

Università degli Studi di Padova  
DIPARTIMENTO DI MATEMATICA "TULLIO LEVI-CIVITA"  
CORSO DI LAUREA MAGISTRALE IN INFORMATICA



Prototipo dell'adattamento digitale di  
“The Mind’s Ear and Eye Training”

*Tesi di laurea magistrale*

*Relatore*

Prof.ssa Ombretta Gaggi

*Laureando*

Emanuele Bortolin  
1191651

---

ANNO ACCADEMICO 2021-2022

Emanuele Bortolin:  
*Prototipo dell'adattamento digitale di "The Mind's Ear and Eye Training"*,  
Tesi di laurea magistrale, © Dicembre 2022.

*“Non perdo mai, o vinco o imparo”*

— Nelson Mandela

Dedico il lavoro svolto alla mia famiglia e alle persone a me più care.  
In particolare, desidero dedicarlo ad Enrica, la bambina che mi ha fatto capire quanto i disturbi specifici dell'apprendimento rappresentino una difficoltà per lo svolgimento di attività quotidiane che riteniamo “normali” ed automatiche.



# Sommario

I disturbi specifici dell'apprendimento comportano difficoltà nell'apprendimento di abilità specifiche che acquisiamo durante l'età scolare. I disturbi sono principalmente quattro: dislessia, disortografia, disgrafia, e discalculia. Ognuno di essi coinvolge un'abilità in particolare ed una persona può presentare anche più disturbi. In generale, compromettono le attività di lettura, scrittura, e calcolo, comportando numerose difficoltà durante il percorso scolastico. Se non trattati adeguatamente, i disturbi specifici dell'apprendimento portano a numerose conseguenze psicologiche che riducono la qualità della vita del soggetto che ne soffre come, ad esempio, calo del rendimento scolastico, abbandono prematuro degli studi, stress, isolamento sociale, e perdita di autostima.

È importante notare che i bambini con disturbi specifici dell'apprendimento hanno intelligenza e capacità cognitive adeguate alla loro età. La differenza consiste nel fatto che elaborano le informazioni in modo diverso e, se la modalità didattica non asseconda le caratteristiche individuali di apprendimento, i bambini apprendono consumando maggiori energie e più tempo rispetto ai propri coetanei.

È possibile adoperarsi per aiutare i soggetti a convivere con i disturbi e, prima si interviene, maggiori sono i risultati.

Lo scopo di questa tesi è fornire uno strumento che permetta di intervenire in età scolare, indicativamente tra i 6 e i 9 anni. Lo strumento a cui ci riferiamo consiste in un'applicazione web che propone un metodo già utilizzato per questo scopo: un metodo frontale che prevede l'interazione tra i bambini e una persona che li guida nell'apprendimento della lettura e della scrittura. Ciò che ci proponiamo di sviluppare, è una versione digitale del metodo, in modo che i bambini possano esercitarsi autonomamente senza la necessità di una figura di riferimento. Oltre a questo, non deve risultare come "un compito aggiuntivo per casa" ma come un gioco, che invogli i bambini ad esercitarsi quotidianamente.

Il risultato ottenuto è solamente un prototipo. Per verificare la sua reale efficacia è necessario eseguire dei test appropriati con gruppi di bambini. Solo dopo questi test sarà possibile valutarlo propriamente e, in caso dimostri del potenziale, potrà continuare il suo sviluppo.



*“Lavora affinché il tuo idolo diventi il tuo rivale.”*

— Anonimo

# Ringraziamenti

*Innanzitutto, vorrei esprimere la mia gratitudine alla Prof.ssa Ombretta Gaggi, relatrice della mia tesi, per l'aiuto e il sostegno fornитomi durante la stesura del lavoro.*

*Desidero ringraziare tutti coloro che in questi anni mi sono stati vicini, mi hanno supportato, e hanno creduto in me.*

