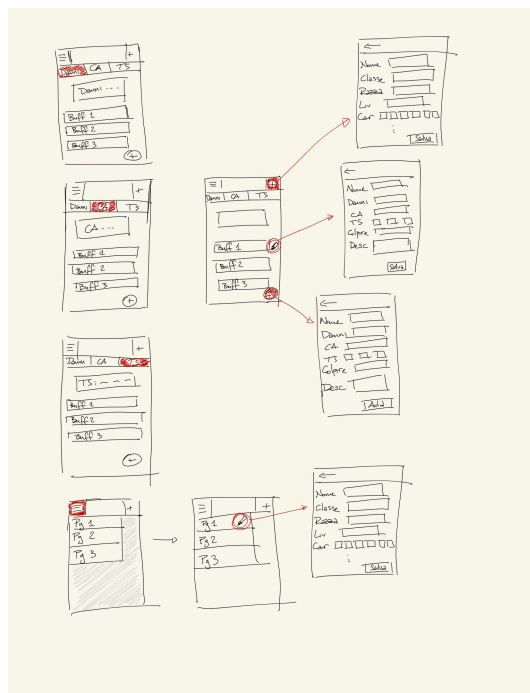


Figure 1. Il primo mock-up dell'applicazione.

## 3. Sviluppo dell'Applicazione

### 3.1. Tecnologie Utilizzate

È stato utilizzato un framework per la conversione di codice Javascript per Android e iOS (quindi codice multiplatforma), chiamato *Ionic* [6]. Il codice dell'applicazione vera e propria è stato sviluppato mediante il framework di Google *Angular*, il quale gestisce le chiamate tipiche di un sistema *Model View Controller* tramite il linguaggio *Typescript* e delle viste tipiche del mondo web scritte in *html* [7].

A livello pratico, le interfacce (tra cui bottoni, tabs, steppers, input...) sono stati resi a schermo grazie ai modelli di *Angular Material*, i quali hanno permesso un'integrazione con il framework completo abbastanza indolore [8].

La conversione del progetto da Angular per i sistemi operativi Android e iOS è stata effettuata tramite *Cordova*, un software in grado di convertire del tipico codice web in specifico codice mobile [9]. Esso aggiunge informazioni per lo smartphone, come ad esempio l'icona o lo splash-screen, attraverso un file di configurazione *config.xml*

### 3.2. Progettazione del Database

Il primo passo è stato definire un database nel quale salvare i vari *Characters*. Ciascun personaggio deve contenere tutte e sole le informazioni necessarie

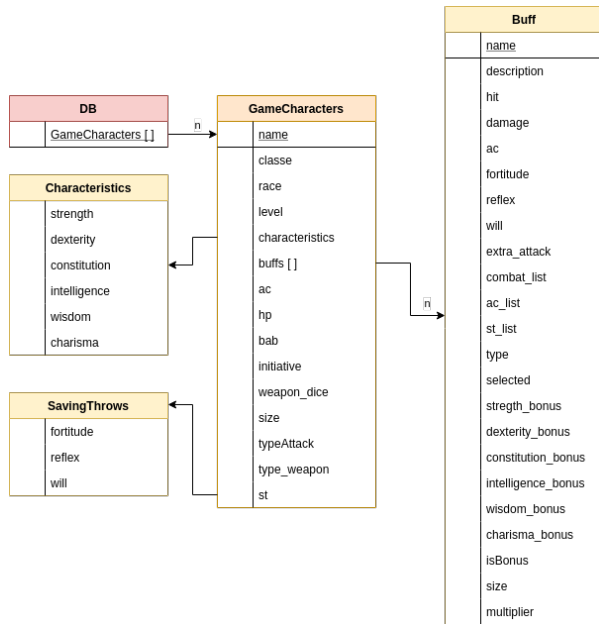


Figure 2. Lo schema generale del database.

alla gestione del combattimento, se avessimo aggiunto troppe informazioni avremmo rischiato di perderci tra le innumerevoli statistiche esistenti; al contrario non avremmo realizzato un'app in grado di gestire in modo preciso la maggior parte dei buff.

Il Characters, oltre alle informazioni generali come il nome, comprende quindi le caratteristiche, i tiri salvezza base, la classe armatura, il bonus attacco base, la taglia del personaggio e, ultimo ma non meno importante, la lista dei *Buff*, il cuore dell'applicazione.

