

# **UNIVERSITÀ DEGLI STUDI DI SALERNO**

Dipartimento di Informatica



*Corso di laurea in Informatica*

**TESI DI LAUREA IN INFORMATICA**

***Disturbi di lettura: un'app Android di supporto  
al training***

**Relatore:**

**Prof.ssa  
Rita Francese**

**Candidata:**

**Benedetto Sommese  
Matricola: 0512104801**

**Anno Accademico 2018/2019**

*Dedicato,  
Alle persone importanti della mia vita*

# Sommario

<b>INTRODUZIONE.....</b>	<b>5</b>
<b>CAPITOLO 1: STATO DELL'ARTE.....</b>	<b>8</b>
<b>CAPITOLO 2: DSA- DISTURBI SPECIFICI SULL'APPRENDIMENTO .....</b>	<b>12</b>
ART.1 RICONOSCIMENTO E DEFINIZIONE DI DISLESSIA, DISGRAFIA, DISORTOGRAFIA E DISCALCULIA. ....	13
ART.2 FINALITÀ .....	13
ART.3 DIAGNOSI.....	14
ART. 4 FORMAZIONE NELLA SCUOLA .....	14
ART. 5 MISURE EDUCATIVE E DIDATTICHE DI SUPPORTO .....	15
ART. 6 MISURE PER I FAMILIARI .....	15
<b>CAPITOLO 3: DISLESSIA .....</b>	<b>16</b>
DISLESSIA EVOLUTIVA .....	16
<b>CAPITOLO 4: SOFTWARE E LINGUAGGI UTILIZZATI.....</b>	<b>20</b>
UNITY .....	20
C# .....	21
TOUCH E TOUCHPHASE .....	26
ANDROID STUDIO .....	28
XAMPP .....	28
MYSQL .....	29
SQL .....	29
ECLIPSE .....	29
SERVLET E JSP .....	29
APACHE TOMCAT.....	30
<b>CAPITOLO 5: SISTEMA PROPOSTO .....</b>	<b>31</b>
PROGETTAZIONE BASE DATI.....	33
SPECIFICHE DELLA REALTÀ DI INTERESSE .....	33
PROGETTAZIONE MODELLO ER .....	34
PROGETTAZIONE SCHEMA LOGICO-RELAZIONALE .....	36
IMPLEMENTAZIONE DELLO SCRIPT .....	37
REQUISITI FUNZIONALI .....	42
REQUISITI NON FUNZIONALI .....	46
REQUISITI DI PRODOTTO .....	46
REQUISITI ORGANIZZATIVI .....	46
SCENARI .....	46
DESCRIZIONE DELLE INTERFACCE.....	48

IMPLEMENTAZIONE PRIMA VERSIONE .....	50
IMPLEMENTAZIONE SECONDA VERSIONE .....	55
<b>CAPITOLO 6: CONCLUSIONI.....</b>	<b>62</b>
<b>BIBLIOGRAFIA .....</b>	<b>63</b>
<b>RINGRAZIAMENTI .....</b>	<b>64</b>

# Introduzione

La dislessia è un disturbo specifico della lettura che si manifesta con una difficoltà nella decodifica del testo.

Per aiutare ad identificare chi soffre di questa patologia è stata realizzata un'applicazione, "DSA App", che offre la possibilità di eseguire un test sulla lettura, che non effettua alcuna diagnosi, ma bensì cerca di individuare un probabile disturbo sull'apprendimento. Restituisce, quindi, un risultato per il quale viene consigliata o meno una visita da uno specialista.

Per la realizzazione, si è ispirati ai problemi relativi alla dislessia caratterizzata dalla difficoltà di effettuare una lettura accurata e da abilità scadenti nella scrittura e nella decodifica. In generale il 20% [1] dei bimbi ha problemi di apprendimento una percentuale di questi soffre di dislessia. I problemi di lettura possono essere dovuti a svariate cause, quali, ad esempio: la stanchezza, al deficit di attenzione o a disturbi dello spettro autistico.

L'applicazione consiste di una serie di esercizi articolati in livelli. I livelli sono organizzati in modo tale da aumentare la lunghezza e la complessità della struttura ortografica delle parole e si suddividono in quattro sezioni:

1. Parole bisillabe;
2. Parole trisillabe;
3. Parole complesse bisillabe e trisillabe;
4. Parole complesse con gruppi ortografici;

Lo scopo di questa tesi è di realizzare un gioco da integrare alla fine di ogni sezione in modo da motivare i bambini in modo simpatico anche attraverso l'utilizzo di smartphone. A questo è stato aggiunto lo sviluppo di un'area admin da cui un amministratore può:

1. visualizzare gli errori commessi dai vari utenti

2. aggiungere nuovi livelli
3. rimuovere livelli esistenti nel caso in cui questi non si considerino più necessari.

Questo documento descrive tutte le motivazioni e le scelte intraprese per portare a termine l'obiettivo fissato; in particolare si propone un percorso che porta, dallo studio degli strumenti disponibili e dei meccanismi di funzionamento di essi stessi, alla risoluzione del problema seguendo ordinatamente le seguenti attività:

- Analisi dei requisiti, per definire le richieste del cliente;
- Studio di fattibilità, per comprendere i funzionamenti e i metodi necessari per il compimento del lavoro;
- Sviluppo software;

All'interno dei seguenti capitoli verrà raccontato l'intero mio studio.

Nel primo capitolo verrà discusso lo stato dell'arte relativo alla rilevazione di problemi di apprendimento nella lettura, con riferimento sia agli articoli scientifici, sia alle app presenti negli store nelle varie lingue.

Nel secondo capitolo sono descritti i concetti principali relativi ai Disturbi Specifici sull'Apprendimento e la legge di riferimento, uscita nel 2010.

Nel terzo capitolo viene presentato un approfondimento sul disturbo d'apprendimento centrale che è stato di ispirazione per il mio studio: la dislessia.

Nel quarto capitolo vengono descritti brevemente gli strumenti utilizzati per la realizzazione del mio progetto.

Nel quinto capitolo vengono descritti l'implementazione della base dati che verrà utilizzata durante lo sviluppo di tutto il progetto, l'implementazione del videogioco e l'implementazione dell'area amministratore, partendo dal sistema proposto fino alla realizzazione vera e propria. Presenterò, infatti, il funzionamento generale del sistema con degli esempi per rendere tutto più chiaro.

Nel sesto capitolo avremo le conclusioni, con le relative proposte future.

# Capitolo 1: Stato dell'arte

Nel presente capitolo verrà descritto lo stato dell'arte relativo alla rilevazione di problemi di apprendimento nella lettura, quindi, principalmente la dislessia, con riferimento sia agli articoli scientifici, sia alle app presenti negli store nelle varie lingue. Attualmente sono stati ritrovati diversi articoli, che attraverso la descrizione di propri lavori, riportano numeri e descrizione del problema, con riferimento a paesi diversi dall'Italia, ma che allo stesso tempo rendono l'idea di quale siano le proporzioni.

La dislessia è una disabilità nascosta. Le persone con dislessia non possono percepire se stanno leggendo o scrivendo correttamente. Infatti, la dislessia è una disabilità di lettura neurologica che è caratterizzata da difficoltà con riconoscimento di parole accurate e/o fluenti e da cattive capacità di ortografia e decodifica. Queste difficoltà derivano in genere da un deficit nella componente fonologica del linguaggio che spesso non è correlato ad altre disabilità cognitive. Conseguenze secondarie includono problemi nella comprensione della lettura. In Spagna, si stima che quattro casi su sei di insuccesso scolastico siano correlati a questa disabilità. Dal 10 al 17,5% della popolazione negli Stati Uniti e dall'8,6 all'11% della popolazione di lingua spagnola hanno questa disabilità cognitiva [2]. In tutto il mondo, circa il 15-20% della popolazione ha una disabilità di apprendimento basata sulla lingua. Probabilmente il 70-80% di loro ha dislessia [3].

Per quanto riguarda l'interferenza lessicale e fonologica, uno studio sperimentale ha dimostrato che questo è più alto negli studenti di 2 ° grado (6-7 anni), diminuendo negli studenti di 4 ° grado (8-9 anni) e inesistenti negli studenti di 6 ° grado (10-11 Anni). Questo stesso studio ha rivelato che l'automazione nel riconoscere le parole scritte avviene nel 4 ° grado. [4]



È questa fascia di età (6-10 anni) che verrà presa in esame nel mio studio.

Sono stati proposti diversi approcci per aiutare le persone dislessiche a leggere. Ad esempio, [2] propone un sistema che permette al lettore di trovare facilmente dei sinonimi per parole complesse, l'articolo valuta DysWebxia, un'app di lettura per dispositivi iOS. Il sistema analizza il testo e sottolinea le parole che possono essere più difficili per un dislessico e propone un link ad una lista di sinonimi. Per valutare DysWebxia hanno effettuato due diversi studi utente. Uno per valutare la qualità dei sinonimi su richiesta che includeva 32 partecipanti con dislessia e 38 lettori forti senza dislessia, e un altro per valutare l'usabilità dell'app basata su 12 partecipanti. I risultati mostrano che la qualità dei sinonimi generati dall'algoritmo supera la base di frequenza e che i partecipanti hanno trovato che DysWebxia è molto utilizzabile.

Nell'articolo [4] hanno effettuato una ricerca con lo scopo di migliorare la consapevolezza fonologica nei bambini con dislessia utilizzando un gioco per computer progettato specificamente per questo studio e applicandolo come risorsa didattica in un ambiente educativo formativo. Hanno utilizzato un metodo scientifico quasi sperimentale con un campione di 12 bambini, di età compresa tra 6 e 12 anni, divisi in due gruppi: sperimentale (sei) e controllo (sei). Applicando un test con la metodologia pre-test e post-test per l'intero campione. Le sessioni sperimentali sono durate circa 20 minuti al giorno, tre volte alla settimana, per cinque settimane, per un totale di 15 sessioni. La ricerca mostra differenze significative, con un livello di confidenza del 95%, nella popolazione sperimentale dopo l'applicazione delle strategie di gioco. Questi risultati superano quelli degli studi precedenti con risorse di gioco digitale e aprono la strada a una seconda fase di ricerca con una popolazione più ampia. L'obiettivo principale di questo studio è stato quello di convalidare

le risorse educative utilizzando un videogioco informatico come strumento per migliorare la consapevolezza fonologica nei bambini tra i 6 ei 10 anni di età. Hanno pianificato di determinare se l'applicazione (gioco) progettata ha migliorato il livello di lettura dei bambini in età scolare, e se può servire da guida per lo sviluppo di risorse aggiuntive che possono essere adattate ai videogiochi in modo da avvantaggiare gli studenti con dislessia.

Nell'articolo [5] presentano un metodo che mira a migliorare l'ortografia dei bambini con dislessia attraverso esercizi giocosi e mirati. In contrasto con gli approcci esistenti, il loro metodo non usa parole corrette o esempi positivi da seguire, ma presenta al bambino una parola errata come esercizio da risolvere. Hanno creato questi esercizi di allenamento sulla base delle conoscenze linguistiche estratte dagli errori trovati nei testi scritti da bambini con dislessia. Per verificare l'efficacia di questo metodo in spagnolo, hanno integrato gli esercizi in un gioco per iPad, DysEggxia (in spagnolo), e hanno effettuato un esperimento interno. Durante otto settimane, 48 bambini hanno giocato a DysEggxia oa Word Search, che è un altro gioco di parole. I bambini che hanno giocato a DysEggxia per quattro settimane di fila hanno avuto errori di scrittura significativamente meno nei test che, dopo aver giocato a Word Search per la stessa ora. Ciò dimostra che gli esercizi basati su errori presentati in una tavoletta aiutano i bambini con dislessia a migliorare le loro capacità di ortografia.

