**SPECIFICHE Esame per superamento del Corso Programmazione Avanzata**

**A.A. 2023/2024**

Gruppo: SI

---------------------------------------------------------------------------

Si chiede di realizzare un back-end utilizzando i seguenti framework / librerie:

* Node.JS
* express
* Sequelize

# INDICE:

1. [Dizionario dei termini](#_bookmark0)
2. [Descrizione in linguaggio naturale delle funzionalità](#_bookmark1)
   1. [Preparazione del combattimento](#_bookmark2)
   2. [Iniziativa](#_bookmark3)
   3. [Schermata del giocatore](#_bookmark4)
   4. [Tiri contro morte](#_bookmark5)
   5. [Schermata del master](#_bookmark6)
3. [Proposta dell’architettura di sistema](#_bookmark7)
4. [Elenco requisiti del backend](#_bookmark8)
   1. [Node](#_bookmark9)
   2. [WebSocket](#_bookmark10)
   3. [Redis](#_bookmark11)

# DIZIONARIO DEI TERMINI

|  |  |
| --- | --- |
| **VOCABOLO** | **DEFINIZIONE** |
| **Aiutanti** | Personaggi non giocabili, controllati dal master durante il combattimento.  Essi aiutano i giocatori a sconfiggere i nemici. |
| **CA** | La Classe Armatura (CA) è il valore numerico sotto al quale un tiro per colpire non ha effetto. La tua CA viene attaccata ogni volta che si  brandisce un’arma, o si lancia un incantesimo che richiede un attacco. Ogni punto di CA significa che un attaccante deve tirare 1 in più per  infliggerti danni. |
| **CD** | La Classe Difficoltà (CD) è un numero che indica quanto sia difficile un compito o una prova di caratteristica. Per effettuare una prova di  caratteristica, si tira un d20 e si somma il modificatore di caratteristica appropriato. Se il totale è uguale o superiore alla CD, la prova di  caratteristica è superata e la creatura ha superato la sfida. |
| **D20** | Dado a 20 facce, i cui valori sono determinanti per effettuare molteplici tipi di azioni nel gioco: i valori causano effetti il cui successo è  proporzionale al numero uscito. I valori estremi, 1 e 20, determinano  rispettivamente il fallimento o il successo critico. |
| **Disimpegno** | Questa azione mette in risalto l’abilità del personaggio di stordire gli  avversari, manovrarli attraverso e ritirarsi senza che i nemici sferrino un contrattacco. Se effettui l’azione Disimpegno, per il resto del turno il tuo movimento non provocherà attacchi di opportunità. È essenziale notare  che il disimpegno previene solo gli attacchi dovuti al movimento del tuo |

|  |  |
| --- | --- |
|  | personaggio. Un avversario può comunque attaccare il tuo personaggio se  è il suo turno o se ha altri mezzi al di fuori di un attacco di opportunità. |
| **Effetto** | Alcune condizioni alterano le capacità di una creatura in molti modi e possono essere provocate da un incantesimo, un privilegio di classe, un attacco di un mostro o da altri effetti. Può durare finché non viene  contrastata o non termini la sua durata. Per esempio, un giocatore invisibile non può essere visto da alcuni mostri, mentre un giocatore  assordato avrà uno svantaggio in alcuni tiri. |
| **Master** | È il principale narratore e l’arbitro del gioco. Chiamato anche DM  (Dungeon Master), egli controlla l’avventura e gli NPC mentre i personaggi  affrontano i pericoli e decidono quali sentieri esplorare. |
| **Modificatore** | Il Modificatore è un valore derivato dal punteggio di una caratteristica, dalla competenza di abilità e di alcuni strumenti. I modificatori si sommano tra di loro. Possono essere anche negativi se un personaggio  compie un’attività per cui non è portato. |
| **Mostri** | I mostri sono avversari che i personaggi dei giocatori possono affrontare.  Questi possono variare da creature semplici come goblin o orchi, a creature potenti e terrificanti come draghi o demoni.  Ogni mostro ha un blocco di statistiche che fornisce le informazioni essenziali per gestirlo in combattimento. Queste statistiche includono la Classe Armatura (CA), la vita, la velocità, le abilità, i tratti, le azioni e molto altro. Queste informazioni aiutano il DM a determinare come il mostro si comporta in combattimento, quali azioni può intraprendere, e come  reagisce alle azioni dei personaggi dei giocatori. |
| **Nemici** | Personaggi non giocabili, controllati dal master durante il combattimento.  Essi vanno contro i giocatori. |
| **NPC** | Personaggi non giocabili, controllati dal master. Nell’app vengono salvati con il nome, il legame che hanno con il party e alcune info per permettere  ai giocatori di ricordarsi chi tale personaggio sia. |
| **Punteggi caratteristica** | Sono la forza, la destrezza, la costituzione, l’intelligenza, la saggezza e il carisma che definiscono un personaggio. Tramite una formula è possibile ottenere da questi punteggi, che vanno da 3 a 20, un valore che andremo  ad aggiungere al tiro del dado. Questo valore è il modificatore. |
| **Reazione** | Una reazione è una risposta istantanea a un innesco di qualche tipo, che può verificarsi nel turno del personaggio o nel turno di qualcun altro. La reazione comprende o un attacco di opportunità nel turno di un  avversario, oppure lanciare un incantesimo con un tempo di lancio pari ad una reazione. È possibile fare una sola reazione, che si ricarica dopo aver effettuato un proprio turno. |
| **Scatto** | Quando effettui l’azione Scatto, aumenti il tuo movimento per il turno in corso. L’incremento è uguale alla tua velocità, dopo l’applicazione di qualsiasi modificatore. Ad esempio, con una velocità di 9 metri, se scatti  durante il tuo turno potrai muoverti di 18 metri. |
| **Scheda personaggio** | Raccoglie tutte le informazioni di un personaggio controllato da un  giocatore. È la base di qualsiasi gioco di ruolo. |
| **Tiri contro morte** | Situazione che avviene quando un giocatore ha vita pari a 0 o negativa. Il giocatore non è morto, ma nel suo turno può solo tirare un D20. Se  ottiene 3 successi si riprende e torna a combattere, mentre se ottiene 3 insuccessi, muore definitivamente. |