Ogni Buff comprende diversi campi utili, si può potenziare ad esempio i tiri per colpire o i danni, la difesa, i tiri salvezza, fornire un attacco in più oppure potenziare una caratteristica base.

Il database in sé è gestito mediante il sistema nativo del framework *Ionic Storage*, il quale consente di salvare con relativa semplicità i personaggi mediante un file *JSON* [10].

### 3.3. Angular e Model View Controller

Un'applicazione Angular è suddivisa in componenti, ciascuna può essere una pagina web. Ogni componente è formata da un file di *import*, il file *html* con *scss*, i quali rappresentano la *View*, mentre il *Controller* è separato e inserito nel file *.ts* e rappresenta il fulcro della logica della componente.

In Angular la separazione tra Modello e Controller non è però così evidente: il modello infatti è sincronizzato con il database e mantenuto in memoria nel file principale di *typescript*, il quale contiene però anche

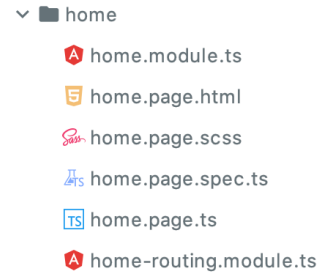


Figure 3. La componente Angular Home.

funzioni delegate al controllo.

La sincronizzazione automatica dei valori tra la vista e il modello avviene attraverso funzionalità offerte dal framework stesso, in particolare da una direttiva inserita nell'html che consente di fare un *binding* (o un doppio *binding* nel nostro caso) tra i valori visualizzati e quelli effettivamente memorizzati nel modello: *ng-Model*. Nel momento in cui si modifica la vista, viene immediatamente avvertito e modificato il modello e viceversa.

Essendoci varie componenti è necessario definire il passaggio tra una e l'altra e, inoltre, capire come far comunicare le varie componenti. In Angular questo è gestito mediante un meccanismo di *routing*: nel costruttore di ciascuna componente viene inizializzato un router, il quale è in grado di reindirizzare il controllo ad un'altra pagina una volta chiamato (mediante un meccanismo di indirizzi preimpostati a livello più alto). Ma non solo: il router è in grado di passare anche variabili come argomenti, in modo tale che la componente figlia possa capire il proprio contesto di azione. Ad esempio, come vedremo, se la pagina figlia deve modificare un buff, dovrò passarglielo dal padre come argomento di routing per consentire la visualizzazione e la modifica.

### 3.4. Interfaccia e Funzionalità

La prima volta che viene aperta l'applicazione, viene automaticamente visualizzata la pagina di creazione del personaggio (Figura 4), suddivisa, tramite un elemento grafico chiamato *stepper*, in 3 diverse schermate secondo l'ordine logico di compilazione. È possibile in questa fase inserire le generalità del personaggio con tutte le statistiche che abbiamo ritenuto necessarie al funzionamento base del personaggio in battaglia.

Ovviamente la creazione del personaggio non è fissata una volta per tutte, ma è possibile modificarlo totalmente tramite un'apposita schermata successiva. Come da requisiti scoperti in fase di *Needfinding*, è possibile creare più personaggi.

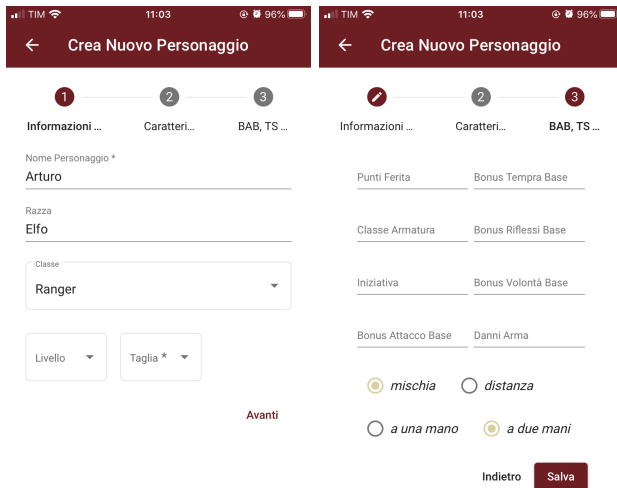


Figure 4. Schermate di aggiunta del personaggio.

Una volta creato il proprio personaggio, si è in grado di visualizzare la schermata principale. L'idea di base è stata quella di suddividerla in 3 tab, ciascuno di essi corrisponde a una specifica fase di gioco: turno di *attacco*, di *difesa* e *tiri salvezza*.