Attraverso l'articolo [6] spagnolo hanno fatto una ricerca sulle app esistenti per bambini con dislessia. Presentando anche ricerche sullo sviluppo dell'alfabetizzazione utilizzando quindici applicazioni per tablet selezionate per lavorare sull'alfabetizzazione e la matematica nei bambini con dislessia. Hanno utilizzato uno strumento ad-hoc che analizza i principali elementi delle app, il loro approccio pedagogico, le risorse del gioco e l'approccio alla

dislessia. Le quindici app selezionate sono in spagnolo e sono raccomandate dagli esperti di dislessia. I risultati che sottolineano sono che nessuna applicazione ha un approccio globale, la maggior parte delle app considera solo la lettura fluente o il riconoscimento automatico delle parole. I risultati sono riportati nella seguente tabella.

Google Play	AppStore	Android Market
252 applications, of which only 14 are in Spanish and related to the dyslexia, distributed: 4 readers of screen/correctors. 2 reading speed, 4 crosswords, 1 sources(fountains), 1 alphabet, 1 numbers, 1 written recording / reproduction	5 applications, of which only 3 are in Spanish and are related to the dyslexia, distributed: 1 activities orientated to the dyslexia 1 engraver / breeding(reproductive) 1 generic Activities	8 applications, of which only 3 are in Spanish and are distributed: 1 activities orientated to the dyslexia 1 alphabet 1 tables of multiplying

Anche in questo lavoro di tesi è stata effettuata una ricerca su Google Play Store e App Store per i-os. Sono state rilevate diverse applicazioni per i disturbi sull'apprendimento, molte propongono giochi riabilitativi, pochi test diagnostici, per lo più test in cui vengono fatte domande relative ai comportamenti, chiaramente non è possibile dire se effettivamente esiste un focolare evidente di uno specifico disturbo, perché esistono diverse motivazioni per le difficoltà di un individuo. Esistono pochi test che restituiscono un risultato in base ad un esercizio svolto, ma la pecca di queste applicazioni è che hanno come scopo finale l'obiettivo di venderti corsi a pagamento, e quindi la fiducia può venire a mancare in quanto un risultato, borderline potrebbe essere considerato facilmente come un risultato cattivo solo per fini di marketing. Di seguito qualche esempio:

- Dyslexia Test & Tips.

Questa applicazione in inglese ad esempio ha feedback molto negativi. Il test che offre è solo un questionario comprensivo.

- Dyslexia Reading Test,

Questa applicazione, in inglese, è carina, offre un test con durata 5-10 minuti, dove l'app legge ad alta voce sillabe e l'utente deve cliccare la corrispondente sillaba. Il problema è che ha l'obiettivo di vendere corsi a pagamento e segnala problemi anche se uno risponde esattamente l'80% delle volte.

# Capitolo 2: DSA- Disturbi specifici sull'apprendimento

I disturbi specifici sull'apprendimento con sigla (DSA) sono disturbi delle abilità scolastiche, di tipo settoriale, che riguardano difficoltà specifiche della lettura, scrittura e calcolo in presenza di una intelligenza nella norma. Fanno parte della famiglia dei Disturbi Evolutivi Specifici (DES) che comprende: deficit del linguaggio, delle abilità non verbali, della coordinazione motoria, dell'attenzione e dell'iperattività, mentre il funzionamento intellettivo limite può essere considerato un caso di confine fra la disabilità e il disturbo specifico.

Nonostante un quoziente intellettivo nella norma, la capacità di questi bambini di apprendere codici alfabetici e numerici resta limitata. Ciò significa che quasi tutti i bambini affetti da DSA riescono ad apprendere lettura, scrittura e calcolo, ma in modo non altrettanto automatizzato rispetto ai loro compagni. L'espressività di questi disturbi procede verso un graduale miglioramento del quadro funzionale, che può essere accelerato da interventi abilitativi tempestivi e mirati.

Nel 2010 fu stipulata una nuova legge, composta da ben 9 articoli, a sostegno di tale problematica, riconoscendo i vari disturbi e definendo l'uso di vari metodi per salvaguardare scolasticamente ragazzi che soffrono di disturbi sull'apprendimento.

**Legge 8 ottobre 2010, n.170** “Nuove norme in materia di disturbi specifici di apprendimento in ambito scolastico.”

Di seguito riporto solo gli articoli che sono stati di interesse per lo sviluppo della tesi.

## **Art.1 Riconoscimento e definizione di dislessia, disgrafia, disortografia e discalculia.**

La presente legge riconosce la dislessia, la disgrafia, la disortografia e la discalculia quali disturbi specifici di apprendimento, di seguito denominati «DSA», che si manifestano in presenza di capacità cognitive adeguate, in assenza di patologie neurologiche e di deficit sensoriali, ma possono costituire una limitazione importante per alcune attività della vita quotidiana.

Ai fini della presente legge, si intende per dislessia un disturbo specifico che si manifesta con una difficoltà nell'imparare a leggere, in particolare nella decifrazione dei segni linguistici, ovvero nella correttezza e nella rapidità della lettura.

Si intende per disgrafia un disturbo specifico di scrittura che si manifesta in difficoltà nella realizzazione grafica.

Si intende per disortografia un disturbo specifico di scrittura che si manifesta in difficoltà nei processi linguistici di transcodifica.

Si intende per discalculia un disturbo specifico che si manifesta con una difficoltà negli automatismi del calcolo e dell'elaborazione dei numeri.

La dislessia, la disgrafia, la disortografia e la discalculia possono sussistere separatamente o insieme.

## **Art.2 Finalità**

La seguente legge ha lo scopo di raggiungere le seguenti funzionalità:

- ❖ Garantire il diritto all'istruzione.
- ❖ Favorire successo scolastico, attraverso misure didattiche di supporto.
- ❖ Ridurre disagi relazionali ed emozionali.

- ❖ Adottare forme di verifica e di valutazione adeguate alle necessità formative degli studenti.
- ❖ Preparare professori e sensibilizzare genitori a riguardo.
- ❖ Favorire la diagnosi precoce e percorsi didattici riabilitativi.
- ❖ Incrementare la comunicazione e la collaborazione tra famiglia, scuola e servizi sanitari durante il percorso di istruzione e di formazione.
- ❖ Assicurare eguali opportunità di sviluppo delle capacità in ambito sociale e professionale

### **Art.3 Diagnosi**

La diagnosi dei DSA è effettuata nell'ambito dei trattamenti specialistici già assicurati dal Servizio sanitario nazionale a legislazione vigente ed è comunicata dalla famiglia alla scuola di appartenenza dello studente.

Se lo studente presenta problematiche, nonostante adeguate attività di recupero didattico mirato, la scuola deve informare la famiglia

È compito delle scuole di ogni ordine e grado, attivare, previa apposita comunicazione alle famiglie interessate, interventi tempestivi, idonei ad individuare i casi sospetti di DSA degli studenti, sulla base dei protocolli regionali di cui all'articolo 7, comma 1. L'esito di tali attività non costituisce, comunque, una diagnosi di DSA.

### **Art. 4 Formazione nella scuola**

Assicura la formazione da parte di docenti di qualsiasi grado riguardo alle problematiche relative ai DSA. Finalizzata ad acquisire la competenza per individuarne precocemente i segnali e la conseguente capacità di applicare strategie didattiche, metodologiche e valutative adeguate.

### **Art. 5 Misure educative e didattiche di supporto**

Gli studenti con diagnosi di DSA hanno diritto a fruire di appositi provvedimenti dispensativi e compensativi di flessibilità didattica nel corso dei cicli di istruzione e formazione e negli studi universitari.

### **Art. 6 Misure per i familiari**

I familiari fino al primo grado di studenti del primo ciclo dell'istruzione con DSA impegnati nell'assistenza alle attività scolastiche a casa hanno diritto di usufruire di orari di lavoro flessibili.

# Capitolo 3: Dislessia

Il disturbo di apprendimento protagonista per lo sviluppo della tesi è stata la dislessia. Andiamo a descrivere in maniera approfondita tale disturbo.

## Dislessia Evolutiva

La Dislessia Evolutiva è una disabilità specifica di origine neurobiologica. Essa è caratterizzata dalla difficoltà di effettuare una lettura accurata e da abilità scadenti nella scrittura e nella decodifica. Queste difficoltà tipicamente derivano da un deficit nella componente fonologica nel linguaggio che spesso è inattesa in rapporto alle altre abilità cognitive e alla garanzia di un'adeguata istruzione scolastica.

Conseguenze secondarie possono includere i problemi di comprensione nella lettura e una ridotta pratica della lettura che può impedire la crescita del vocabolario e della conoscenza generale.

Tipicamente il soggetto dislessico ha un quoziente intellettivo nella norma, lettura ad alta voce stentata, difficoltà ortografiche nella scrittura, difficoltà col sistema dei numeri e del calcolo e a volte sono presenti difficoltà di comprensione del testo, difficoltà nel linguaggio orale, instabilità motoria e disturbo di attenzione.

Si manifesta soprattutto in età infantile quando il bambino inizia la scuola primaria, infatti un bambino trova molta difficoltà nell'acquisizione del codice alfabetico e nell'applicazione delle "mappature" Grafema-Fonema e viceversa. Ha un controllo limitato delle operazioni di analisi e sintesi fonemica con errori che alterano in modo grossolano la scrittura fonologica delle parole lette o scritte. Accesso lessicale limitato o assente anche quando



la parola viene letta correttamente. Andando avanti nel percorso scolastico le cose migliorano gradualmente anche se resta evidente la problematica.

Sono la scuola, la famiglia, il pediatra a dover prestare attenzione, controllando le difficoltà nello svolgimento dei compiti (lentezza, svogliatezza ecc...), stando attenti ai disturbi della sfera emotiva/motivazionale (apatia, disturbi psicosomatici e comportamentali) e incongruenza tra capacità percepite (nella norma) e rendimento scolastico (scarso).

Prima di sottoporsi ad una valutazione clinica è importante rilevare presenza di familiarità per il disturbo (disturbi analoghi dei genitori/fratelli o livello scolastico dei genitori/fratelli), ricordare principali tappe dello sviluppo linguistico, storia scolastica del bambino.

La diagnosi clinica prevede osservazioni informali cioè l'osservazione generale del comportamento e degli elaborati scolastici del bambino, colloqui sulla percezione del problema e i relativi vissuti e infine valutazione psicometrica e neuropsicologica attraverso test. Importante sottolineare che una diagnosi vera non dovrebbe essere effettuata prima della 2° elementare in quanto un bambino con età compresa tra i 5 e i 7 anni può avere problemi di apprendimento anche per altri fattori quali una semplice difficoltà a concentrarsi [1].

Le tipologie di intervento variano in relazione all'eterogeneità dei profili funzionali e di sviluppo. Di conseguenza gli interventi possono essere:

#### ❖ **Preventivi**

Sono tutti quegli interventi mirati ad una identificazione precoce del disturbo e ad un rafforzamento delle abilità (soprattutto meta-fonologiche) necessarie all'acquisizione della lingua scritta ed un suo uso efficiente. Rivolto ai bambini che stanno per concludere la scuola dell'infanzia, si utilizzano

prove che valutano le abilità fonologiche (prove di decisione, di ripetizione) e meta-fonologiche (riconoscimento di rime e del suono iniziale della parola). I vantaggi di questo intervento sono avere un lavoro mirato e tempestivo sugli stessi prerequisiti dell'apprendimento della lettura-scrittura che facilita l'acquisizione e l'uso del codice alfabetico, contribuire e prevenire un insuccesso scolastico, evitare la catena di eventi negativi (colpevolizzazioni) che da esso spesso conseguono.