|  |  |
| --- | --- |
| **TS** | Tiro Salvezza. Il master, o il manuale, imposta un CD. I giocatori selezionati devono tirare un D20, aggiungerci il modificatore del punteggio  caratteristica con l’obiettivo di pareggiare o superare il CD. |
| **Vantaggio/Svantaggio** | Nel compiere alcune azioni, un giocatore potrebbe trovarsi in una situazione di vantaggio, come colpire un nemico alle spalle oppure  compiere un’azione in cui si è molto capaci. Altre volte potrebbe trovarsi in situazioni di svantaggio, schivare un oggetto quando non lo si vede  bene oppure muoversi su un terreno pericoloso.  Il Master decide se c’è vantaggio o svantaggio. In entrambi i casi si tirano due D20. Se abbiamo vantaggio si prende in considerazione il dado con il  risultato più alto, mentre si prende il peggiore in caso di svantaggio. |

* + 1. **DESCRIZIONE IN LINGUAGGIO NATURALE DELLE FUNZIONALITÀ**

## Preparazione del combattimento

All’interno della schermata della campagna di gioco, il master può cliccare un bottone per avviare una nuova sessione di combattimento.

Nella schermata che gli si apre dovrà indicare:

* + - * + i **giocatori partecipanti** (selezionandoli dalla lista dei giocatori che già partecipano alla campagna)
        + la **mappa** su cui avverrà il combattimento (che il master caricherà dalla galleria specificando la dimensione della griglia in un’apposita schermata interattiva, oppure selezionerà da una mappa usata in precedenza)
        + **aggiungere aiutanti e/o mostri**
        + **posizionare i combattenti** all’interno della mappa. Per aggiungere aiutant i e i mostri il master dovrà indicare:
        + Selezionare **se “mostro” o “aiutante”**
        + il **nome** (Se nuovo dovrà inserirlo. Se invece è stato creato già nella sezione “NPC” della campagna, è possibile collegarlo per poter permettere ai giocatori di visualizzare le sue info).

Se l’NPC presenta il legame “nemico” o “boss” o “nemesi”, sarà selezionato automaticamente “mostro”, ma sarà comunque possibile cambiarlo.

* + - * + le **caratteristiche**: forza, destrezza, costituzione, intelligenza, saggezza, carisma.
        + la **vita**, la **CA** (classe armatura) e la **velocità**.
        + le **armi** che usa (con il bonus d’attacco oppure con il TS (tiro salvezza) richiesto, e il danno) e/o gli

**incantesimi**. Sia le armi base che gli incantesimi sono selezionabili dal database. Il master può aggiungere all’occorrenza armi personalizzate.

Una volta che il master ha terminato i passaggi precedenti, potrà selezionare se “avviare il combattimento” subito oppure “salvare ed uscire” per farlo partire in seguito. Quest’ultima opzione risulta comoda nel caso si volesse interrompere la sessione prima di avviare un combattimento già preparato.

## Iniziativa

Una volta partito il combattimento, ai giocatori apparirà nella loro sezione “combattimento” una nuova

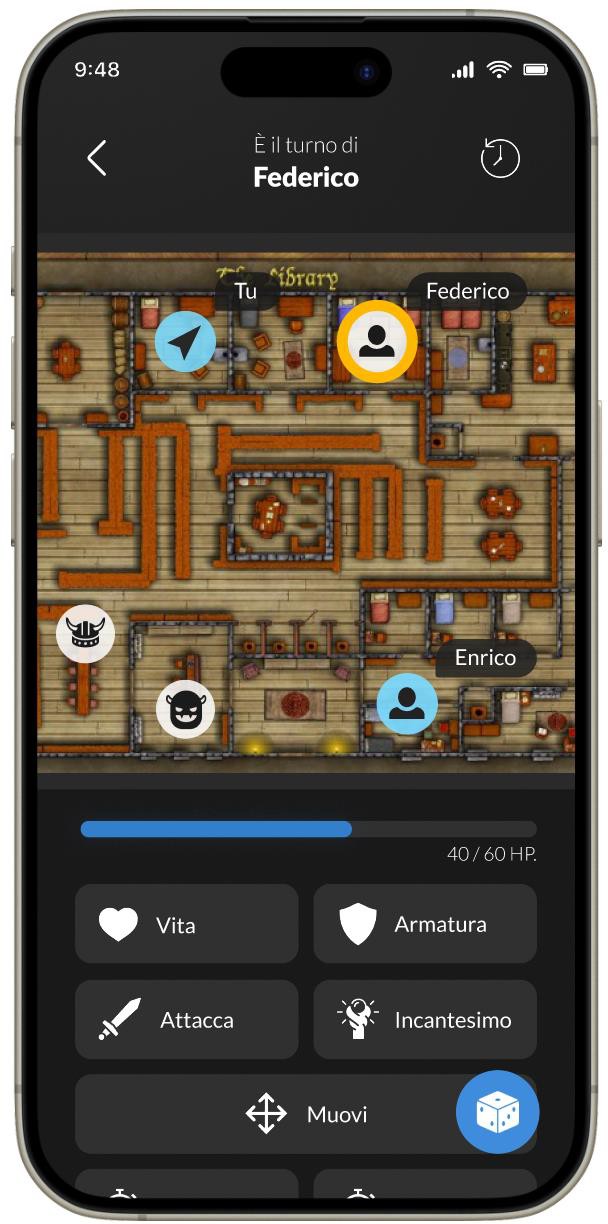
card. Cliccandoci verranno portati all’interno della mappa dove sarà richiesto loro un tiro d’iniziativa (D20 +modificatore di destrezza del giocatore). Una volta lanciato e salvato il tiro, **i giocatori rimarranno in attesa finché tutte le iniziative non saranno tirate. In caso di parità** tra due o più personaggi **verrà richiesto loro di rieffettuare il tiro** di iniziativa per decidere chi parte prima.