*Padova, Dicembre 2022*

Emanuele Bortolin



# Indice

<b>1</b>	<b>Introduzione</b>	<b>1</b>
1.1	Disturbi Specifici dell’Apprendimento (DSA) . . . . .	1
1.1.1	The Mind’s Ear and Eye Training . . . . .	5
1.2	Obiettivo . . . . .	6
<b>2</b>	<b>Stato dell’arte</b>	<b>7</b>
2.1	Eye games . . . . .	7
2.2	Percivals Help For Dyslexia . . . . .	10
2.3	LogopedicSciSce . . . . .	11
2.4	Dyslexia.ai . . . . .	13
<b>3</b>	<b>Serious Game e Gamification</b>	<b>17</b>
3.1	Serious Game . . . . .	17
3.2	Gamification . . . . .	18
3.3	Classificazione di Librania . . . . .	21
3.4	Elementi utilizzati nel prototipo . . . . .	21
<b>4</b>	<b>Librania</b>	<b>23</b>
4.1	Scopo . . . . .	23
4.2	Tecnologie utilizzate . . . . .	23
4.2.1	Flutter . . . . .	24
4.2.2	Firebase . . . . .	27
4.3	Struttura . . . . .	31
4.4	Componenti . . . . .	33
4.4.1	FirebaseBridge . . . . .	33
4.4.2	Word e WordFamily . . . . .	33
4.4.3	Level . . . . .	37
4.4.4	WordManager . . . . .	38
4.4.5	Child . . . . .	39
4.4.6	Progress . . . . .	40
4.4.7	ExerciseManager . . . . .	44
4.4.8	TrainingManager . . . . .	47
4.5	Applicazione Web . . . . .	48
4.5.1	Login . . . . .	49
4.5.2	Tutorial . . . . .	50
4.5.3	Home . . . . .	51
4.5.4	Istruzioni . . . . .	53
4.5.5	Allenamento . . . . .	54

4.5.6	Community . . . . .	67
4.5.7	Profilo . . . . .	71
4.5.8	Impostazioni . . . . .	75
4.6	Elementi di gamification e per la piacevolezza della web app . . . . .	76
4.7	Problematiche . . . . .	78
<b>5</b>	<b>Test</b>	<b>81</b>
5.1	Test funzionali e di usabilità . . . . .	81
5.2	Test di efficacia . . . . .	90
<b>6</b>	<b>Conclusioni e sviluppi futuri</b>	<b>91</b>
6.1	Conclusioni . . . . .	91
6.2	Sviluppi futuri . . . . .	92
6.3	Potenzialità e possibile evoluzione dell'applicazione . . . . .	96
<b>A</b>	<b>Parole utilizzate e informazioni correlate</b>	<b>99</b>
<b>B</b>	<b>Informazioni di input per il database</b>	<b>105</b>
B.1	Utenti . . . . .	105
B.2	Parole . . . . .	106
B.3	Famiglie . . . . .	114
B.4	Livelli . . . . .	117
	<b>Bibliografia</b>	<b>121</b>

# Elenco delle figure

1.1	Dislessia . . . . .	2
1.2	Disortografia . . . . .	2
1.3	Disgrafia . . . . .	2
1.4	Discalculia . . . . .	2
1.5	Alunni con DSA in percentuale del totale alunni - serie storica: aa.ss. 2010/2011 - 2017/2018 . . . . .	3
1.6	Alunni con DSA per tipologia di disturbo - confronto aa.ss. 2013/14 - 2015/16 - 2017/18 . . . . .	4
2.1	Alcuni livelli da 3 a 4 anni . . . . .	8
2.2	Alcuni livelli da 4 a 5 anni . . . . .	8
2.3	Alcuni livelli da 5 a 6 anni . . . . .	9
2.4	Alcuni livelli da 6 anni in su . . . . .	9
2.5	Pagina iniziale . . . . .	10
2.6	Lista parola dalla “m” alla “o” . . . . .	10
2.7	Lettura parola . . . . .	10
2.8	Frase per ricordare la parola (1) . . . . .	10
2.9	Frase per ricordare la parola (2) . . . . .	11
2.10	Scrittura con risultato . . . . .	11
2.11	Svolgimento esercizi “MENU SCISCE” . . . . .	11
2.12	Svolgimento esercizi “COPPIE MINIME” . . . . .	12
2.13	Svolgimento esercizi “Gla Gle Gli Glo Glu” . . . . .	12
2.14	Home page di Dyslexia.ai . . . . .	13
2.15	Schermata “Spelling” . . . . .	13
2.16	Schermata “Reading” . . . . .	13
2.17	Fase lettura . . . . .	14
2.18	Fase scrittura . . . . .	14
2.19	Risultati . . . . .	14
2.20	Lettura della parola da parte dell’utente . . . . .	14
2.21	Completare la parola con le lettere mancanti . . . . .	15
3.1	Gioco da tavolo Monopoly . . . . .	18
3.2	Piramide di Maslow . . . . .	20
3.3	Piramide di Maslow (informatica) . . . . .	20
4.1	Logo di Flutter . . . . .	24
4.2	Architettura di Flutter . . . . .	25
4.3	Logo di Firebase . . . . .	27