### ❖ **Riabilitativi**

Sono tutti quegli interventi che si indirizzano in senso stretto ad un recupero della funzione o di quelle sue componenti che risultano più deficitarie, attraverso cicli di esercitazioni mirate e specifiche. Rivolto a bambini che hanno già intrapreso la scuola primaria, la riabilitazione si pone obiettivi diversi in relazione alle diverse fasi di acquisizione dell'abilità di lettura e alla conseguente modificazione nell'espressione del disturbo: costruzione dell'abilità (acquisizione del codice alfabetico e del sistema di mappatura Grafema-Fonema, costruzione delle operazioni basilari di analisi e sintesi fonemica), l'automatizzazione dell'abilità (delle operazioni di analisi ortografica, delle operazioni di conversione Grafema-Fonema, delle operazioni di sintesi fonemica, dei processi di accesso lessicale) e lo sviluppo di strategie "top-down" (riconoscimento visivo e processi di accesso al lessico ortografico, utilizzo del contesto e processi di anticipazione lessicale, individuazione dei punti "focali" di un racconto).

### ❖ **Compensativi**

Sono tutti quegli interventi che si attuano in fase più avanzata del percorso scolastico (scuola media o oltre), nella situazione in cui il disturbo è più severo e ormai poco modificabile, per cui non è più fattibile ipotizzare un ripristino della funzione e diventa necessario trovare soluzioni alternative di supporto. Esistono numerosi tipi di ausilio più o meno sofisticati da un punto

di vista tecnologico come: Registratori, calcolatrici, sintesi vocale, correttori ortografici, riconoscimento di voce, enciclopedie multimediali, ecc...

# Capitolo 4: Software e linguaggi utilizzati

Durante lo sviluppo sono state realizzate due versioni differenti che andremo a descrivere nel dettaglio nei prossimi capitoli; durante lo sviluppo di queste due versioni, sono stati utilizzati diversi ambienti di sviluppo, quali:

- Unity per la realizzazione del videogioco;
- AndroidStudio per l'importazione del gioco sviluppato con Unity e la generazione dell'APK;
- XAMPP per lo sviluppo di pagine PHP per la gestione delle varie richieste da parte dei client;
- MySQL per la gestione del database relativo al gioco;
- Eclipse per lo sviluppo di Servlet che gestiscano le varie richieste dai clients, e per lo sviluppo di un'area Admin che permetta di gestire i vari livelli del gioco;
- Apache Tomcat come web-server.

Nei seguenti paragrafi verranno descritti nel dettaglio i vari ambienti e le relative tecnologie associate ad essi.

## Unity

Unity è un motore grafico multiplatforma sviluppato da Unity Technologies che consente lo sviluppo di videogiochi e altri contenuti interattivi, quali visualizzazioni architettoniche o animazioni 3D in tempo reale.

L'ambiente di sviluppo Unity gira sia su Microsoft Windows sia su MacOS e sia su Linux, e i giochi che produce possono essere eseguiti su Microsoft Windows, Mac, Linux, X-Box 360, X-Box One, PlayStation 3, PlayStation 4,

PlayStation Vita, Wii, iPhone, Android, Switch. Può anche produrre giochi per browser web che utilizzano il plugin Unity web player, supportate su Mac e Windows.

Unity consente di utilizzare sia un editor per lo sviluppo e progettazione di contenuti sia un motore di gioco per l'esecuzione del prodotto finale.

Il 12 novembre 2013 viene pubblicata la versione 4.3 che permette lo sviluppo facilitato di giochi bidimensionali e un notevole tool-kit per la creazione di GUI personalizzate.

Per lo sviluppo del videogioco sono stati implementati script scritti in C# che permettono principalmente di:

- gestire l'interazione con l'utente durante la partita;
- gestire il menu visualizzato all'utente;
- effettuare richieste al web-server;
- elaborare le risposte ricevute dal web server.

## **C#**

Il C# è un linguaggio di programmazione orientato agli oggetti sviluppato da Microsoft all'interno dell'iniziativa .NET, e successivamente approvato come standard dalla ECMA e ISO.

La sintassi e struttura del C# prendono spunto da vari linguaggi nati precedentemente, in particolare Delphi, C++ e Java.

Alcune delle caratteristiche di base sono:

- i nomi delle variabili, funzioni, classi ed altri elementi sono sempre case-sensitive, ovvero sensibili alle minuscole;
- ogni specifica deve essere chiusa dal carattere punto e virgola;
- le parentesi graffe sono usate per raggruppare specifiche;

- secondo le consuetudini dei linguaggi orientati agli oggetti, le specifiche sono di regola raggruppate in metodi, i metodi sono raggruppati in classi e le classi sono raggruppate nei namespace.

I metodi principalmente utilizzati durante il mio lavoro sono stati:

- Monobehaviour.Start() che viene eseguito, quando uno script è abilitato, poco prima che uno dei metodi Update() venga chiamato per la prima volta; durante il mio lavoro è stato utilizzato principalmente per inizializzare le varie variabili utilizzate come è possibile vedere nell'immagine sottostante;

```
void Start()
{
    numero_parole_caricate = 0;
    numero_frase_caricate = 0;
    progressi_da_aggiornare = true;
    progresso_aggiornato = false;
    gioco_completato = false;
    giochiamo = false;
    loginFallito = false;
    usernameEsistente = false;
    menu = true;
    registrazione = false;
    login = false;
    giocatoreOn = false;
    nome = "username";
    password = "password";
}
```

- Monobehaviour.Update() che viene chiamato ad ogni frame se lo script è abilitato; è stato utilizzato principalmente per recuperare

informazioni circa le azioni effettuate sul touch del dispositivo come è possibile vedere nell'immagine sottostante;

```
void Update()
{
    Vector3 vector2 = transform.localPosition;
    if (Input.touchCount > 0)
    {
        Touch touch = Input.GetTouch(0);
        switch (touch.phase)
        {
            //When a touch has first been detected, change the message and record the starting position
            case TouchPhase.Began:
                // Record initial touch position.
                Debug.Log("Inizio tocco");
                startPos = touch.position;
                break;

            //Determine if the touch is a moving touch
            case TouchPhase.Moved:
                // Determine direction by comparing the current touch position with the initial one
                Debug.Log("Mi sto muovendo");
                break;

            case TouchPhase.Ended:
                // Report that the touch has ended when it ends
                Debug.Log("Fine tocco");
                endPos = touch.position;
                if (startPos.x - endPos.x > 0f)
                {
                    if(transform.position.x == 0f)
                    {
                        Debug.Log("Startpos. " + startPos);
                        Debug.Log("Endpos" + endPos);
                        transform.position = new Vector3(-10f, transform.position.y, transform.position.z);
                    }
                    else if(transform.position.x == 10f)
                    {
                        Debug.Log("Startpos. " + startPos);
                        Debug.Log("Endpos" + endPos);
                        transform.position = new Vector3(0f, transform.position.y, transform.position.z);
                    }
                }
                else if (startPos.x - endPos.x < 0f)
                {
                    if(transform.position.x == 0f)
                    {
                        Debug.Log("Startpos. " + startPos);
                        Debug.Log("Endpos" + endPos);
                        transform.position = new Vector3(10f, transform.position.y, transform.position.z);
                    }
                    else if(transform.position.x == -10f)
                    {
                        Debug.Log("Startpos. " + startPos);
                        Debug.Log("Endpos" + endPos);
                        transform.position = new Vector3(0f, transform.position.y, transform.position.z);
                    }
                }
                break;
        }
    }
}
```

- MonoBehaviour.OnGui() che viene utilizzato per il rendering e la gestione degli eventi della GUI; in particolare, durante il mio lavoro è stato utilizzato principalmente per la gestione e generazione del menu di gioco (classe GUILayout) come è possibile vedere nell'immagine sottostante;

```
private void OnGUI()
{
    if (GestioneCollisone.perso)
    {
        if(istruzioni_perso)
        {
            GUILayout.BeginArea(new Rect(Screen.width / 2 - 250, Screen.height / 2 - 150, 500, 500));
            GUILayout.Label("Obiettivo", stileistruzioni1);
            GUILayout.Label("Seleziona le parole visualizzate in alto", stileistruzioni2);
            GUILayout.Label("Comandi", stileistruzioni1);
            GUILayout.Label("Trascina verso sinistra: sposta a sinistra", stileistruzioni2);
            GUILayout.Label("Trascina verso destra: sposta a destra", stileistruzioni2);
            if (GUILayout.Button("Torna al menu", stilebottoni))
            {
                istruzioni_perso = false;
            }
            GUILayout.EndArea();
        }
        else
        {
            GUILayout.BeginArea(new Rect(Screen.width / 2 - 250, Screen.height / 2 - 150, 500, 500));
            GUILayout.Label("Hai perso", stiletesto);
            if (GUILayout.Button("Riprova", stilebottoni))
            {
                SceneManager.LoadScene("SampleScene");
            }
            if (GUILayout.Button("Istruzioni", stilebottoni))
            {
                istruzioni_perso = true;
            }
            GUILayout.EndArea();
        }
    }
}
```



- MonoBehaviour.StartCoroutine(IEnumerator routine) che permette di mettere in pausa una funzione e restituire il controllo a Unity continuando da dove era stata interrotta nel frame seguente. Particolarmente utile quando una funzione, per essere eseguita, ha bisogno di più frame; è stata utilizzata principalmente per effettuare chiamate al web-server come è possibile vedere nell'immagine sottostante;

```
public void addGiocatore(string username, string password)
{
    StartCoroutine(Register(username, password));
}

IEnumerator Register(string username, string password)
{
    WWWForm form = new WWWForm();
    form.AddField("username", username);
    form.AddField("password", password);
    UnityWebRequest www = UnityWebRequest.Post("http://193.205.161.130/GiocoDislessia/Registrazione", form);
    yield return www.SendWebRequest();
    if (www.downloadHandler.text == "1")
    {
        Debug.Log("Registrazione riuscita");
        usernameEsistente = false;
        giocatore = new Giocatore(nome, password, 1);
        giocatoreOn = true;
        getFrase(giocatore.getProgresso());
    }
    else if (www.downloadHandler.text == "0")
    {
        Debug.Log("Connessione non riuscita");
    }
    else if (www.downloadHandler.text == "2")
    {
        Debug.Log("Username esistente");
        usernameEsistente = true;
    }
}
```

- Network.UnityWebRequest.Post(string uri, string postData) che prende come parametri un form e l'uri a cui verranno trasmessi i

dati del form; questo metodo permette di creare una `UnityWebRequest` configurata per inviare i dati del form a un server tramite `http POST`; è possibile vedere l'utilizzo di questo metodo nella precedente immagine;

- `Networkkink.UnityWebRequest.SendWebRequest()` che inizializza la comunicazione con il server remoto ed elabora la risposta del server; anche in questo caso è possibile vedere l'utilizzo di questo metodo nella precedente immagine.

## Touch e TouchPhase

Per gestire le azioni effettuate sul touch del dispositivo è stata utilizzata la struttura `Touch` che descrive lo stato di un dito che tocca lo schermo; permette di tenere traccia della fase del tocco, della sua posizione e del numero di tocchi.