Se il master durante il combattimento inserisce un nuovo mostro e/o aiutante, quest’ultimo dovrà

effettuare il tiro d’iniziativa. **Se pareggia** il primo risultato di uno o più combattenti, **verrà messo sotto di essi**, al fine di non costringere i giocatori ad un altro tiro d’iniziativa.

## Schermata del giocatore

Per quanto concerne la visualizzazione della schermata verrà mostrato (partendo dall’alto):

* + - * + di chi è il turno
        + il bottone “history”
        + la mappa
        + la barra della vita
        + il bottone “vita”
        + il bottone “armatura”
        + il bottone “attacca”
        + il bottone “incantesimo”
        + il bottone “muovi”
        + il bottone “posticipa”
        + il bottone “termina”
        + FAB “dado”

**Turnistica:** verrà riportato di chi è il turno in base all’ordine di avvio della battaglia. Quando non è il proprio turno i seguenti bottoni saranno **disabilitati**: “*posticipa*”, “*termina* turno”, “*incantesimo*” e “*attacco*”.

**History:** serve a tenere traccia sia dei tiri effettuati dal FAB “dado”, sia dai lanci e azioni generate dal bottone “attacca” o “incantesimo” che saranno evidenziati diversamente. Al tempo stesso verranno mostrati dei messaggi generati dal cambio di effetto di alcuni giocatori/NPC.

**Mappa:** nella mappa un giocatore potrà interagire:

* + - * + con la sua icona per poter visualizzare la sua scheda personaggio,
        + con le icone degli altri giocatori per vedere alcune loro caratteristiche (nome, classe, razza, livello, vita, CA, punteggi caratteristica, effetto, se il giocatore è online)
        + con le icone dei mostri e aiutanti: se è collegato ad un NPC precedentemente creata, verrà mostrato il nome, il legame e le info, altrimenti soltanto il nome di quel NPC.

Sempre nella mappa le varie icone avranno un bordo con dei colori diversi a seconda che siano aiutanti, giocatori o nemici.

**Barra della vita:** rappresenta la vita di un giocatore e scalerà proporzionalmente. Essa cambierà colore, in rosso, arrivata ad 1/10 di quella iniziale. Quando la vita è pari o inferiore allo 0, si avrà la condizione di “Tiri contro morte” (vedere sotto).

**Bottone “vita:”** permette di modificare la vita del proprio personaggio sia in positivo che in negativo. Senza alcun limite, ad esempio, sarà possibile avere 65/60 come vita in alcuni casi particolari definiti dalle

dinamiche di gioco.

**Bottone “armatura”:** permette di modificare la CA di un personaggio. È già impostata la CA presa della scheda personaggio.

Ogni modifica apportata da questi due bottoni rimane all’interno della battaglia e non andrà ad influenzare il salvataggio su Firebase.

**Bottone “attacca”:** il giocatore deve fare due scelte ovvero chi attaccare (obbligatorio) e con che arma

attaccare (potrebbe non possedere alcuna arma e decidere di attaccare a mani nude. In questo caso i danni che infliggerà saranno stabiliti dal Master).

Dopo aver effettuato l’attacco, comparirà tale messaggio nella History: *Valerio ha attaccato Goblin con spadone*.

Effettuate le scelte, il giocatore vedrà comparire una schermata di lancio con un D20, le opzioni facoltative di vantaggio, svantaggio e il modificatore da inserire. Sarà, inoltre, possibile al giocatore accedere alla sua scheda personaggio per controllare i vari bonus da considerare.

Se il tiro effettuato supera la CA del nemico apparirà il messaggio: *L’attacco è andato a buon fine*. Altrimenti apparirà: *L’attacco è fallito.*

**Bottone “incantesimo”:** il giocatore vedrà la lista dei suoi incantesimi selezionabili. Ogni incantesimo ha una caratteristica nascosta (attacco, tiro salvezza, descrittivo) che genererà tre possibili andamenti:

* + - * + **Attacco**: il giocatore dovrà scegliere uno o più nemici su cui sferrare l’attacco e il sistema si comporterà nello stesso modo del bottone attacca.
        + **Tiro salvezza**: apparirà il messaggio: *Valerio ha usato (Mani brucianti).*

Apparirà al master un pop-up in cui dovrà selezionare quali giocatori/mostri/aiutanti dovranno fare il tiro salvezza, su cosa dovranno farlo, e con quale CD. Ai giocatori interessati apparirà una

schermata di lancio con un D20, le opzioni facoltative di vantaggio e svantaggio ed il modificatore da inserire.

Se lo superano apparirà: *Enrico ha superato il tiro salvezza*. Altrimenti: *Enrico non ha superato il tiro salvezza*.