Il giocatore seleziona il tab corrispondente desiderato a seconda del momento all'interno del combattimento. Dall'alto in basso ogni schermata è suddivisa, partendo dall'alto, in:

- Una *toolbar* con il nome del personaggio, a sinistra il bottone del menu principale mentre a destra quello che rimanda alla lista dei buff completa.
- I tre tab principali sopra descritti.
- La *card* nel quale è contenuto il valore da utilizzare in battaglia, che siano i tiri per colpire, la classe armatura o i tiri salvezza. Vengono inoltre visualizzati dei triangolini per avere un colpo d'occhio riguardo ai buff e ai debuff attivi su quel determinato valore (cioè quanti influiscono nel calcolo dell'app su quel numero).
- Una lista orizzontale di *chips*, che visualizzano in sostanza le descrizioni (qualora ci fossero) di ciascun buff attivo.
- La lista verticale dei buff: un utente può selezionarli e vedere in tempo reale il cambio delle statistiche attuali. Ciascuna riga ha inoltre un bottone che richiama la pagina di modifica e visualizzazione degli effetti del singolo buff.

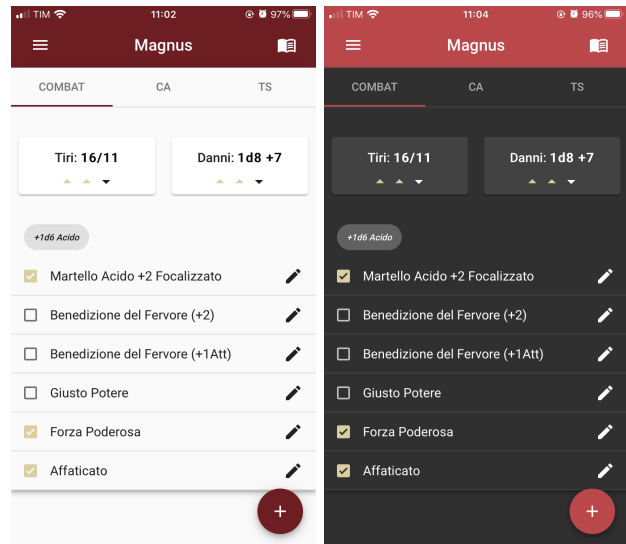


Figure 5. Schermata principale dell'applicazione, visualizzata sia nel tema chiaro standard che in *dark mode*.

- Un *fab button* che consente di eseguire "l'azione principale", secondo le regole di Material Design, cioè, in questo caso, aggiungere un *buff*; richiama quindi quella precisa pagina.

In una situazione tipica di gioco, la quantità di buff disponibili può essere davvero elevata. Per cercare di ridurre la lista all'essenziale è stato ideato un semplice stratagemma. Ciascun buff, al momento della creazione, viene catalogato a seconda che influenzi le statistiche in COMBAT, CA oppure TS (un buff può in generale appartenere a più liste). In ogni lista vengono visualizzati soltanto i buff che servono effettivamente a quella lista, ad eccezione di quelli con una descrizione (che in generale può riferirsi a più categorie). Ovviamente se si seleziona un buff che risiede anche in un'altra lista, gli effetti vengono propagati automaticamente in tempo reale.

Per aggiungere un nuovo buff è necessario cliccare sul bottone *fab* e compilare il form, includendo il nome e indicando i bonus e malus. Cliccando su "Salva" il buff verrà automaticamente smistato verso le liste giuste e sarà pronto ad essere selezionato ed usato.

Il menu principale consente di avere un colpo d'occhio su tutti i personaggi presenti all'interno del database. È sufficiente toccarne uno per selezionarlo (sarà in pratica ricaricato il modello contenuto nel file typescript della home page); il character selezionato sarà evidenziato attraverso un leggero cambio di tonalità dello sfondo. È possibile inoltre richiamare la pagina di modifica e visualizzazione del personaggio, oltre che crearne uno nuovo tramite il bottone in basso,