`TouchPhase` è un tipo `enum` che descrive la fase del tocco di un dito; contiene gli stati dei possibili tocchi delle dita ed ogni stato rappresenta un'azione che il dito può eseguire sull'ultimo aggiornamento del frame.

Sono stati utilizzati principalmente i seguenti metodi e attributi:

- `Input.GetTouch(int index)` che prende come input il tocco effettuato sul dispositivo e che restituisce una struttura contenente i dettagli del tocco; nella foto seguente è possibile vedere l'utilizzo di questo metodo (da notare che prima di prendere il tocco in input, viene effettuato un controllo sul numero di tocchi presenti sullo schermo attraverso l'attributo `touchCount`; in particolare controlliamo che il numero di tocchi sia maggiore di 0);

```
if (Input.touchCount > 0)
{
    Touch touch = Input.GetTouch(0);
```

- Touch.phase che descrive la fase di un tocco, la quale può assumere i seguenti valori:

- 1) TouchPhase.Began che sta ad indicare che è stato rilevato un tocco;
- 2) TouchPhase.Moved che sta ad indicare che il tocco si sta spostando;
- 3) TouchPhase.Ended che sta ad indicare che il tocco è terminato e che il dito non si trova più sul device.

È possibile vedere un effettivo utilizzo delle tre fasi nelle immagini sottostanti (in base alla fase del tocco in cui si trova un dito vengono eseguire azioni differenti):

```
case TouchPhase.Began:  
    // Record initial touch position.  
    Debug.Log("Inizio tocco");  
    startPos = touch.position;  
    break;
```

In questo primo esempio si descrive il caso in cui il tocco viene rilevato ed in questo caso specifico viene memorizzata la posizione del tocco.

```
case TouchPhase.Moved:  
    // Determine direction by comparing the current touch position with the initial one  
    Debug.Log("Mi sto muovendo");  
    break;
```

In questo secondo esempio viene descritto il caso in cui la posizione iniziale del tocco subisce delle variazioni ed in questo caso specifico non facciamo altro che stampare un messaggio di Debug che sta ad indicare il movimento in corso.

```

case TouchPhase.Ended:
    // Report that the touch has ended when it ends
    Debug.Log("Fine tocco");
    endPos = touch.position;
    if (startPos.x - endPos.x > 0f)
    {
        if(transform.position.x == 0f)
        {
            Debug.Log("Startpos. " + startPos);
            Debug.Log("Endpos" + endPos);
            transform.position = new Vector3(-10f, transform.position.y, transform.position.z)
        }
        else if(transform.position.x == 10f)
        {
            Debug.Log("Startpos. " + startPos);
            Debug.Log("Endpos" + endPos);
            transform.position = new Vector3(0f, transform.position.y, transform.position.z);
        }
    }
}

```

In questo terzo esempio viene descritto il caso in cui il dito lasci lo schermo ed il tocco termini; per semplicità viene mostrato un piccolo pezzo di codice che descrive il caso in cui l'ultima posizione del dito sul touch sia più a destra rispetto la posizione del primo tocco del touch. Nel caso descritto si sposta l'oggetto nella nuova posizione.

## Android Studio

Android Studio è un ambiente di sviluppo integrato per lo sviluppo per la piattaforma Android. I linguaggi supportati da Android Studio sono Kotlin e Java. Durante il mio lavoro questo IDE è stato utilizzato principalmente per:

- importare il videogioco sviluppato con Unity;
- generare l'APK del videogioco importato.

## XAMPP

XAMPP è il più popolare ambiente di sviluppo PHP. XAMPP è una distribuzione di Apache completamente gratuita e semplice da installare, contenente MySQL, PHP e Perl. Mediante XAMPP è possibile avere un application server capace di interpretare pagine web dinamiche. Il software è

disponibile per sistemi unix-like (GNU/Linux, Oracle Solaris, MacOS) e per Microsoft Windows.

## **MySQL**

MySQL è un RDBMS (relational database management system) composto da un client a riga di comando e un server. MySQL è un software libero ed è sviluppato per essere il più possibile conforme allo standard ANSI SQL e ODBC SQL ed è stato utilizzato per la gestione del database relazionale.

## **SQL**

SQL è un linguaggio standardizzato per database basati sul modello relazionale progettato per le seguenti operazioni:

- creazione e modifica di database (DDL);
- inserimento, modifica e gestione dei dati memorizzati (DML);
- interrogazione dei dati memorizzati (DQL);

## **Eclipse**

Eclipse è un ambiente di sviluppo integrato multi-linguaggio e multiplatforma.

Eclipse può essere utilizzato per la produzione di software di vario genere, si passa infatti da un completo IDE per il linguaggio Java a un ambiente di sviluppo per il linguaggio C++.

Durante il mio lavoro è stato utilizzato principalmente per lo sviluppo di Servlet e JSP scritte in Java che gestiscano le varie richieste da parte dei client e per lo sviluppo di un'area amministratore.

## **Servlet e JSP**

JavaServer Pages è una tecnologia di programmazione Web in Java per lo sviluppo della logica di presentazione di applicazioni Web che fornisce contenuti dinamici in formato HTML o XML. Una pagina JSP può essere vista come una rappresentazione ad alto livello di una Servlet in quanto le

pagine JSP vengono tradotte automaticamente da un compilatore JSP in Servlet.

Le Servlet sono oggetti scritti in linguaggio Java che operano all'interno di un server web oppure un server per applicazioni permettendo la creazione di applicazioni web.

## **Apache Tomcat**

Apache Tomcat è un web-server open-source sviluppato dalla Apache Software Foundation. Implementa le specifiche JavaServer Pages (JSP) e Servlet, fornendo quindi una piattaforma software per l'esecuzione di applicazione Web sviluppate in linguaggio Java.

Ad implementare le specifiche per le Servlet Java e le JavaServer Pages è Catalina. Catalina è il contenitore di Servlet Java di Tomcat.

# Capitolo 5: Sistema proposto

Si vuole realizzare un videogioco che ha come scopo la possibilità di sostenere bambini con dislessia ed un'area admin che permetta di gestire i vari livelli all'interno del gioco e visualizzare gli errori ed i progressi effettuati dai giocatori.

Per realizzare il mio lavoro sono state sviluppate due versioni differenti; la prima versione prevedeva solamente lo sviluppo del videogioco con l'utilizzo di pagine sviluppate in PHP per la gestione delle richieste inviate dai vari client.

Con lo sviluppo della seconda versione, è stato deciso un cambio di tecnologia e le pagine PHP sono state sostituite gradualmente da Servlet. È stata effettuata questa scelta per poter riutilizzare un server Apache Tomcat già online e risparmiare nell'acquisto di un hosting che supporti PHP.

Oltre al cambio di tecnologie, è solamente in questa seconda versione che è stata sviluppata l'area amministratore. Nella precedente versione per poter testare effettivamente le varie funzionalità, le istanze delle varie tabelle venivano inserite durante l'esecuzione dello script per la creazione del database.

Per realizzare questa seconda versione è stato utilizzato come pattern architetturale il modello MVC che consente di separare la logica di presentazione dei dati dalla logica di business. MVC è basato principalmente su tre componenti software che interpretano tre ruoli principali:

- Model che fornisce i metodi per accedere ai dati utili all'applicazione e che si occupa di effettuare la connessione al database;
- View che permette di visualizzare i dati e gestire l'interazione tra utente ed oggetti;

- Control che riceve comandi dall'utente ed interagisce con le due componenti portando generalmente ad un cambiamento di stato della View.

Nello sviluppo dell'area amministratore, le View sono rappresentate dalle pagine JSP mentre nello sviluppo del videogioco le view sono rappresentate dalle scene implementate in Unity ed i relativi script associati ad esse. Sia gli script che le scene hanno il compito di:

1. effettuare richieste al web-server per richiedere o memorizzare dati;
2. inviare dei dati memorizzati all'interno di un form durante la richiesta al web server;
3. modificare la vista dell'utente.

Nello sviluppo di entrambi i prodotti, i Control sono rappresentati dalle Servlet che:

1. ricevono chiamate dalle JSP implementate con Java o dagli script implementati con C#;
2. recuperano i dati memorizzati all'interno del form;
3. effettuano la chiamata al metodo, contenuto nei manager, che si occupa di eseguire la query corrispondente;
4. ricevono i dati, richiesti al metodo richiamato;
5. restituiscono il controllo ad una JSP o script.

I Model sono rappresentati dai bean, dalla classe che effettua la connessione al database e dai manager che contengono i metodi che andranno ad interrogare il database; quindi il model si occuperà di:

1. effettuare la connessione al database;
2. nel caso in cui la connessione al database non venga effettuata con successo, comunicare il problema alle Servlet;



3. nel caso in cui la connessione al database avvenga con successo, effettuare le query (inserimento, modifica o selezione, rimozione) sul database;
4. restituire i risultati dell'esecuzione delle query alle Servlet.

## **Progettazione base dati**

Il primo passo effettuato durante il lavoro svolto è stato progettare un database che permetta di memorizzare, aggiornare ed eliminare dati. Questo primo passo è composto da varie fasi che possono sintetizzarsi in:

1. raccolta delle specifiche della realtà di interesse;
2. progettazione modello ER;
3. progettazione schema logico-relazionale;
4. implementazione dello script.

## **Specifiche della realtà di interesse**

Durante questa fase si è cercato di raccogliere quante più informazioni possibili riguardo le specifiche della realtà di interesse. Come specificato nei precedenti capitoli l'obiettivo era di creare un videogioco per aiutare bambini che soffrono di dislessia e di creare un'area admin che permetta di gestire i vari livelli del gioco.

Il gioco doveva essere suddiviso in quattro livelli che consistono nelle quattro sezioni presenti nell'app DSA App. Questi livelli sono organizzati in base alla struttura ortografica delle parole ed all'incremento della lunghezza delle stesse; possono riassumersi in:

1. parole bisillabe;
2. parole trisillabe;
3. parole complesse bisillabe e trisillabe;
4. parole complesse con gruppi ortografici.