* + - * + **Descrittivo**: questi sono incantesimi che di per sé non fanno danno, ma potrebbero farlo. Siccome sono molto specifici e difficili da gestire, apparirà solo il messaggio che il giocatore ha usato

l’incantesimo: *Valerio ha usato (Mano magica).*

**Bottone “muovi”:** una volta cliccato sarà permesso al giocatore di muoversi di un numero di caselle

determinato dalla sua velocità di movimento. Non sarà possibile selezionare una casella contenente già un altro personaggio. Siccome non esiste solo l’azione di movimento classica, ma anche il disimpegno e lo scatto, al giocatore sarà possibile cliccare tale bottone più volte in un turno. Spetterà al master fermarlo se commette qualche infrazione.

**Bottone “posticipa”:** il giocatore posticiperà il turno indicando dopo quale giocatore vuole cominciare il suo turno. Potrà indicare solo i giocatori/mostri/aiutanti che si trovano dopo di lui nell’ordine di iniziativa.

**Bottone “termina turno”:** il giocatore termina il proprio turno passando al successivo. Le sue limitazioni sono elencate nella “Turnistica” (vedi sopra).

**FAB “dado”:** permette al giocatore di tirare dadi senza limitazioni. Ciò permette di evitare il problema al sistema di calcolare i dadi che un giocatore lancerà per i danni, ma permetterà anche di gestire situazioni impreviste nella campagna. Il messaggio che comparirà nella history sarà di **colore diverso** rispetto ai

precedenti e del tipo: *Valerio ha lanciato D20 = 16*.

## Tiri contro morte

Quando la vita di un giocatore, e non di un aiutante/mostro, è uguale o inferiore allo 0, verrà mostrato solo:

* + - * + di chi è il turno (vedi sopra)
        + il bottone “history” (vedi sopra)
        + la mappa (vedi sopra)
        + la barra della vita (vedi sopra)
        + il bottone “vita” (vedi sopra)
        + Sei pallini.

**6 pallini:** saranno divisi in 2 gruppi da 3 e saranno vuoti. 3 di questi rappresentano i successi, 3 gli insuccessi. Quando il giocatore è a terra, potrà solo tirare un semplice D20.

Risultati del D20:

* 1 = 2 pallini degli insuccessi si riempiono
* 2-10 = 1 pallino degli insuccessi si riempie
* 11-19 = 1 pallino dei successi si riempie
* 20 = tutti e 3 i pallini dei successi si riempiono

Dopo aver effettuato il tiro, il turno passa al giocatore successivo.

Se si riempiono i 3 pallini dei successi: il giocatore torna ad 1 di vita e la schermata torna ad essere quella normale.

Se si riempiono i 3 pallini degli insuccessi: il giocatore dovrà essere eliminato dalla turnistica e la sua icona sparisce dalla mappa. Seppur il giocatore è morto, non deve essere eliminata la scheda personaggio e può continuare ad osservare come spettatore il combattimento fino alla sua conclusione.

Ogni volta che il giocatore finisce a fare i tiri contro morte, i pallini cominciano vuoti. Non vengono salvati i tiri precedentemente fatti.

## Schermata del Master

Per quanto concerne la visualizzazione della schermata del master verrà mostrato (partendo dall’alto):

* + - * + di chi è il turno (vedi sopra)
        + il bottone “history” (vedi sopra)
        + la mappa
        + la barra della vita (vedi sopra)
        + il bottone “vita”
        + il bottone “armatura” (vedi sopra)
        + il bottone “attacca” (vedi sopra)
        + il bottone “incantesimo” (vedi sopra)
        + il bottone “muovi” (vedi sopra)
        + il bottone “termina turno” (vedi sopra)
        + il bottone “**tiro salvezza**”
        + il bottone “**modifica effetto**”
        + il bottone “**aggiungi NPC**”
        + il bottone “**ordine**”
        + il bottone “**info**”
        + il bottone “**esci**”
        + FAB “dado” (vedi sopra)

**Mappa:** nella mappa il master potrà interagire:

* + - * + con le icone dei giocatori per vedere le loro caratteristiche (nome, classe, razza, livello, vita, vita massima, CA, punteggi caratteristica, effetto, stato online del giocatore)
        + con le icone degli aiutanti/mostri per poter assumerne il controllo.

**Barra della vita:** rappresenta la vita di un mostro/aiutante e scalerà proporzionalmente. Cambierà colore, in rosso, arrivata ad 1/10 di quella iniziale.

Quando la vita è pari o inferiore allo 0:

* + - * + Se è un **mostro** scomparirà la sua icona dalla mappa e non sarà più possibile interagirci.
        + Se è un **aiutante**, verranno effettuati i tiri contro morte (vedi sopra).

**Bottone “tiro salvezza”:** è possibile che durante la battaglia il master richieda ad alcuni di fare un tiro salvezza per una circostanza imprevista. Apparirà al master un pop-up in cui dovrà selezionare quali

giocatori/mostri/aiutanti dovranno fare il tiro salvezza, su cosa dovranno farlo, e con quale CD. Ai giocatori interessati apparirà una schermata di lancio con un D20, le opzioni facoltative di vantaggio e svantaggio, ed infine il modificatore sotto da inserire.

Se lo superano apparirà nella history: *Enrico ha superato il tiro salvezza*. Altrimenti: *Enrico non ha superato il tiro salvezza*.

**Bottone “modifica effetto”:** il master indica quale giocatore/mostro/aiutante subisce il cambiamento di effetto e seleziona uno dei seguenti effetti, che possono essere anche 2 o più contemporaneamente e dovranno sempre essere tolti dal master.