4.4	Esempio pagina di autenticazione . . . . .	28
4.5	Esempio pagina del Firestore Database . . . . .	29
4.6	Esempio di regole del Firestore Database . . . . .	30
4.7	Esempio di utilizzo Firestore Database . . . . .	30
4.8	Esempio di Storage. I quattro file visualizzati sono contenuti nella cartella <code>disegno</code> , corrispondente al percorso <code>/words/disegno/</code> . . . . .	30
4.9	Esempio delle regole dello Storage . . . . .	31
4.10	Ogni dispositivo esegue l'applicazione sviluppata con Flutter ed è connesso a Firebase . . . . .	32
4.11	Struttura dell'applicazione con componenti principali . . . . .	32
4.12	Divisione in sillabe di una parola con temporizzazione . . . . .	34
4.13	Memorizzazione di una parola in Firebase . . . . .	34
4.14	Memorizzazione di una famiglia di parole in Firebase . . . . .	35
4.15	Trasferimento file da dispositivo locale (a sinistra) allo Storage di Firebase (a destra) . . . . .	36
4.16	Memorizzazione dei livelli in Firebase . . . . .	38
4.17	Memorizzazione di un utente in Firebase . . . . .	40
4.18	Memorizzazione delle informazioni riguardanti le parole, per ciascun utente in Firebase . . . . .	41
4.19	Memorizzazione delle informazioni riguardanti i livelli, per ciascun utente in Firebase . . . . .	42
4.20	Memorizzazione delle informazioni riguardanti le statistiche in Firebase . . . . .	43
4.21	Workflow di ExerciseManager . . . . .	46
4.22	Pulsante senza descrizione . . . . .	48
4.23	Pulsante con descrizione . . . . .	48
4.24	Pulsante “Sì / No” . . . . .	48
4.25	Pagina di login . . . . .	49
4.26	Esempio di tutorial con 5 pagine . . . . .	50
4.27	Sezioni a cui conducono i pulsanti . . . . .	51
4.28	Home page di Librania . . . . .	52
4.29	Home page di Librania (applicazione Android) . . . . .	52
4.30	Conferma di uscita (applicazione Android) . . . . .	52
4.31	Schermata con una parte delle istruzioni che spiegano come svolgere gli esercizi . . . . .	53
4.32	Pagina in cui si sceglie il livello da giocare . . . . .	54
4.33	Pagina in cui si sceglie il livello da giocare con messaggio di reminder . . . . .	54
4.34	Tutorial della pagina della scelta del livello . . . . .	55
4.35	Messaggio livello bloccato . . . . .	56
4.36	Solo il primo livello è giocabile . . . . .	56
4.37	Primo livello “terminato”, secondo livello “completato”, terzo livello “sbloccato”, quarto e quinto livello “bloccati” . . . . .	57
4.38	Lettura parola . . . . .	58
4.39	Lettura parola in iOS . . . . .	58
4.40	Richiesta significato parola . . . . .	59
4.41	Spiegazione significato parola . . . . .	59
4.42	Selezione colore per test accessibilità . . . . .	60
4.43	Fase di spelling . . . . .	61
4.44	Lettura delle sillabe . . . . .	62
4.45	Fase della fotografia . . . . .	62
4.46	Fase di scrittura . . . . .	63