Ciascun livello può contenere più frasi contenenti parole con le adeguate caratteristiche ortografiche. A queste parole possono essere associati dei

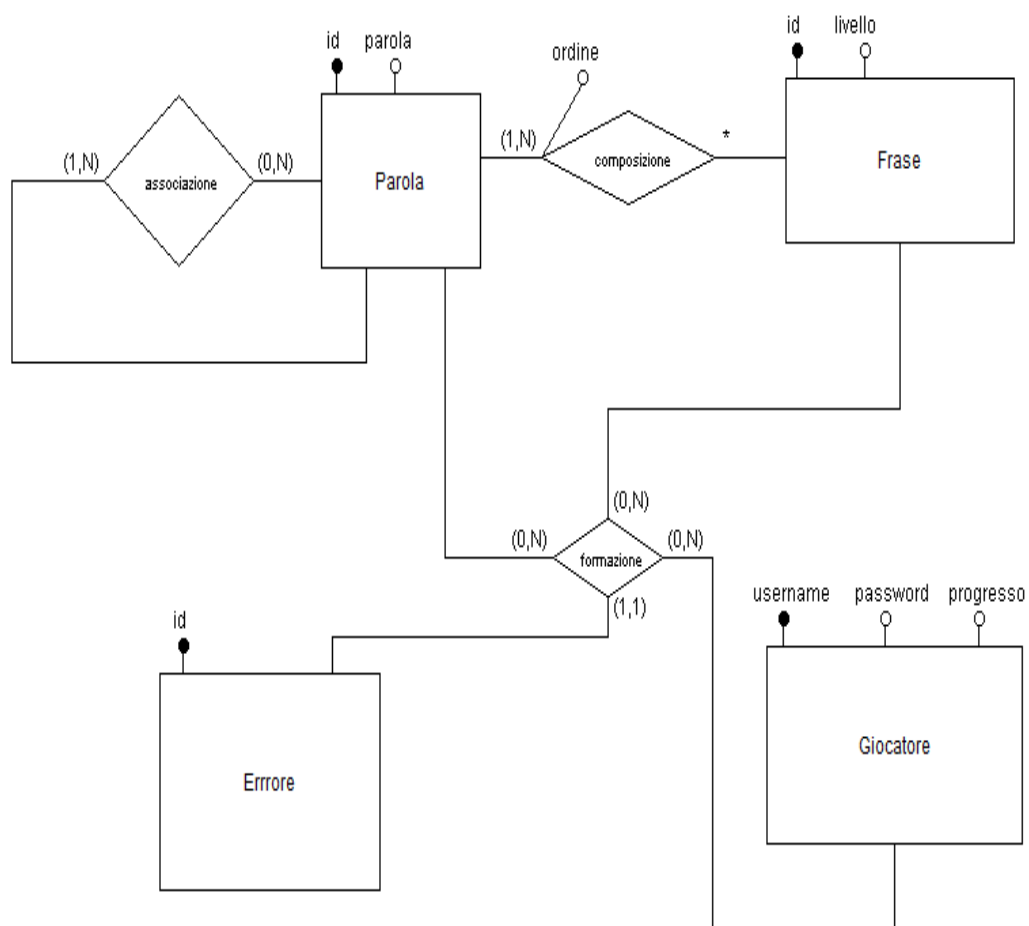
distrattori, ovvero, parole molto simili a quella a cui sono associate. I distrattori hanno come scopo di indurre il giocatore all'errore.

Durante questa analisi si è accorti della necessità di tenere traccia delle credenziali di accesso dei vari utenti e dei relativi progressi raggiunti.

A tutto ciò si è aggiunta la necessità di tenere traccia anche dei vari errori commessi dagli utenti per poter permettere di analizzare tutto a chi di competenza.

## Progettazione modello ER

Tenendo presente di tutte le informazioni raccolte durante la precedente fase, è stato progettato il modello ER allegato nella foto sottostante.



Come è possibile vedere sono state individuate quattro entità e tre relazioni che corrispondono rispettivamente alle entità Parola, Frase,

Giocatore ed Errore ed alle relazioni Formazione, Composizione ed Associazione. Ecco una breve descrizione di ogni entità e relazione e dei motivi per cui si è scelta la seguente soluzione:

- Parola è un'entità che rappresenta le parole all'interno del database; ha come attributo un id per identificare ogni parola e 'parola' che rappresenta la stringa. Questa entità, oltre che a rappresentare le parole presenti all'interno delle frasi, rappresenta anche i distrattori. Si è deciso di non utilizzare entità separate per questi due elementi per motivi di efficienza, in quanto un distrattore di una determinata parola potrebbe rappresentare una parola di un'altra frase o viceversa. In questo modo non andremo a memorizzare le stesse parole in due entità differenti.
- Associazione è una relazione che rappresenta l'associazione tra Parola ed un distrattore (Parola). Ovviamente ad una parola possono corrispondere zero o più distrattori mentre un distrattore deve essere associato ad almeno una parola.
- Frase è un'entità che rappresenta le frasi. Questa entità ha come attributo un id per identificare ogni frase e livello per indicare la frase a quale dei quattro livelli appartiene.
- Composizione è una relazione che rappresenta l'associazione tra Frase e Parola. Ad una frase possono corrispondere più parole ed è per questo che per ogni parola appartenente alla frase serve tener traccia anche della posizione della parola all'interno della frase. Per fare ciò utilizziamo 'ordine' come attributo della relazione; infine possiamo notare che una parola può comunque far parte di più frasi.
- Giocatore è un'entità che rappresenta il giocatore. Questa entità ha come attributo 'username' per identificare ogni singolo giocatore, 'password' che serve ad effettuare l'autenticazione per un

determinato username e progresso che sta ad indicare un determinato giocatore quale livello del gioco ha raggiunto.

- Errore è l'entità che rappresenta gli errori. Questa entità ha come attributo solamente un id per identificare ogni singolo errore.
- Formazione è una relazione che rappresenta l'associazione tra Errore, Parola, Frase e Giocatore. Come è facile intuire questa relazione tiene traccia degli errori commessi dai giocatori nelle varie frasi tenendo traccia anche di quale parola ha indotto all'errore. Ovviamente un giocatore può commettere zero o più errori in zero o più frasi per via di zero o più parole all'interno della frase. Quindi ad ogni frase possono essere associati zero o più errori commessi da zero o più giocatori su zero o più parole. Lo stesso discorso va fatto anche sulle parole. Infine uno specifico errore può essere commesso da un determinato giocatore in una determinata frase per via di una determinata parola.

## Progettazione schema logico-relazionale

Dopo la fase di progettazione del modello entità-relazioni è iniziata la fase di progettazione dello schema logico-relazionale.

Questa fase prende in input il modello della fase precedente ed al seguito di alcune decisioni si è ottenuto lo schema logico relazionale dove le parole in grassetto stanno ad indicare i nomi delle tabelle e le parentesi tonde racchiudono gli attributi delle tabelle; gli attributi che indicano una chiave primaria sono stati evidenziati con del testo sottolineato e gli attributi esterni sono evidenziati dalla presenza del simbolo '#'. Ecco in dettaglio lo schema ottenuto:

- **Parola** (id, parola)
- **Frase** (id, livello)
- **Giocatore** (username, password, progresso)

- **Errore** (id, #parola, #distrattore, #frase, #giocatore)

In questa tabella esistono vincoli di integrità referenziale rispettivamente tra gli attributi esterni #parola, #distrattore, #frase e #giocatore e gli attributi id della tabella Parola, id della tabella Parola, id della tabella Frase ed username della tabella Giocatore; in caso di rimozione o aggiornamento di un giocatore o di una frase è prevista l'opzione cascade.

- **Composizione** (id, #frase, #parola, ordine)

In questa tabella esistono vincoli di integrità referenziale rispettivamente tra gli attributi esterni #frase e #parola e gli attributi id della tabella Frase ed id della tabella Parola; in caso di rimozione o aggiornamento di una frase è prevista l'opzione cascade. È stato deciso di aggiungere un attributo id come chiave primaria all'interno della stessa per identificare facilmente in modo univoco le varie parole associate ad ogni frase.

- **Associazione** (id, #parola, #distrattore)

In questa tabella esistono vincoli di integrità referenziale rispettivamente tra gli attributi esterni #parola e #distrattore e gli attributi id della tabella Parola ed id sempre della tabella Parola; in caso di rimozione o aggiornamento di una parola è prevista l'opzione cascade. È stato deciso anche in questo caso di aggiungere un attributo id come chiave primaria per identificare facilmente in modo univoco i distrattori associati alle varie parole.

## Implementazione dello script

Ottenuto lo schema logico-relazionale dalla precedente fase, si è passati all'implementazione dello script SQL. Ecco in dettaglio lo script ottenuto:

```
Drop database if exists dislessia;  
create database dislessia;  
use dislessia;  
set global max_allowed_packet=100*1024*1024;
```

In questa parte dello script ci si preoccupa di:

- eliminare il database 'dislessia' nel caso in cui già esista;
- creare il database 'dislessia';
- utilizzare il database appena creato.

```
drop table if exists Frase;  
  
create table if not exists Frase(  
id integer auto_increment primary key,  
livello integer not null  
);
```

In questa parte dello script ci si preoccupa di:

- eliminare la tabella 'Frase' nel caso in cui già esista;
- creare la tabella 'Frase';

```
drop table if exists Parola;  
  
create table if not exists Parola(  
id integer auto_increment primary key,  
parola varchar(70) not null  
);
```

In questa parte dello script ci si preoccupa di:

- eliminare la tabella 'Parola' nel caso in cui già esista;
- creare la tabella 'Parola';

```
drop table if exists Giocatore;

create table if not exists Giocatore(
username varchar(16) primary key,
password varchar(16) not null,
progresso integer not null
);
```

In questa parte dello script ci si preoccupa di:

- eliminare la tabella 'Giocatore' nel caso in cui già esista;
- creare la tabella 'Giocatore';

```
drop table if exists Errore;

create table if not exists Errore(
id integer auto_increment primary key,
frase integer not null,
parola integer not null,
distrattore integer not null,
giocatore varchar(16) not null,
foreign key (frase) references Frase(id)
on update cascade
on delete cascade,
foreign key (parola) references Parola(id)
on update cascade
on delete cascade,
foreign key (distrattore) references Parola(id)
on update cascade
on delete cascade,
foreign key (giocatore) references Giocatore(username)
on update cascade
on delete cascade
);
```

In questa parte dello script ci si preoccupa di:

- eliminare la tabella 'Errore' nel caso in cui già esista;
- creare la tabella 'Errore';

- creare i vincoli di integrità referenziale imposti nella fase precedente;
- implementare il comportamento, definito nella fase precedente, nel caso di eliminazione o aggiornamento di elementi dalle tabelle con vincoli di integrità referenziale;

```
drop table if exists Composizione;

create table if not exists Composizione(
id integer auto_increment primary key,
frase integer not null,
parola integer not null,
ordine integer not null,
foreign key (frase) references Frase(id)
on update cascade
on delete cascade,
foreign key (parola) references Parola(id)
on update cascade
on delete cascade
);
```

In questa parte dello script ci si preoccupa di:

- eliminare la tabella ‘Composizione’ nel caso in cui già esista;
- creare la tabella ‘Composizione’;
- creare i vincoli di integrità referenziale imposti nella fase precedente;
- implementare il comportamento, definito nella fase precedente, nel caso di eliminazione o aggiornamento di elementi dalle tabelle con vincoli di integrità referenziale;



```
drop table if exists Associazione;

create table if not exists Associazione(
id integer auto_increment primary key,
parola integer not null,
distrattore integer not null,
foreign key (parola) references Parola(id)
on update cascade
on delete cascade,
foreign key (distrattore) references Parola(id)
on update cascade
on delete cascade
);
```

In questa parte dello script ci si preoccupa di:

- eliminare la tabella ‘Associazione’ nel caso in cui già esista;
- creare la tabella ‘Associazione’;
- creare i vincoli di integrità referenziale imposti nella fase precedente;
- implementare il comportamento, definito nella fase precedente, nel caso di eliminazione o aggiornamento di elementi dalle tabelle con vincoli di integrità referenziale;

## Requisiti Funzionali

Durante lo sviluppo della prima versione sono stati individuati gli attori principali del videogioco che corrispondono in linea generale ai giocatori ed alcuni requisiti funzionali che il sistema deve garantire ad essi, quali:

- R01 – Login e caricamento dei dati
- R02 – Registrazione
- R03 – Nuova partita
- R04 – Riprovare livello
- R05 – Uscire dalla partita
- R06 – Visualizzare i comandi
- R07 – Salvataggio dei progressi

R01 – Login e caricamento dei dati consiste nel garantire ai vari giocatori di poter effettuare l'accesso al videogioco con le opportune credenziali scelte in fase di registrazione e di poter recuperare i dati associati ad essi.

R02 – Registrazione consiste nel garantire ai vari giocatori di poter effettuare la registrazione al videogioco inserendo opportune credenziali in modo da poter memorizzare i progressi effettuati.

R03 – Nuova partita consiste nel garantire al giocatore di iniziare una nuova partita una volta effettuato il login con le opportune credenziali.