Gli effetti in game sono:

* + - * + accecato (non vede la mappa)
        + affascinato (non può attaccare la persona che lo ha affascinato)
        + afferrato (non può muoversi)
        + assordato (nessun effetto da gestire)
        + avvelenato (nessun effetto da gestire)
        + incapacitato (non può attaccare, lanciare incantesimi o effettuare reazioni)
        + invisibile (la sua icona è visibile solo a sé stesso e al master)
        + paralizzato (non può muoversi, attaccare, lanciare incantesimi o effettuare reazioni)
        + pietrificato (non può muoversi, attaccare, lanciare incantesimi o effettuare reazioni e non vede la mappa)
        + privo di sensi (non può muoversi, attaccare, lanciare incantesimi o effettuare reazioni e non vede la mappa)
        + prono (non può attaccare, lanciare incantesimi o effettuare reazioni)
        + spaventato (non può avvicinarsi alla fonte della sua paura)
        + stordito (non può muoversi, attaccare, lanciare incantesimi o effettuare reazioni)
        + trattenuto (non può muoversi)

Si nota che tali effetti avranno conseguenze visive nella mappa solo per i giocatori coinvolti dall’effetto, mentre per gli aiutanti/mostri controllati dal master non ci sarà nessun cambiamento.

**Bottone “aggiungi NPC”:** il master durante il combattimento può aggiungere un mostro/aiutante. Vedere il paragrafo *Preparazione Combattimento* per vedere cosa il master deve inserire. Una volta preparato, il master dovrà indicare sulla mappa dove posizionarlo. Vedere il paragrafo *Iniziativa* per gestire il suo

inserimento nella turnistica.

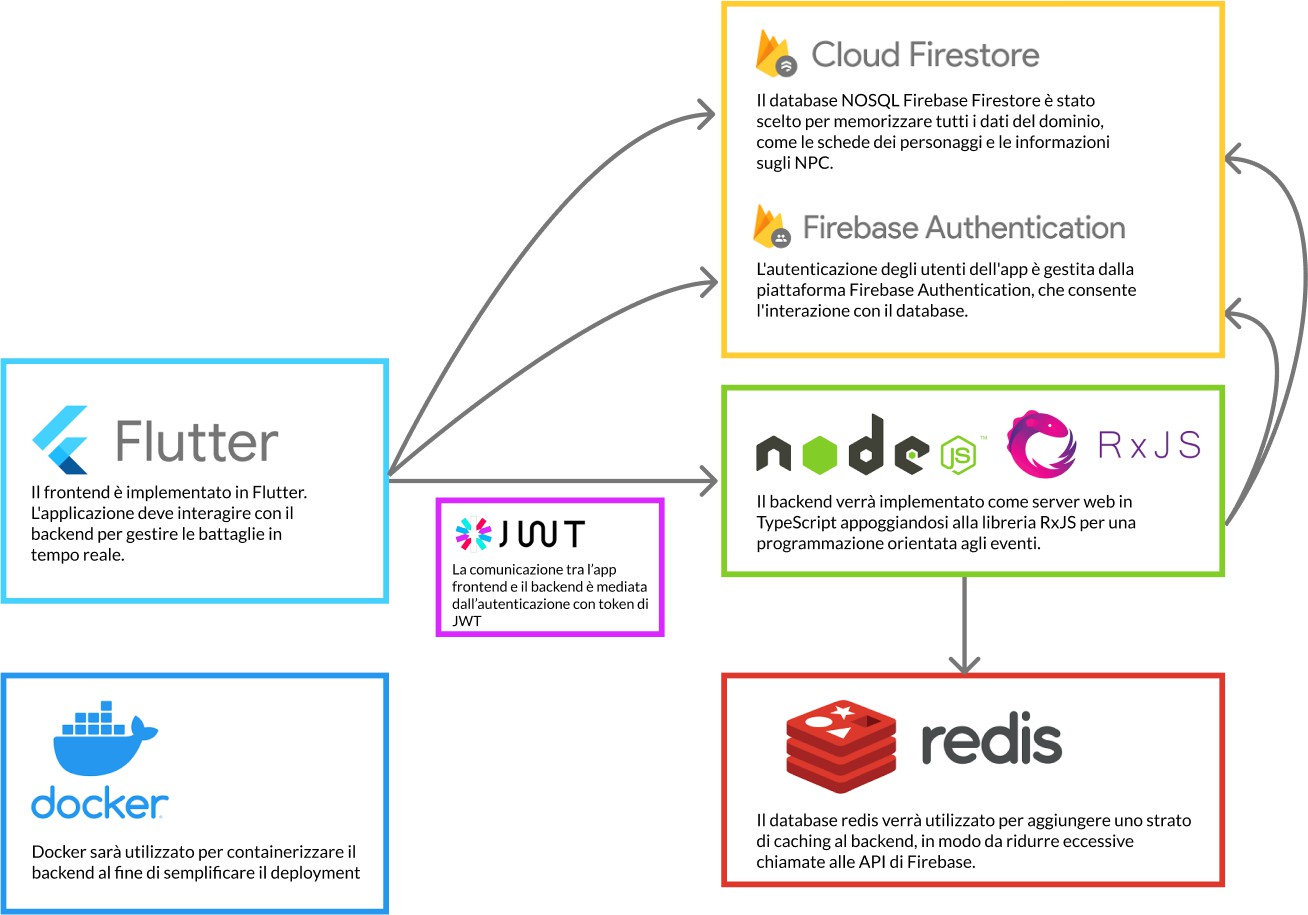
**Bottone “ordine”:** al master appare un pop-up con l’ordine dei giocatori/mostri/aiutanti che agiscono nel combattimento. Questo sembra superfluo, ma al master è necessario avere il quadro generale della

situazione.

**Bottone “info”:** al master appaiono le info del mostro/aiutante che sta controllando al momento. Di esso saranno mostrati: nome, vita, CA, punteggi caratteristica, effetto.

**Bottone “esci”**: cliccandoci, verrà chiesto al master se si vuole far uscire dal combattimento un singolo giocatore (nel caso in cui volesse fuggire), oppure se si vuole far terminare l’intero combattimento.