4.47 Parola scritta correttamente . . . . .	64
4.48 Parola sbagliata al primo tentativo . . . . .	64
4.49 Parola sbagliata al primo tentativo . . . . .	65
4.50 Parole nuove . . . . .	66
4.51 Risultati livello . . . . .	66
4.52 Pagina community . . . . .	67
4.53 In ordine da sinistra: classifica livelli, sfide, e sfide settimanali . . . . .	69
4.54 Selezione colore per test accessibilità delle classifiche . . . . .	70
4.55 Pagina profilo . . . . .	71
4.56 Pagina medaglie . . . . .	72
4.57 Descrizione medaglia . . . . .	72
4.58 Pagina livelli . . . . .	73
4.59 Progressi livello . . . . .	73
4.60 Pagina parole . . . . .	74
4.61 Pagina delle impostazioni . . . . .	76
4.62 La mascotte Tobi . . . . .	76
4.63 Layout iniziale delle parole . . . . .	79
 5.1 Logo Windows . . . . .	82
5.2 Schermate della pagina “Medaglie” con dimensioni diverse della finestra	83
5.3 Schermate della pagina dei livelli con dimensioni diverse della finestra	83
5.4 Schermate della fase di lettura della parola dell’allenamento con dimensioni diverse della finestra . . . . .	83
5.5 Logo MacOS . . . . .	84
5.6 Touch bar di un MacBook Pro . . . . .	84
5.7 Schermata scelta del livello con messaggio di una nuova medaglia in Safari	85
5.8 Schermata risultati allenamento con messaggio di una nuova medaglia in Safari . . . . .	85
5.9 Logo Ubuntu . . . . .	85
5.10 Tutorial iniziale . . . . .	86
5.11 Tutorial iniziale (finestra ridotta) . . . . .	86
5.12 Medaglia di benvenuto (finestra ridotta) . . . . .	86
5.13 Logo Android . . . . .	87
5.14 Android 5.0 versione nativa . . . . .	87
5.15 Android 5.0 versione web . . . . .	87
5.16 Android 12 versione nativa . . . . .	88
5.17 Android 12 versione web . . . . .	88
5.18 Logo iOS . . . . .	88
5.19 Schermate web app in browser Safari in dispositivo iPhone . . . . .	89
 6.1 Esempio dimostrativo dell’utilizzo di un font diverso . . . . .	96

# Elenco delle tabelle

4.1 Piattaforme supportate dal framework Flutter . . . . .	25
A.1 Lista parole . . . . .	99

# Capitolo 1

## Introduzione

La tesi prevede lo sviluppo di un'applicazione web che aiuti i bambini con disturbi specifici dell'apprendimento ad imparare a leggere e a scrivere.

### 1.1 Disturbi Specifici dell'Apprendimento (DSA)

I disturbi specifici dell'apprendimento (abbreviati DSA) sono disturbi neurologici che rendono difficile leggere, scrivere, ed elaborare i numeri. Quando leggiamo e scriviamo, il nostro cervello esegue più processi, tra i quali:

- riconoscere la forma delle lettere;
- associare un suono a ciascun fonema;
- associare un significato ad una parola;
- gestire il movimento del braccio e della mano per scrivere.

I disturbi specifici dell'apprendimento causano difficoltà in uno o più di questi processi. In base alle difficoltà che emergono, i disturbi sono classificati in:

- **dislessia:** rende difficile leggere. In figura 1.1 è riportato un esempio di come una persona con dislessia vede una frase scritta: nella realtà le parole sono ordinate in righe, ma la persona le vede confuse e sovrapposte;
- **disortografia:** ostacola l'associazione tra una lettera o fonema al suono corrispondente e porta a commettere numerosi errori nella scrittura. In figura 1.2 sono riportate delle parole scritte da un bambino con disortografia, come si può notare gli errori sono numerosi;
- **disgrafia:** è un disturbo motorio. La figura 1.3 è la foto di alcune parole scritte da un bambino con disgrafia. I tratti sono imprecisi e di conseguenza la grafia risulta quindi disordinata, difficilmente leggibile e poco chiara;
- **discalculia:** ha a che fare con i numeri e le operazioni. In figura 1.4 è riportato l'esercizio svolto da un bambino con discalculia, nel quale è evidente la difficoltà di approssiarsi al mondo dei numeri.