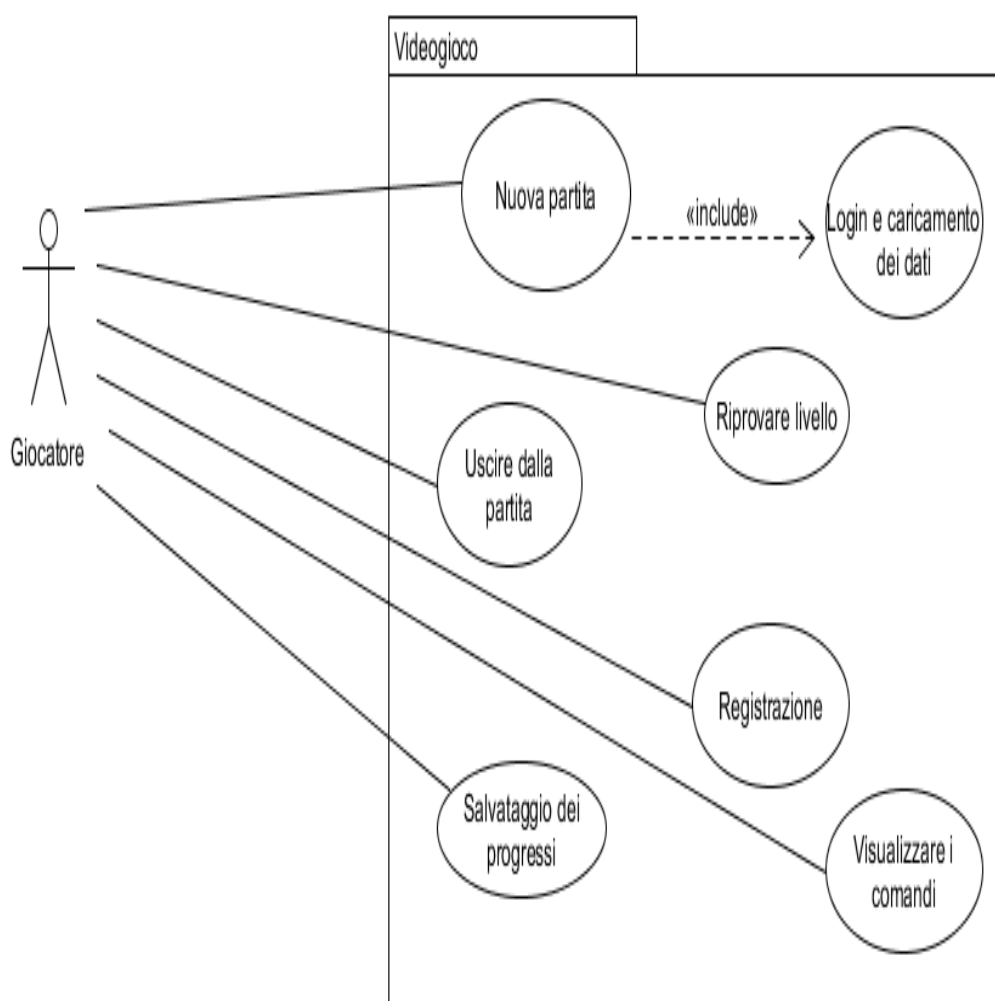
R04 – Riprovare livello consiste nel garantire al giocatore di ritentare a passare al livello successivo nel caso in cui questo perda una partita. Nel fornire questa possibilità si deve fare in modo che venga caricata ogni volta una frase random.

R05 – Uscire dalla partita consiste nel garantire al giocatore di uscire dalla partita nel caso in cui non volesse più giocare.

R06 – Visualizzare i comandi consiste nel garantire al giocatore di visualizzare gli obiettivi del gioco ed i comandi relativi ad esso. Questa possibilità deve essere data prima di effettuare il login o la registrazione, dopo aver effettuato il login, dopo aver effettuato la registrazione e dopo che un giocatore perde una partita e gli viene mostrato il menu di pausa.

R07 – Salvataggio dei progressi consiste nel garantire al giocatore di salvare i propri progressi.

Tutto ciò è riassunto nel seguente Use Case Diagram.



Durante lo sviluppo della seconda versione non sono stati modificati o aggiunti requisiti funzionali riguardanti il videogioco. Per quanto riguarda lo sviluppo dell'area admin, invece, è stato individuato un unico attore che

possiamo identificare come 'amministratore'. L'amministratore deve occuparsi di gestire l'area admin e sono stati individuati alcuni requisiti funzionali da garantire ad essi, quali:

- R08 – Login admin
- R09 – Visualizzazione degli errori
- R10 – Visualizzazione dei giocatori
- R11 – Visualizzazione delle frasi
- R12 – Visualizzazione dettagli frase
- R13 – Eliminazione frase
- R14 – Inserimento frase

R08 – Login admin consiste nel garantire all'amministratore di poter effettuare l'accesso all'area admin con le opportune credenziali.

R09 – Visualizzazione degli errori consiste nel garantire all'amministratore la possibilità di visualizzare l'elenco degli errori effettuati da un determinato giocatore durante l'utilizzo del videogioco.

R10 – Visualizzazione dei giocatori consiste nel garantire all'amministratore la possibilità di visualizzare l'elenco dei giocatori registrati al videogioco.

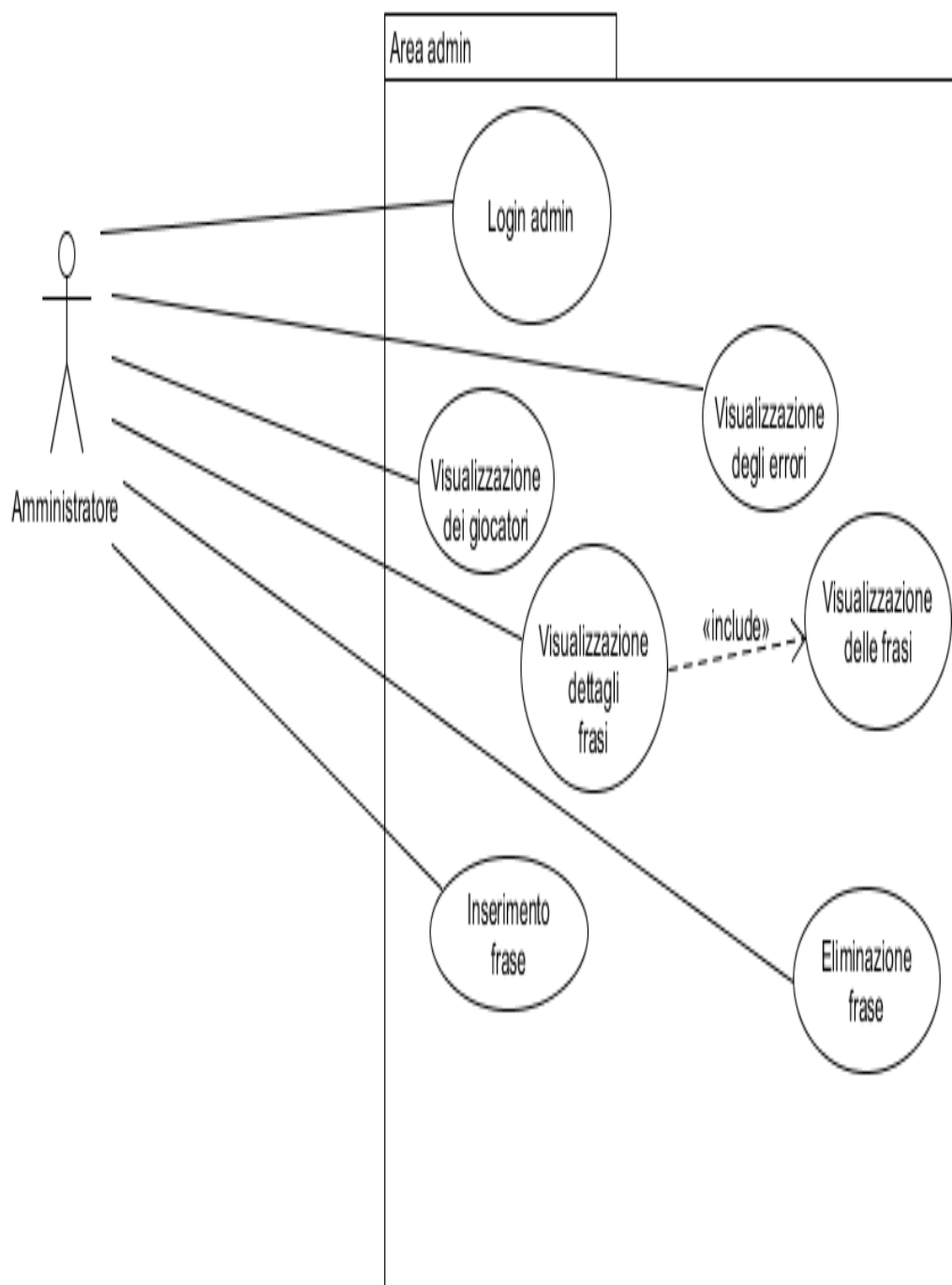
R11 – Visualizzazione delle frasi consiste nel garantire all'amministratore la possibilità di visualizzare l'elenco delle frasi presenti all'interno del videogioco.

R12 – Visualizzazione dettagli frase consiste nel garantire all'amministratore la possibilità di visualizzare in dettaglio una determinata frase presente all'interno del videogioco.

R13 – Eliminazione frase consiste nel garantire all'amministratore la possibilità di eliminare una frase presente all'interno del videogioco.

R14 – Inserimento frase consiste nel garantire all'amministratore la possibilità di inserire una frase nuova all'interno del videogioco.

Tutto ciò è riassunto nel seguente Use Case Diagram.



## **Requisiti non funzionali**

Sono stati individuati alcuni requisiti non funzionali che per comodità andrò ad elencare in due sezioni distinte:

- requisiti di prodotto;
- requisiti organizzativi;

## **Requisiti di prodotto**

Durante lo sviluppo della prima versione è stato individuato un unico requisito di prodotto e durante lo sviluppo della seconda versione questo non ha subito modifiche o aggiunte. Ecco il requisito nello specifico:

- Performance: il sistema permetterà di svolgere le operazioni di una singola funzionalità per volta in modo da evitare inconsistenza dei dati.

## **Requisiti organizzativi**

Anche i requisiti organizzativi sono stati individuati durante lo sviluppo della prima versione del software e non hanno subito modifiche nel tempo. Ecco in dettaglio i requisiti organizzativi individuati:

- Supportabilità: il sistema potrà supportare aggiornamenti e miglioramenti nel tempo.
- Implementazione: l'applicazione funzionerà su dispositivi mobile con sistema operativo in Android con possibilità di estensione su ulteriori piattaforme in un futuro.

## **Scenari**

In questa sezione mostrerò alcuni scenari che descrivono un tipico utilizzo del videogioco sviluppato nella prima versione e dell'area amministratore sviluppata nella seconda versione. Da notare che nella seconda versione sono cambiate solamente le tecnologie utilizzate del videogioco, mentre le funzionalità sono rimaste essenzialmente inalterate.