# PROPOSTA DELL’ ARCHITETTURA DI SISTEMA



* + 1. **ELENCO DEI REQUISITI DEL BACKEND**

## Node

Di seguito sono elencate le specifiche che il backend dovrà soddisfare al fine di permettere il funzionamento dei vari aspetti del combattimento sopra elencati.

Nota: con il termine *entità* si intenderà in egual modo un mostro, un aiutante o un giocatore.

|  |  |
| --- | --- |
| **Informazioni che il backend deve ricevere (e da chi)** | |
| **Gestione della sessione** | |
| Creare una nuova sessione di combattimento | Master |
| Avviare/Riprendere una sessione di combattimento | Master |
| Mettere in pausa una sessione di combattimento | Master |
| Concludere definitivamente una sessione di combattimento | Master |
| **Presenze e turnazione** | |
| Modificare l’ordine della turnazione | Player, Master |
| Aggiungere/Rimuovere un’entità dal combattimento | Master |
| **Attacchi e incantesimi** | |
| Eseguire un’azione di attacco/incantesimo di danno semplice  verso un’altra entità | Player, Master |

|  |  |
| --- | --- |
| Eseguire un’azione di incantesimo con tiro salvezza verso un’altra  entità | Player, Master |
| Eseguire un’azione di incantesimo descrittivo | Player, Master |
| Far effettuare un tiro salvezza ad una entità | Master |
| **Variazione dello stato delle entità** | |
| Assegnare un effetto ad un’entità | Master |
| Aggiornare lo stato di un’entità (Vita, CA, Posizione) | Player, Master |
| Modificare il numero di slot disponibili per livello di incantesimo | Player, Master |
| Abilitare la reazione per una o più entità | Master |
| **Informazioni che il backend deve fornire (e a chi)** | |
| Comunicare chi è di turno terminatone un altro | Player, Master |
| Comunicare lo stato di un’entità (Vita, CA, effetto, se è online o  no, slot incantesimi, tiri contro morte) | Player, Master |
| Aggiornare e comunicare la history | Player, Master |
| Richiedere, a uno o più entità, di effettuare un tiro salvezza (su  quale caratteristica, CD, eventuale vantaggio/svantaggio) | Player, Master |
| Comunicare la possibilità di effettuare una reazione | Player, Master |

Ogni funzione è associata ad una diversa richiesta *http*, per ciascuna delle quali è prevista un'autenticazione tramite token JWT.

Qui sotto sono riportate in dettaglio le descrizioni di ogni specifica presente nella tabella precedente.

### Creare una nuova sessione di combattimento

Nella fase di preparazione di un combattimento il master deve comunicare al backend le seguenti informazioni:

* Elenco dei personaggi (uid, per poterli ripescare dal Firestore, e posizione sulla mappa)
* Elenco degli aiutanti e mostri (nome, legame, skills, ca, hp\_max, velocità, armi, incantesimi e posizione sulla mappa)
* Dimensione griglia della mappa (per poter validare i movimenti in base alla velocità delle entità)

Il backend implementerà una politica di caching attraverso Redis per recuperare gli oggetti relativi ai personaggi dal Firestore a partire dal loro uid.

### Avviare/Riprendere una sessione di combattimento

Quando il master ritiene che tutti i preparativi siano pronti, può procedere ad avviare la sessione di combattimento, informando il backend di tale intenzione, che andrà a recuperare tutte le informazioni

salvate dopo la fase di preparazione antecedente. A quel punto, il combattimento risulterà “attivo” e tutte le entità coinvolte potranno iniziare ad interagire in base all’ordine di turnazione, a partire dallo stato iniziale

della mappa.

Il combattimento può essere anche ripreso dopo essere stato messo in pausa e il suo avvio avviene come descritto sopra, con l’unica eccezione che non partirà dallo stato iniziale ma dall’ultimo presente dopo la sospensione. Ciò implica che il backend dovrà memorizzare le informazioni relative alle sessioni di combattimento.

### Mettere in pausa una sessione di combattimento

Qualora il master volesse mettere in pausa il combattimento, notificherà tale intenzione al backend e il combattimento risulterà “sospeso”: questa condizione implica la memorizzazione dello stato della sessione di combattimento.

In questa fase, il backend impedirà ai giocatori e al master qualsiasi tipo di interazione volta a modificare lo stato attuale finché il combattimento non sarà avviato nuovamente, con la stessa procedura del primo

avvio, a partire dall’ultimo stato salvato prima della sospensione.

### Concludere definitivamente una sessione di combattimento

Il combattimento può concludersi secondo vari scenari:

* Tutti i giocatori vengono sconfitti, con conseguente sconfitta dei giocatori;
* Tutti i mostri vengono sconfitti, con conseguente vittoria dei giocatori;
* Il master comunica esplicitamente al backend che intende concludere il combattimento arbitrariamente.

In ognuno di questi casi, il backend impedirà sia ai giocatori che al master qualunque tipo di interazione volta a modificare lo stato attuale, che a quel punto sarà lo stato finale del combattimento. Il combattimento risulterà “concluso” e sarà possibile accedere soltanto alla sua history per visualizzarne il contenuto.

### Modificare l’ordine della turnazione

La turnazione è interamente gestita dal backend che provvederà ogni volta ad abilitare le azioni disponibili al giocatore che deve effettuare il proprio turno, disabilitandole a tutti gli altri.

L’ordine della turnazione delle entità viene memorizzato subito dopo la fase di iniziativa: il backend memorizzerà i risultati dei lanci dei dadi e stabilirà l’ordine della turnazione partendo dal risultato più alto, proseguendo in ordine decrescente.