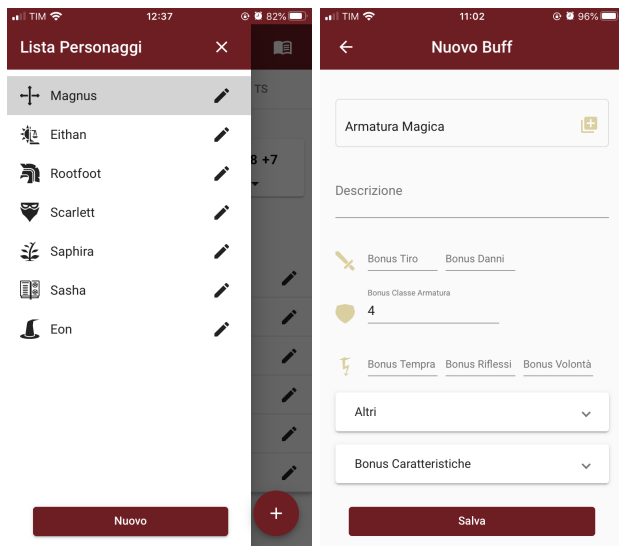


Figure 6. Il menu principale e la schermata di aggiunta di un buff.

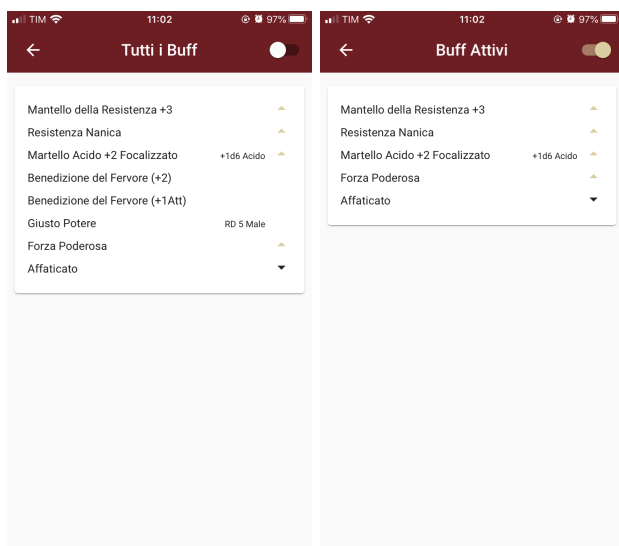


Figure 7. La lista di tutti i buff e quella dei soli buff attivi.

all'interno della quale è possibile inoltre l'eliminazione dello stesso. A livello grafico ciascuna classe base possibile è associata ad un'icona dedicata disegnata dal nostro grafico.

## 4. Usability Testing

### 4.1. Obiettivi del test

Dopo aver completato lo sviluppo dell'applicazione è necessaria una fase di testing in cui vengono messe alla prova tutte le funzionalità dell'applicazione, detta Usability Testing. Tale tipo di testing serve non tanto per

individuare errori di programmazione e bug, quanto per riuscire a capire se in fase di progettazione siano stati rispettati i requisiti che erano stati posti nelle precedenti fasi di sviluppo.

In particolare è stato necessario verificare:

- L'usabilità e la gradevolezza di utilizzo dell'app in generale.
- Quanto è noiosa e *time consuming* la fase di creazione del personaggio.
- L'accuratezza sul calcolo dei buff e se l'utente si fida dei risultati che essa restituisce.
- Se l'app è intuitiva e se sono riconosciute il *modello mentale* e la filosofia *tutto è un buff*.

Ci saremmo aspettati dei buoni punteggi per quanto riguarda la gestione dei buff in sé e dei problemi per la creazione del personaggio, la quale ci sembrava noiosa. Ma dai test sono emersi degli interessanti risultati e anche delle possibili modifiche sicuramente di valore.

### 4.2. Test scriptato

Lo Usability Testing è stato svolto con le persone intervistate durante la fase di Needfinding, articolandolo come una mini-sessione di gioco (in particolare, come un combattimento) con alcune restrizioni introdotte per indurre l'utente ad usare tutte le funzionalità dell'applicazione. È stato quindi ideato un *test scriptato* che si è svolto seguendo il seguente protocollo:

- A causa dell'elevato numero di tester, è stato necessario chiedere loro di installare l'app sul proprio smartphone (o tablet): abbiamo quindi fornito il file *.apk* agli utenti e questi hanno eseguito l'installazione senza intoppi;
- Abbiamo concesso a tutti i tester di esplorare liberamente l'applicazione per un paio di minuti, in modo tale da svolgere i vari test in un ambiente familiare;
- Per il test sono stati resi disponibili tre personaggi pre-impostati: ciascun tester ha avuto il compito di scegliere uno di questi personaggi, senza curarsi di eventuali personaggi "doppioni", in quanto non rilevanti ai fini del test;
- È stato inoltre assegnato al party un Bardo come Personaggio Non Giocante (NPC), controllato dagli sviluppatori;
- **Test 1:** Ogni tester ha avuto il compito di aggiungere il proprio personaggio all'applicazione;