Vi ricordate la storia dei tre porcellini?  
 E' facile, c'era il lupo cattivo.  
 Se non ve la ricordate, potete tornare a  
 rileggerla!

---

Figura 1.1: Dislessia

IL CANTO DEI PIRATI  
I VECCHI PIRATI, DALLI OCCHI  
BENDATI CON LA MANO AD,  
UN CINNO NON FANO PIÙ PAURA  
VIVONO E DORMONO NEI LIBRI  
DI AVENTURA.

Figura 1.2: Disortografia

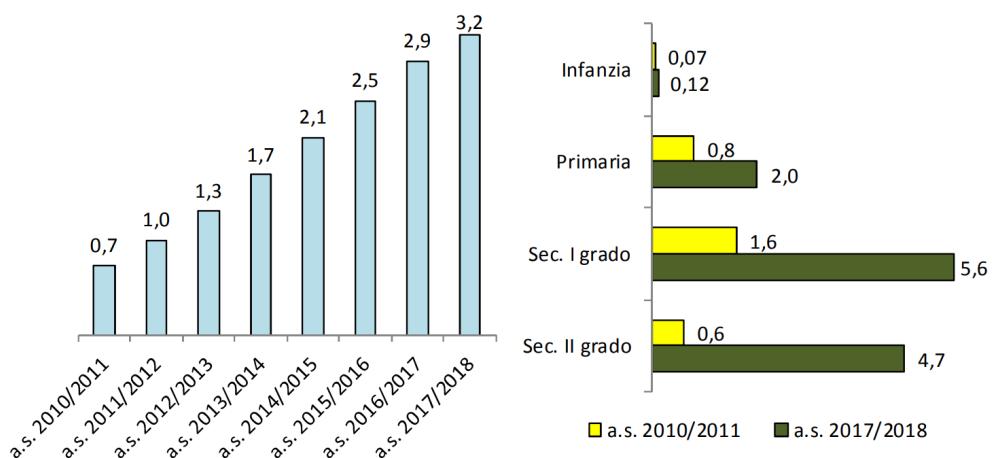
Ho creato un bel po'  
 di libri  
 Ho letto un po' delle  
 storie del libro  
 Ho scritto molto  
 per la scuola

Figura 1.3: Disgrafia

$$\begin{array}{r}
 & 2 \\
 & 3 \\
 & 9 \\
 + & 5 \\
 \hline
 7 & 1 & 8
 \end{array}$$

Figura 1.4: Discalculia

Queste difficoltà non sono indice di una minore intelligenza, ma è un modo diverso di elaborare le informazioni. I bambini che ne soffrono, si rendono conto di essere più lenti rispetto ai loro compagni. Infatti, gli insegnanti assegnano loro compiti facilitati o in quantità minore. Rendendosene conto, si sentono frustrati, tendono ad isolarsi, l'autostima diminuisce, e il rendimento scolastico cala. Se non vengono prese misure adeguate a compensare tali difficoltà, esse potrebbero avere come conseguenza, soprattutto in età adulta, disturbi mentali e maggiori difficoltà nella vita sociale. Riportiamo qualche dato raccolto dal MIUR (Ministero dell'Istruzione, dell'Università, e della Ricerca).