Qualora un’entità posticipasse il proprio turno, indicherà al backend una delle entità che si trova dopo di lui nell’ordine di iniziativa. A quel punto, il backend continuerà a memorizzare l’ordine predefinito, ma solo per questo turno sposterà tale entità dopo quella specificata e gli proibirà qualsiasi azione (ad eccezione della reazione) finché non toccherà di nuovo ad essa. Alla fine di tale turno per tutte le entità, si riprenderà con

l’ordine iniziale di turnazione.

Qualora una entità volesse terminare il proprio turno prima di aver concluso tutte le mosse consentite, indicherà al backend tale volontà e gli sarà impedita qualsiasi azione (ad eccezione della reazione) fino al prossimo giro di turnazione.

### Aggiungere/Rimuovere un’entità dal combattimento

Il master può aggiungere una nuova entità anche quando il combattimento risulta “attivo”, in tal caso la nuova entità dovrà effettuare un tiro di iniziativa, il cui risultato sarà comunicato al backend che si occuperà di inserirlo nel sistema di turnazione.

Per quanto riguarda la rimozione di un’entità dal combattimento, si possono verificare diverse situazioni:

* Un giocatore o un aiutante sono arrivati a tre fallimenti nei tiri contro morte: in questo caso il backend dovrà rimuovere automaticamente questa entità.
* Un mostro viene ridotto a 0 Hp: anche in questo caso il backend dovrà rimuovere automaticamente questa entità, senza la necessità di passare ai tiri contro morte.
* Il master decide arbitrariamente di rimuovere un’entità dal combattimento per altre ragioni legate alle dinamiche di gioco (es. un’entità decide di fuggire).

In ognuno di questi casi, al backend sarà notificato quale entità rimuovere dal combattimento: nello

specifico, essa sarà rimossa dal sistema di turnazione e non potrà interagire più in alcun modo con lo stato del combattimento.

### Eseguire un’azione di attacco/incantesimo di danno semplice verso un’altra entità

Durante il proprio turno, un’entità può comunicare al backend l’intenzione di attacco, notificando ad esso:

* Il nome dell’arma/incantesimo selezionato
* L’entità scelta come bersaglio
* Il risultato del tiro effettuato per colpire il bersaglio

A quel punto, il backend controllerà che l’attacco sia andato a buon fine o sia fallito e, in entrambi i casi, registrerà tale avvenimento nella history.

Qualora l’attacco fosse andato a buon fine, la persona che controlla tale entità effettuerà i tiri per i danni e il backend si occuperà di scalare tale valore alla vita attuale dell’entità scelta come bersaglio.

### Eseguire un’azione di incantesimo con tiro salvezza verso un’altra entità

Durante il proprio turno, un’entità può comunicare al backend l’intenzione di lanciare un incantesimo che richiede un tiro salvezze da parte di chi lo subisce. Dovrà quindi specificare:

* Il nome dell’incantesimo selezionato
* L’entità scelta come bersaglio
* Il risultato del tiro effettuato per colpire il bersaglio

A quel punto, il backend richiederà all’entità bersaglio di effettuare il tiro salvezza e, in base all’esito stabilirà se l’attacco è andato a buon fine registrando tale avvenimento nella history.

### Eseguire un’azione di incantesimo descrittivo

Durante il proprio turno, un’entità può comunicare al backend l’intenzione di lanciare un incantesimo

descrittivo. Dovrà quindi specificare il nome di tale incantesimo. Il backend registrerà tale avvenimento nella history.

### Far effettuare un tiro salvezza ad una entità

Il Master può richiedere, per svariati motivi, ad un’entità di effettuare un tiro salvezza su una determinata caratteristica, con una certa CD e con eventuale vantaggio/svantaggio. Ricevuta la richiesta, il backend la inoltrerà all’entità coinvolta.

### Assegnare un effetto ad un’entità

A seguito del lancio di un incantesimo o per altri motivi, il master può assegnare un effetto ad un’entità. Alcuni effetti (come “cieco”) avranno delle conseguenze sulla visualizzazione lato frontend. Il backend registrerà la modifica e la memorizzerà nella history.

### Aggiornare lo stato di un’entità (Vita, CA, Posizione)

Dopo ogni azione che modifica aspetti legati allo stato di un’entità, il giocatore o il master comunicheranno al backend il nuovo stato. Il backend registrerà la modifica e la comunicherà a tutti i giocatori e master del combattimento filtrandola per i giocatori.

I giocatori, ad esempio, non possono conoscere la vita dei mostri controllati dal master, ma possono conoscere la loro posizione.

### Modificare il numero di slot disponibili per livello di incantesimo

Ogni qualvolta un giocatore o il master decidano di lanciare un incantesimo, devono comunicare al backend quale livello di slot incantesimo intendano utilizzare. Il backend verificherà la disponibilità e in caso

contrario non permetterà l’azione.

I giocatori e il master possono ripristinare manualmente il numero degli slot disponibili, poiché dipendono da svariati fattori non gestibili, comunicandolo al backend.

### Abilitare la reazione per una o più entità

Di base, se non è il proprio turno, non è permesso alle entità di attaccare o usare incantesimi.

Tuttavia, se la situazione di gioco lo consentirà, il master può notificare al backend l’intenzione di abilitare la reazione di un’entità. (Per reazione si intende la capacità di sferrare un attacco di opportunità o di lanciare un incantesimo con un particolare tempo di lancio nei limiti descritti nel manuale di gioco.)