#	Affermazione	$\mu$	$\sigma$
1	L'applicazione è visivamente gradevole.	5.25	0.66
2	La creazione del personaggio è stata frustrante.	1.12	0.33
3	I contenuti sono stati inseriti in un posto logicamente corretto all'interno della pagina.	5.87	0.33
4	Le animazioni sono fluide.	6.5	1.0
5	Le icone rispecchiano il comportamento atteso dell'applicazione.	6.37	0.99
6	La navigazione tra le diverse pagine di statistiche è chiara.	6.75	0.43
7	La creazione di un buff è noiosa e/o incompleta.	2.75	1.39
8	Selezionare un buff è intuitivo e utile.	6.62	0.48
9	Il calcolo del buff è corretto.	6.0	1.73
10	Trovo inutile il riepilogo di tutti i buffs nella schermata in alto a destra.	1.37	0.49
11	La gestione dei buffs è esaustiva.	5.75	0.43
12	I buff vengono suddivisi efficacemente in bonus e malus.	6.0	1.11
13	È difficile capire quali siano i buffs attivi.	1.37	0.69
14	La modifica del personaggio è rapida ed intuitiva.	6.37	1.11

Table 1. Le domande SEQ del questionario e i relativi risultati.

- **Test 2:** Ogni personaggio possiede un pezzo di equipaggiamento, che deve essere aggiunto e selezionato dagli utenti;
- **Test 3:** Vengono introdotti brevemente il contesto e l'ambientazione di gioco: durante la notte, un Coboldo ha derubato i giocatori delle preziosissime rune incastonate nelle loro armi. Dopo aver seguito con successo le tracce, i giocatori incontrano il mostro e ingaggiano il combattimento. Inizia il Bardo NPC, che infonde "Eroismo" a tutti i personaggi del party. Ai tester viene dunque chiesto di inserire il buff nell'applicazione;
- **Test 4:** Il combattimento si svolge su un terreno accidentato, che fornisce un "malus" di 2 punti alle prove sui riflessi. Ai tester è dunque richiesto di aggiungere e selezionare questa nuova voce nella lista;
- **Test 5:** Il combattimento procede con le routine d'attacco di ciascun giocatore e il mostro viene sconfitto quando tutti hanno effettuato il proprio turno. I giocatori si riappropriano delle rune per le loro armi e le incastonano nuovamente. Esse forniscono un bonus di +3 ai danni e ai tiri per colpire;
- **Test 6:** Inoltre, il combattimento ha fatto salire di livello il personaggio, che deve dunque essere aggiornato nell'applicazione da ciascun tester.
- **Test 7:** Ai tester è stato infine chiesto di cancellare il personaggio usato per il test.

Alla fine della sessione (della durata di circa 10 minuti) è stato chiesto agli utenti di rispondere ad un breve questionario composto da 14 affermazioni

cui è possibile assegnare un valore da 1 a 7 con una scala numerata, secondo lo standard SEQ, ai cui estremi si trovano valori completamente opposti (1 corrisponde a "fortemente in disaccordo" e 7 a "fortemente d'accordo") e, a seguire, due domande aperte opzionali che ci aiutassero a comprendere se ci fossero aspetti che richiedessero una miglioria.

## 5. Risultati e conclusioni

Al momento del test i partecipanti hanno trovato subito intuitivo l'utilizzo dell'applicazione e non hanno commesso errori. Non è stata subito così chiara invece l'eliminazione del personaggio. L'obiettivo della semplicità e quindi una learning curve piuttosto corta è stato raggiunto. I test (soprattutto la fase di creazione del personaggio) si sono svolti in tempi piuttosto brevi, comunque minori di quelli che potevamo immaginare.