**Figura 1.5:** Alunni con DSA in percentuale del totale alunni - serie storica: aa.ss. 2010/2011 - 2017/2018

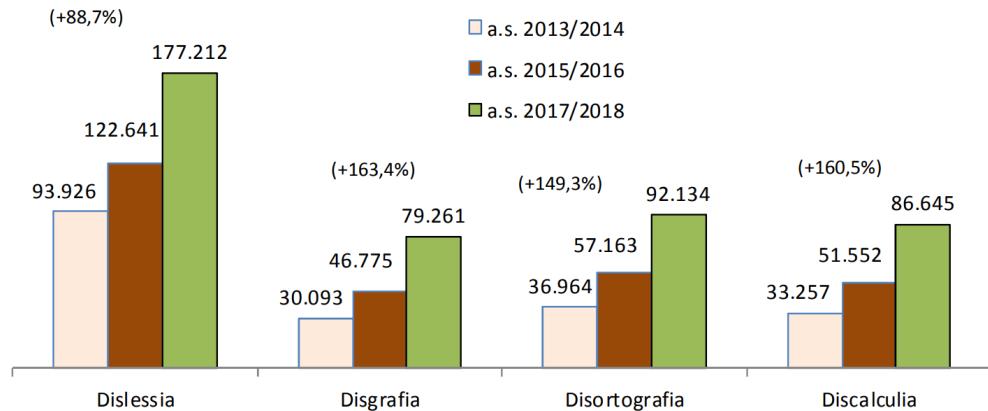
Il grafico in figura 1.5 riporta le percentuali dei bambini con disturbi specifici dell'apprendimento. Nell'istogramma a sinistra è riportata, per ogni anno scolastico dal 2010/2011 al 2017/2018, la percentuale di studenti con almeno un disturbo specifico dell'apprendimento sul totale degli studenti frequentanti un qualsiasi ordine scolastico. A destra, invece, le percentuali sono suddivise per ordine di scuola e sono presi in considerazione solo anni scolastici 2010/2011 (in giallo) e 2017/2018 (in verde). La scuola secondaria di primo grado risulta avere una percentuale molto più alta rispetto alla scuola primaria, ma non vuol dire che c'è un numero maggiore di bambini con disturbi. L'aumento della percentuale deriva dal fatto che i test per diagnosticare i disturbi specifici dell'apprendimento vengono effettuati dopo la scuola primaria: non tutti i bambini presenti in quel 5,6% erano rappresentati anche dal 2,0% della scuola primaria. L'ideale sarebbe diagnosticare i disturbi il prima possibile, perché è più semplice agire per compensare le difficoltà che comportano. Il periodo più appropriato per agire è durante i primi anni della scuola primaria, per questo, le diagnosi effettuate durante la scuola secondaria sono definite "tardive".

Come si può notare, il numero di soggetti che presentano almeno un disturbo cresce di anno in anno. Questo non vuol dire che sta aumentando il numero di bambini con disturbi, ma che gli screening stanno diventando più efficaci e riconoscono un numero maggiore di bambini con difficoltà. Ciò significa che la percentuale indicata è un limite inferiore.

Si può notare dunque che il numero di bambini con disturbi specifici dell'apprendimento non è affatto trascurabile: statisticamente, ogni classe contiene almeno un bambino

che presenta un disturbo specifico dell'apprendimento.

Nel grafico in figura 1.6, ogni barra rappresenta la somma degli alunni che presentano un disturbo. Gli alunni considerati appartengono alla scuola primaria, alla secondaria di primo grado, e alla secondaria di secondo grado. Nel grafico gli studenti non sono divisi per scuola frequentata, in questo modo si ha un quadro complessivo di quanto i disturbi siano presenti nella fascia più giovane della nostra società.



**Figura 1.6:** Alunni con DSA per tipologia di disturbo - confronto aa.ss. 2013/14 - 2015/16 - 2017/18

L'aumento annuale del numero di studenti con disturbi deriva dal fatto che, di anno in anno, la sensibilizzazione verso i disturbi specifici dell'apprendimento cresce e le diagnosi eseguite sono sempre maggiori.

Il disturbo più comune è la dislessia, ed è proprio quello su cui si focalizza questa tesi.

### 1.1.1 The Mind's Ear and Eye Training

“The Mind’s Ear and Eye Training” è un metodo di apprendimento implicito studiato dalla professoressa Barbara Arfè per la lingua italiana.

Esistono due tipi di apprendimento:

- **esplicito:** avviene tramite l'insegnamento di regole e istruzioni che devono essere applicate per portare a termine un obiettivo;
- **implicito:** avviene tramite esempi e genera “esperienza”.