### Comunicare chi è di turno terminatone un altro

Quando un’entità comunica la fine del proprio turno, il backend dovrà effettuare 3 azioni:

* Disabilitare le azioni “*posticipa*”, “*termina* turno”, “*lancia incantesimo*” e “*attacco*” all’entità che ha terminato il proprio turno
* Comunicare di chi è il turno andando a leggere dalla turnistica che ha memorizzato in precedenza
* Abilitare le azioni sopra elencate a chi di turno

### Comunicare lo stato di un’entità (Vita, CA, effetto, se è online o no, slot incantesimi, tiri contro morte)

Quando un giocatore o il master richiedono lo stato di un altro giocatore, il backend dovrà fornire loro una serie di informazioni utili (vita, CA, effetto, slot incantesimi...) andando a riprenderle dalle schede

personaggio salvate su Redis e aggiornate con l’evolversi del combattimento.

Nel caso in cui invece venga richiesto lo stato di un mostro/aiutante, il backend dovrà verificare che a richiederlo sia il master e non un giocatore.

### Aggiornare e comunicare la history

Le azioni di attacco, di incantesimo e di lancio di dado saranno ricordate dal backend per tenere traccia dell’evoluzione del combattimento (la history).

I messaggi memorizzati dipendono dalle diverse situazioni:

* Se l’entità ha sferrato un attacco saranno del tipo “X ha colpito il bersaglio” oppure “X ha mancato l’obiettivo)
* Se l’entità ha effettuato un tiro saranno del tipo “X ha superato il tiro salvezza” oppure “X non ha superato il tiro salvezza”
* Se l’entità ha lanciato un dado casuale saranno del tipo “X: D20+5= 17”
* Se l’entità ha lanciato un dado per fare danni saranno del tipo “X ha inferto D20+5=17 danni a Y”

Nei primi due casi la scelta del messaggio dipende dal risultato dei dadi delle entità che verranno comunicati al backend che gli andrà a confrontare con la CA di un’entità o la CD del tiro scelto.

### Richiedere, a uno o più entità, di effettuare un tiro salvezza (su quale caratteristica, CD, eventuale vantaggio/svantaggio)

Alle entità scelte dai giocatori, nel caso degli incantesimi con il tiro salvezza, o dal master che dispone di un bottone apposito per le situazioni impreviste, il backend mostrerà alle entità coinvolte:

* Un d20 da lanciare
* La possibilità di selezionare vantaggio/svantaggio
* Un modificatore impostato già su quello della caratteristica selezionata, comunque modificabile

Il backend valuterà il risultato del lancio, ovvero se è stato superato o no confrontandolo con la CD, e scriverà l’informazione nella history.

### Comunicare la possibilità di effettuare una reazione

Dopo che il master ha scelto l’entità alla quale permettere di sfruttare la propria reazione, il backend dovrà abilitare sia l’azione di “attacco” sia l’azione “lancio incantesimi” con solo quelli che hanno un tempo di

lancio pari a 1 reazione.

Il backend dovrà impedire che un’entità possa riusare la propria reazione prima che abbia effettuato un

proprio turno di gioco. Quindi esso dovrà tenere traccia degli utilizzi e impedirà al master di selezionare le entità che ne hanno già usufruito.

## WebSocket

Data la forte dinamicità che sussiste nella comunicazione tra il backend e i giocatori, quest’ultima verrà gestita tramite una WebSocket che verrà aperta all’inizio del combattimento (Avviare/Riprendere una sessione di combattimento) e chiusa al termine dello stesso (Mettere in pausa una sessione di combattimento e Concludere definitivamente una sessione di combattimento).

La connessione di tutti i giocatori alla WebSocket permetterà al backend di comunicare in tempo reale le modifiche sullo stato di gioco dovute alle azioni scelte nei turni dei giocatori.

La connessione alla WebSocket consentirà inoltre al backend di stabilire l’eventuale disconnessione di un giocatore dalla sessione.

## Redis

All’inizio di una sessione di combattimento, il backend dovrà scaricare dal database Firestore le schede dei personaggi coinvolti di cui il master comunicherà i relativi UID alla creazione della sessione.

Il backend dovrà inoltre scaricare dal Firestore i documenti relativi ai 362 incantesimi di gioco per poterli consultare ogni qualvolta un giocatore scelga di lanciarli.

In tutti questi (e sicuramente altri) casi, è di vitale importanza l’utilizzo di un sistema di caching il più prestante possibile per evitare tediose attese da parte dei giocatori durante lo svolgimento del combattimento.

Si chiede di utilizzare le funzionalità di middleware.

Si chiede di gestire eventuali errori mediante gli strati middleware sollevando le opportune eccezioni.

Si chiede di commentare opportunamente il codice.

**Note:**

Nello sviluppo del progetto è richiesto l’utilizzo di Design Pattern che dovranno essere documentati opportunamente nel Readme.MD.

Implementazione in typescript.

**Specifiche Repository**

* Il codice deve essere reso disponibile su piattaforma github con repo pubblico
* Nel repository è obbligatario inserire un Readme.md che descriva:
  + Obiettivo del progetto
  + Progettazione
    - diagrammi UML (casi d’uso, diagrammi delle sequenze)
    - descrizione dei pattern usati motivandone la scelta
  + Come avviare il progetto mediante docker-compose per comporre i servizi richiesti (fornire tutti i file neceessari).
  + Test del progetto mediante chiamate effettuate Postman / Newman (fornire collection)
* Il Readme.MD può essere redatto in lingua italiana o inglese (non vi saranno differenziazioni nel processo di valutazione)

**Specifiche Consegna**

* La consegna avviene esclusivamente mediante moodle all’indirizzo di seguito riportato dove dovranno essere indicati:
  + URL del repository pubblico
  + Commit id che verrà usata dal docente per effettuare la valutazione.
  + Data per lo svolgimento dell’esame
* Indirizzo per la consegna: <https://learn.univpm.it/mod/assign/view.php?id=531167>

Buon lavoro 😉

Il docente, Adriano Mancini