<b><i>Nome scenario</i></b>	Nuova partita	
<b><i>Partecipante</i></b>	Mario: Bambino	
<b><i>Flusso degli eventi</i></b>	<b><i>Utente:</i></b> <b>1.</b> Mario apre il videogioco.  <b>3.</b> Mario clicca sul pulsante login.  <b>5.</b> Mario inserisce: username e password e clicca sul pulsante accedi  <b>7.</b> Mario clicca sul pulsante gioco.	<b><i>Sistema:</i></b> <b>2.</b> Il sistema mostra il menu iniziale.  <b>4.</b> Il sistema chiede all'utente di inserire username e password.  <b>6.</b> Il sistema mostra un nuovo menu dove viene mostrato il pulsante gioca se l'autenticazione è avvenuta con successo.  <b>8.</b> Il sistema lancia il gioco.

Lo scenario mostrato nella foto precedente mostra l'interazione tra l'utente ed il sistema per il requisito inizia nuova partita.

<b><i>Nome scenario</i></b>	Visualizzazione delle frasi	
<b><i>Partecipante</i></b>	Luigi: Amministratore	
<b><i>Flusso degli eventi</i></b>	<b><i>Utente:</i></b> <b>1.</b> Luigi apre la pagina web.  <b>3.</b> Luigi digita l'username e la password.  <b>5.</b> Luigi seleziona la sezione frasi.	<b><i>Sistema:</i></b> <b>2.</b> Il sistema chiede di inserire le credenziali.  <b>4.</b> Il sistema mostra la pagina principale dell'amministratore.  <b>6.</b> Il sistema mostra a Luigi l'elenco di tutte le frasi presenti nel videogioco.

Lo scenario mostrato nella foto precedente mostra l'interazione tra l'amministratore ed il sistema per visualizzare l'elenco delle frasi presenti nel videogioco.

## **Descrizione delle interfacce**

Nella seguente sezione andremo a descrivere l'interfaccia grafica dell'area amministratore e del videogioco.



L'immagine precedente raffigura l'interfaccia della pagina iniziale del videogioco. Come è possibile vedere è composta da quattro pulsanti che si riassumono in:

- Registrati e gioca;
- Login e gioca;
- Istruzioni;
- Esci.





---

Frase

Errori

Giocatori

---

## Presentazione sito

Tale area admin è stata implementata per fare in modo che l'amministratore del gioco possa visualizzare gli errori effettuati dai giocatori, visualizzare i giocatori registrati al gioco ed aggiungere o rimuovere livelli dal gioco

---



L'immagine precedente raffigura l'interfaccia della pagina iniziale dell'area amministratore. Come è possibile vedere è composta dal logo, una piccola introduzione sulle funzionalità dell'area e tre sezioni che possono riassumersi in:

- Frasi;
- Giocatori;
- Errori

## Implementazione prima versione

Come detto nei precedenti paragrafi, durante lo sviluppo della prima versione del videogioco è stato utilizzato Unity per lo sviluppo del videogioco, MySQL per la gestione del database e pagine PHP per la gestione delle richieste da parte dei vari client che utilizzano il videogioco.

In questo paragrafo mostrerò un esempio di implementazione di funzionalità del videogioco; in particolare mostrerò come l'app interagisce con il server sviluppato in PHP.

```
public void addGiocatore(string username, string password)
{
    StartCoroutine(Register(username, password));
}

IEnumerator Register(string username, string password)
{
    WWWForm form = new WWWForm();
    form.AddField("username", username);
    form.AddField("password", password);
    UnityWebRequest www = UnityWebRequest.Post("https://dislessia.000webhostapp.com/register.php", form);
    yield return www.SendWebRequest();
    if (www.downloadHandler.text == "0")
    {
        Debug.Log("Registrazione riuscita");
        usernameEsistente = false;
        giocatore = new Giocatore(nome, password, 1);
        giocatore.On = true;
        getFrase(giocatore.getProgresso());
    }
    else if (www.downloadHandler.text == "1")
    {
        Debug.Log("Connessione non riuscita");
    }
    else if (www.downloadHandler.text == "2")
    {
        Debug.Log("Username esistente");
        usernameEsistente = true;
    }
}
```

La foto in alto mostra come viene effettuata la chiamata alla pagina PHP attraverso la classe UnityWebRequest di cui abbiamo parlato nei capitoli precedenti; da notare che in base a quello che viene restituito dalla pagina, lo script reagisce in modi differenti.

La foto in basso mostra come la pagina PHP reagisce alla richiesta di un client. Da notare che:

- cerca di effettuare la connessione al database e stampa un elemento che sta ad indicare un errore nella connessione nel caso in cui questa non riesca;
- se la connessione è stata effettuata con successo, cerca di eseguire una query sul database;
- se l'esecuzione della query avviene con successo, stampa un elemento che sta ad indicare la riuscita operazione di login;
- se l'esecuzione della query non porta alla corrispondenza di elementi all'interno del database, viene stampato un elemento che sta ad indicare che i dati non sono corretti.

```
<?php

$con=mysqli_connect('localhost','id10381116_dislessia','dislessia','id10381116_dislessia');
if(!$con)
{
    echo("-1");
    exit();
}

$username=$_POST["username"];
$password=$_POST["password"];

$search="SELECT progresso FROM Giocatore WHERE username='".$username.'" AND password='".$password.'"';
$check=mysqli_query($con, $search) or die("Ricerca fallita");
if(mysqli_num_rows($check) == 0)
{
    echo("0");
    exit();
}
else
{
    $row = mysqli_fetch_assoc($check);
    echo("". $row["progresso"]);
}

mysqli_close($con);

?>
```

Durante lo sviluppo della prima versione non era ancora stata sviluppata un'area amministratore che permetteva di inserire o rimuovere dinamicamente le varie frasi presenti all'interno del videogioco. È per questo

motivo che le frasi venivano inserite staticamente durante la creazione del database stesso.

Di seguito mostreremo alcune frasi inserite all'interno del gioco e le suddivideremo per livelli:

- Frasi con parole bisillabe (livello 1):

*Mara vede la luna*  
*Lara beve e Sara ride*  
*la mela e la pera sono vere*  
*la nave è del re*  
*il dado è rosa*  
*cora tira la vela nera*  
*il lupo vede il gufo*  
*il pane è di Sara*  
*la mia mano è rosa*  
*io lavo i cubi neri*  
*la lana blu e rosa*  
*il gufo è sul ramo*  
*la tua moto fa fumo*  
*il pane è un dono*  
*le pere sono cadute*  
*il topo nero con la coda rosa*  
*tara fa il sugo buono*  
*il mago sale e la fata vola*  
*devi dare il riso e il pane*  
*la rana vive sul lago*  
*il geco vive sul muro*

- Frasi con parole trisillabe (livello 2):

*il papà riposa sul divano*  
*si vedono le nuvole*  
*vedo un melone e le susine*  
*il pirata lega le reti*  
*bea è una pecora carina*  
*le matite sono sul tavolo*  
*uso patate limone e carote*  
*il papà fuma il sigaro*  
*lava le mani con il sapone*

*la casa e il mulino*  
*il motore tira i vagoni*  
*la sirena è la regina del mare*  
*la mia camera è rosa*  
*il bufalo e un macaco vagano*  
*Maria cucina per la cena*  
*il pirata sapeva dove era il tesoro*  
*Davide cucina un panino*  
*la medusa rosa fa male*  
*Nicola ama il gelato al limone*  
*il limone è amaro*  
*Nicola ripara la bici di Milena*

- Frasi con parole complesse bisillabe e trisillabe (livello 3):

*la borsa profuma di fiori*  
*la festa di Carmela è stata divertente*  
*il gioco tira la corda*  
*silvana è la zia grande*  
*la strada da seguire è questa*  
*il luogo di ritrovo è nel bosco*  
*la scatola contiene i colori*  
*in vacanza porto il costume*  
*le mandorle sono buone*  
*la candela si è consumata*  
*il fantasma fa molta paura*  
*fausta prende la patente per la moto*  
*il carlino è seduto su una poltrona*  
*la tartaruga vince la corsa*  
*la formica lavora in estate*  
*la zanzara è fastidiosa e rumorosa*  
*il pescatore e la sua barca*  
*questa stufa fa calore*  
*i pantaloni di jeans sono carini*  
*queste posate sono pericolose*  
*gli stivali verdi sono fuori moda*

- Frasi complesse con gruppi ortografici (livello 4):

*la scimmia si diverte mangiando*  
*il pesce spada è molto pericoloso*  
*il ghiro ghiotto cerca cibo*

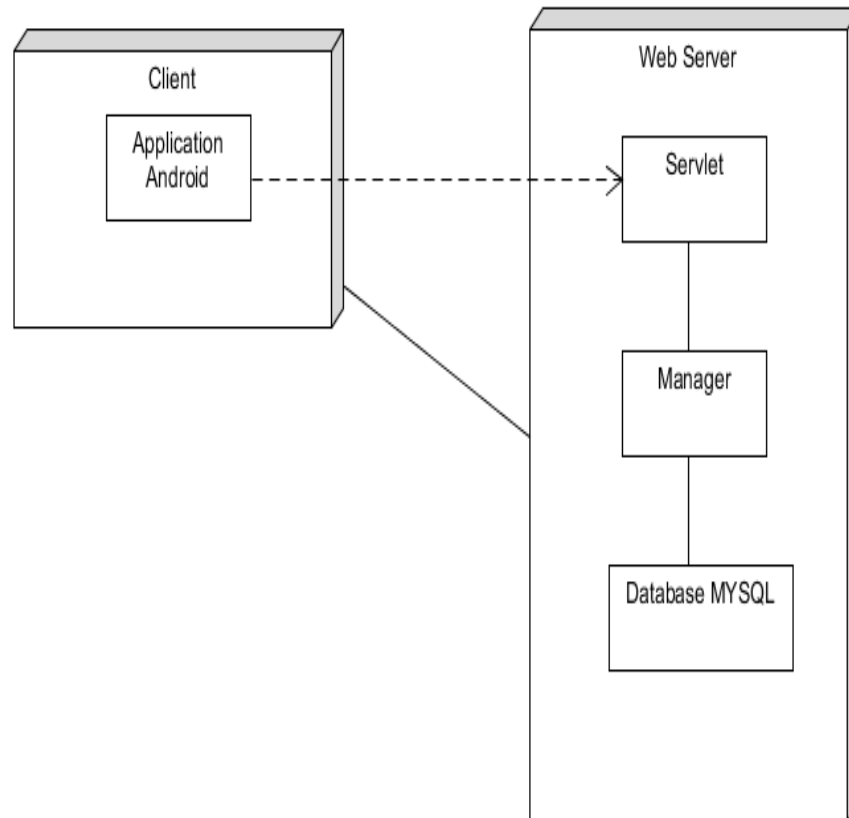
*i coniglietti hanno il pelo bianco  
i pinguini resistono al freddo  
gli scoiattoli sono in letargo  
ogni medaglia ha il suo rovescio  
il falegname lavora il legno  
la nonna ha cucinato le lasagne  
il gigante era cattivo  
ho trovato conchiglie sulla spiaggia  
passami la bottiglia di aranciata  
il ghio vivo sottoterra  
batti il ferro quando è caldo  
chi di spada ferisce di spada perisce  
chi trova un amico trova un tesoro  
il cappello era fatto di paglia  
la mamma di luca cuoce la zuppa  
il contadino lavora in montagna  
una palla rossa e gialla salta  
Chiara corre e balla sulla sabbia  
mi piace andare a scuola*

Il giocatore per superare un livello deve superare tutte le frasi presenti in ciascun livello. Per rendere il gioco più difficile le frasi vengono caricate in modo random. Il gioco essenzialmente consiste nel caricare tutte le parole presenti all'interno della frase lungo un percorso. Il giocatore lungo questo percorso si troverà davanti a tanti bivi; per ogni parola della frase gli verranno mostrati anche due distrattori che avranno come obiettivo di indurre il giocatore all'errore. La parola giusta da selezionare ad ogni bivio viene mostrata in alto. Ovviamente anche l'ordine delle parole con i relativi due distrattori è generato casualmente ogni volta che viene caricata una frase. Nel caso in cui un giocatore non riesca a comporre correttamente una frase, non viene ricaricata la stessa frase; viene generalmente scelta una frase a caso tra quelle non ancora completate.