Nella Tabella 1 sono visibili i risultati dei questionari cui i candidati sono stati sottoposti. Come è possibile notare, su alcuni aspetti quali ad esempio le animazioni, la posizione delle pagine o la scelta delle icone i tester si sono espressi abbastanza favorevolmente, ed in generale le risposte ad ogni domanda soddisfano le nostre aspettative. Stando ai dati, è emerso che:

- Sorprendentemente per noi nessuno ha trovato frustrante la creazione del personaggio.
- La creazione del buff è stata invece un punto debole, un aspetto che sicuramente si può migliorare molto.
- Il calcolo del buff non è stato sempre corretto, perché a volte ci possono essere dei casi limite difficili da replicare nell'applicazione che non sono stati ancora contemplati in fase di sviluppo (per esempio le regole riguardo al *tipo*).



- È stato facile capire quali buff siano attivi e quali no. Conseguentemente è stata trovata utile dai partecipanti la schermata di riepilogo di tutti i buff.
- La modifica del personaggio è abbastanza intuitiva ma è un aspetto da migliorare.

Ricordiamo inoltre che ai tester sono state poste due domande aperte opzionali:

- *Ti ritieni soddisfatto, in generale, della tua esperienza con l'applicazione?*
- *Ti sentiresti di consigliare l'applicazione ai tuoi amici giocatori?*

Dalle risposte dei nostri utenti abbiamo compreso che l'applicazione è piuttosto soddisfacente da usare e che il target di amici a cui dovrebbe essere consigliata maggiormente è quello del giocatore neofita, ma può essere sicuramente utilizzata e utile anche a giocatori esperti e che hanno a che fare con un numero elevato di buff. Tali risposte rispecchiano e confermano le nostre Personas e, pertanto, possiamo concludere che lo scopo dell'applicazione è stato raggiunto.

## 6. Sviluppi Futuri

L'applicazione per come si presenta adesso è ancora un prototipo, ma, anche grazie ai suggerimenti dei tester, sono emerse molti suggerimenti per sviluppi futuri:

- **Gestione dei tipi** (argomento tecnico più avanzato): nel momento in cui vengono selezionati due buff dello stesso tipo che offrono un bonus alla stessa statistica, secondo le regole standard dovrebbe essere applicato il bonus migliore e non sommato. Questa potrebbe essere una cosa da implementare in futuro.
- **Switch mischia/distanza nel tab principale** Durante un combattimento ci si può ritrovare a dover cambiare arma e/o a switchare tra mischia/distanza. È stato suggerito di inserire il pulsante nella schermata principale anziché soltanto nella schermata di modifica personaggio.
- **Buff di gruppo.** Un caso d'uso interessante che non era stato mai contemplato è quello di introdurre dei buff di gruppo (che sono in realtà una costante durante il gioco). L'idea è che uno stesso party sta utilizzando l'app e un giocatore potenzia tutti con un buff di gruppo. Attraverso una connessione (es. bluetooth) viene passato il buff

ai giocatori e viene immediatamente attivato. Se il buffer disattiva il buff, viene automaticamente disattivato anche al resto dei giocatori.

- **Database di buff:** data la lentezza dell'aggiunta di ogni singolo buff, come è emerso durante il sondaggio finale, può essere utile avere un database precostituito di buffs nel quale cercare ed eventualmente aggiungere un buff anziché crearlo tutte le volte a mano.
- **Capacità di classe automatiche:** una volta selezionata la classe, potrebbe essere utile aggiungere automaticamente delle capacità peculiari come buff, automaticamente, che quella determinata classe sicuramente possiede (es. Esibizione bardica per un bardo).
- **Card principali interattive** che consentano di effettuare alcune operazioni, ad esempio in "Classe Armatura" toccando la card (per adesso indica soltanto il punteggio ma è statica) passare a CA di contatto e impreparato. Per la routine degli attacchi potrebbe essere "attacco completo" o "colpo singolo".

## References

- [1] *Pagina ufficiale di Dungeons & Dragons*  
<https://dnd.wizards.com>
- [2] *Pagina ufficiale di Pathfinder*  
<https://paizo.com/pathfinder>
- [3] *Material Design* <https://material.io>
- [4] *Homepage del servizio Google Forms*,  
<https://www.google.it/intl/it/forms/about/>
- [5] *Homepage di Facebook*, <https://www.facebook.com/>
- [6] *Ionic* <https://ionicframework.com>
- [7] *Angular* <https://angular.io>
- [8] *Angular Material* <https://material.angular.io>
- [9] *Cordova* <https://cordova.apache.org/>
- [10] *Ionic Storage*  
<https://ionicframework.com/docs/building/storage>
- [11] *Repostory Github del progetto*,  
<https://github.com/FrancescoGradi/CombatManager>