## Implementazione seconda versione

Come detto nei precedenti paragrafi, durante lo sviluppo della seconda versione del videogioco è stata cambiata la tecnologia del server; siamo passati dall'utilizzo di PHP all'utilizzo delle Servlet.

Il seguente Deployment Diagram descrive la nuova struttura.



Durante lo sviluppo di questa seconda versione, è stata sviluppata un'area amministratore che permetta di gestire dinamicamente le frasi all'interno del videogioco, che permetta di visualizzare gli errori commessi dai vari giocatori e che permetta di visualizzare l'elenco dei giocatori iscritti al videogioco.

Di seguito mostrerò come è cambiata l'implementazione delle varie funzionalità del videogioco rispetto alla versione precedente; prenderò come esempio lo stesso utilizzato nel precedente paragrafo per evidenziare le differenze.

```

public void loginGiocatore(string username, string password)
{
    StartCoroutine(Login(username, password));
}

IEnumerator Login(string username, string password)
{
    WWWForm form = new WWWForm();
    form.AddField("username", username);
    form.AddField("password", password);
    UnityWebRequest www = UnityWebRequest.Post("http://193.205.161.130/GiocoDislessia/Login", form);
    yield return www.SendWebRequest();
    if (www.downloadHandler.text == "0")
    {
        Debug.Log("Login fallito");
        loginFallito = true;
    }
    else if (www.downloadHandler.text == "-1")
    {
        Debug.Log("Connessione non riuscita");
    }
    else
    {
        Debug.Log("Login effettuato");
        loginFallito = false;
        Debug.Log("" + www.downloadHandler.text);
        giocatore = new Giocatore(username, password, int.Parse(www.downloadHandler.text));
        giocatoreOn = true;
        getFrase(giocatore.getProgresso());
    }
}

```

Come è possibile vedere nella foto precedente, la richiesta alla Servlet rispetto alla pagina PHP avviene nello stesso modo poiché entrambe hanno le stesse funzionalità. Tutto rimane invariato mentre cambia solamente l'URL a cui mandare la richiesta.

Nella foto seguente viene mostrato come la Servlet reagisce alla richiesta di un client. Da notare che, per via dell'utilizzo dell'architettura MVC, la Servlet si occuperà solamente di:

- recuperare i dati della richiesta;
- richiamare il metodo all'interno del Manager;
- stampare i risultati in base all'esecuzione del metodo richiamato.



```

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    PrintWriter writer = response.getWriter();
    String username=request.getParameter("username");
    String password=request.getParameter("password");
    Giocatore giocatore=new Giocatore(username,password);
    try {
        int result=ManagerGiocatore.trovaGiocatore(giocatore);
        writer.write(result + "");
    } catch (SQLException e) {
        writer.write("-1");
        e.printStackTrace();
    }
}
}

```

Nella foto seguente viene mostrato il funzionamento del metodo richiamato dalla Servlet. In generale si occuperà di:

- richiedere un'istanza del ConnectionPool;
- eseguire la query sul database;
- elaborare i dati restituiti dall'esecuzione della query;
- rilasciare l'istanza di Connection Pool;
- restituire i risultati alla Servlet.

```

public synchronized static int trovaGiocatore(Giocatore giocatore) throws SQLException
{
    int result;
    Connection connection = null;
    PreparedStatement preparedStatement = null;
    String SQL = "SELECT progresso FROM Giocatore WHERE username='" + giocatore.getUsername() +
        "' AND password='" + giocatore.getPassword() + "'";
    try {
        connection = ConnectionPool.getConnection();
        preparedStatement = connection.prepareStatement(SQL);
        ResultSet rs=preparedStatement.executeQuery(SQL);
        if(rs.next())
        {
            result=rs.getInt(1);
        }
        else
        {
            result=0;
        }
        connection.commit();
    } finally {
        try {
            if (preparedStatement != null)
                preparedStatement.close();
        } finally {
            ConnectionPool.releaseConnection(connection);
        }
    }
    return result;
}

```

Da notare che, per effettuare la connessione al database, è stata creata una classe a parte che si occupa esclusivamente di gestire le varie richieste di connessione. La classe ConnectionPool si occupa di:

- prendere un'istanza di connessione dal Pool e restituirla ad un utente quando richiesta;
- inserire un'istanza di connessione nel Pool quando un utente non ne ha più bisogno;
- creare una nuova connessione nel caso in cui il Pool sia vuoto;

Questa classe permette di migliorare le prestazioni dell'esecuzione dei comandi su un database. L'apertura e la gestione di una connessione al database per ciascun utente sono costose e sprecano risorse. Questa classe permette di posizionare la connessione nel Pool dopo che è stata creata e di riutilizzarla evitando di creare una nuova connessione. Solamente nel caso in cui sono utilizzate tutte le connessioni, ne viene creata una nuova. Questa classe permette anche all'utente di ridurre il tempo di attesa per stabilire una connessione al database.

Di seguito viene fornita l'implementazione della classe.

```
private static List<Connection> freeDbConnections;

static {
    freeDbConnections = new LinkedList<Connection>();
    try {
        Class.forName("com.mysql.cj.jdbc.Driver");
    } catch (ClassNotFoundException e) {
        System.out.println("DB driver not found:" + e.getMessage());
    }
}
```

La foto precedente descrive la creazione della lista di connessioni e definisce il driver da utilizzare per la connessione.

La foto seguente descrive il metodo che permette di creare la connessione al database.

```

private static synchronized Connection createDBConnection() throws SQLException {
    Connection newConnection = null;
    String ip = "localhost";
    String port = "3306";
    String db = "dislessiagioco?allowPublicKeyRetrieval=true&useSSL=false";
    String username = "dislessiagioco";
    String password = "dislessiagioco";
    newConnection = DriverManager.getConnection("jdbc:mysql://" + ip + ":" + port + "/" + db, username, password);
    newConnection.setAutoCommit(false);
    return newConnection;
}

```

La foto seguente descrive il metodo che permette di rimuovere una connessione dal Pool per restituirla ad un utente.

```

public static synchronized Connection getConnection() throws SQLException {
    Connection connection;
    if (!freeDbConnections.isEmpty()) {
        connection = (Connection) freeDbConnections.get(0);
        freeDbConnections.remove(0);
        try {
            if (connection.isClosed())
                connection = getConnection();
        } catch (SQLException e) {
            connection.close();
            connection = getConnection();
        }
    } else {
        connection = createDBConnection();
    }
    return connection;
}

```

Infine, la foto seguente descrive il metodo che permette di aggiungere la connessione al Pool quando viene rilasciata.

```

public static synchronized void releaseConnection(Connection connection) throws SQLException {
    if (connection != null)
        freeDbConnections.add(connection);
}

```

Le funzionalità dell'area amministratore sono state sviluppate con lo stesso principio dello sviluppo del videogioco. L'unica differenza è nell'utilizzo di pagine JSP implementate con Java rispetto all'utilizzo di Script implementati in C#.

Di seguito mostreremo solamente un esempio di come le pagine JSP mandano richieste alle Servlet.

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Login admin</title>
</head>
<body>

<h1>Effettuare l'accesso da amministratore</h1>
<form action="LoginAdmin" method="post" name="login">
<p> Username: <input type="text" name="username" onchange="controllloUsername()"> </p>
<p> Password: <input type="password" name="password" onchange="controllloPassword()"> </p>
<div id="infoDiv" class="alert alert-danger form-group d-none" role="alert"> </div>
<p> <input type="button" value="invia" name="invia" onclick="validazione()">
    <input type="reset" value="reset" name="reset">
</p>
</form>

<script src="JS/jquery.js"></script>
<script src="JS/ValidazioneLogin.js"></script>

</body>
</html>

```

## Capitolo 6: Conclusioni

Questa applicazione nasce con lo scopo di creare qualcosa di simpatico e utile per bambini con dislessia. È importante riconoscere subito chi soffre di questa patologia, perché possano percorrere attraverso misure preventive, compensative e riabilitative un percorso scolastico uguale ai propri compagni, per evitare la catena di eventi negativi che da esso spesso conseguono. Il più grande deficit che lascia un disturbo se non riconosciuto, non è il problema, perché alla fine ci convivi tranquillamente, ma è la mancanza di autostima che può derivare da un percorso scolastico scorretto.

Il videogioco offre la possibilità di allenare i bambini in modo simpatico e l'area amministratore offre la possibilità a chi lo gestisce di caricare nuove frasi, rimuovere frasi esistenti e visualizzare gli errori commessi dai bambini.

Per il futuro l'idea è di continuare questo progetto, realizzando:

- opportune modifiche sull'interfaccia generale del menu del videogioco;
- nuovi modelli 3D che possano andare a sostituire quelli attualmente utilizzati e rendere le scene;
- aggiungere ulteriori funzionalità all'area amministratore;
- sviluppare ulteriori giochini da integrare all'interno dell'applicazione DSA App.

# Bibliografia

- [1]. AID, Associazione Italiana Dislessia. "Consensus Conference, Disturbi evolutivi specifici di apprendimento." Raccomandazioni per la pratica clinica definite con il metodo della Consensus Conference, Montecatini Terme. 2006.
- [2]. Evaluation of DysWebxia: A Reading App Designed for People with Dyslexia.
- [3]. A Method to Improve the Spelling of Children with Dyslexia
- [4]. Enhancing phonological awareness in Children with Dyslexia
- [5]. Review of Android and iOS Tablet Apps in Spanish to improve reading and writing skills of children with dyslexia
- [6]. <https://www.serafico.org/servizi-al-paziente>
- [7]. <https://www.serafico.org/dsa>
- [8]. <https://www.pensareoltre.org>
- [10]. <https://www.tuttodsa.it/dislessia.html>
- [11]. [https://it.wikipedia.org/wiki/Metodo\\_Montessori](https://it.wikipedia.org/wiki/Metodo_Montessori)
- [12]. <https://www.wikihow.it/Riconoscere-i-Segni-della-Dislessia>
- [13]. [https://it.wikipedia.org/wiki/Disturbi\\_specifici\\_di\\_apprendimento](https://it.wikipedia.org/wiki/Disturbi_specifici_di_apprendimento)
- [14]. <https://www.aiditalia.org/it/la-dislessia>
- [15]. <https://www.disabili.com/scuola-a-istruzione/speciali-scuola-a-istruzione/dsa-disturbi-specifici-dell'apprendimento/dislessia-dsa-a-scuola>

# Ringraziamenti

In questo percorso di studio ho incontrato persone meravigliose che hanno contribuito nello sviluppo del mio progetto dandomi dono delle loro conoscenze. Questo progetto segna per me la conclusione di un percorso iniziato tre anni fa.

Ringrazio la mia famiglia per il supporto in questi anni.

Ringrazio gli amici ed i colleghi di università con cui ho passato tantissimi momenti insieme.

Ringrazio i professori che mi hanno insegnato tanto durante questo percorso ed in particolare ringrazio la professoressa Rita Francese che mi ha seguito durante lo sviluppo di tale progetto.

Ringrazio tutte le persone che sono stati presenti e che hanno creduto in me.

Infine, ringrazio me stesso.