“The Mind’s Ear and Eye Training” è un metodo di apprendimento implicito perché non insegna quali sono le “regole” o le “istruzioni” per leggere e scrivere una parola generica, ma insegna al bambino come leggere e scrivere determinate parole, in modo che il suo cervello astragga le regole e crei autonomamente una procedura per raggiungere l’obiettivo. Ci si riferisce a quest’ultimo concetto col termine “generalizzazione”, vale a dire, in questo caso, riuscire a leggere e a scrivere correttamente parole diverse da quelle usate durante l’insegnamento. Infatti, l'apprendimento implicito comporta risultati migliori con parole mai viste prima rispetto all'apprendimento esplicito.

Il metodo è composto da nove fasi:

1. lettura della parola, seguendo la lettura col dito;
2. viene insegnato al bambino come analizzare la parola scritta e mappare i fonemi sulle unità ortografiche da sinistra a destra;
3. ripetere la parola e chiedere al bambino di pronunciarla di nuovo ad alta voce;
4. al bambino viene chiesto di nominare le lettere della parola una per una, da sinistra a destra, mentre lo sperimentatore le indica;
5. dopo aver analizzato la struttura fonologica e ortografica della parola, il bambino chiude gli occhi e crea un’immagine mentale della parola scritta;
6. tenendo gli occhi chiusi, il bambino deve leggere la parola nella sua mente e sillabarla da sinistra a destra;
7. scrittura della parola;
8. il bambino confronta la parola scritta con quella di riferimento;
9. se dal confronto le due parole risultano diverse, ripetere la procedura.

## 1.2 Obiettivo

L’obiettivo della tesi è sviluppare uno strumento in grado di dare un aiuto ai bambini con un disturbo specifico dell’apprendimento nell’imparare a leggere e a scrivere. Esistono a questo scopo metodi frontali, con “carta e penna”, utilizzati da psicologi e logopedisti che danno risultati provati e riescono, da caso a caso, a compensare le difficoltà (uno di questi è proprio “The Mind’s Ear and Eye Training”). Sarebbe interessante capire se è possibile ottenere risultati anche con uno strumento digitale, tramite PC o smartphone, e questa tesi serve proprio a sviluppare uno strumento di questo tipo.

Come scritto nella sezione precedente, chi soffre di uno di questi disturbi ha difficoltà nella lettura e nella scrittura, risultando più lento rispetto ai propri compagni. Questo è dovuto alla necessità di un maggior impegno, che comporta un maggior utilizzo di energie e porta i bambini a stancarsi più velocemente.

Il target di utenza è proprio i bambini dei primi anni di scuola primaria, durante i quali imparano a leggere e a scrivere. L’applicazione può essere un supporto sia per chi ha un disturbo, ma anche per chi non ne ha. Inoltre, dovrebbe risultare piacevole all’utilizzo: non dovrebbe essere vista come un’applicazione “scolastica”, ma bensì come un gioco nel quale cimentarsi e sfidare i propri compagni.

L’applicazione sviluppata è un prototipo, che verrà testato ed eventualmente adattato per poi essere utilizzato.

Essa contiene una versione “digitalizzata” di “The Mind’s Ear and Eye Training”, che consiste nei seguenti passi:

1. lettura della parola per intero da parte dell’applicazione;
2. lettura della parola per lettere da parte dell’applicazione, aumentando la spazatura tra esse;
3. lettura della parola per sillabe da parte dell’applicazione, evidenziando la sillaba letta;
4. creazione di un’immagine mentale della parola (“fotografia” della parola);
5. scrittura della parola tramite tastiera;
6. controllo correttezza della parola da parte dell’applicazione;
7. se la parola è stata scritta in modo errato ripetere dal punto 1 (al massimo una volta).

Il training prevede blocchi da sei parole, quindi la procedura sopra riportata verrà ripetuta consecutivamente per sei parole diverse.

Per raggiungere il maggior numero di utenti è stato scelto di sviluppare il prodotto come applicazione web, utilizzabile dunque da un browser, presente in qualsiasi dispositivo. L’applicazione è raggiungibile tramite URL ed è quindi utilizzabile sia tramite dispositivi Windows, che Android, iOS, e Mac. È disponibile anche l’apk dell’applicazione, in modo che possa essere inserita nel Play Store di Google. Per motivi burocratici, invece, al momento non è disponibile su Apple Store.

# Capitolo 2

## Stato dell'arte

*In questa sezione vedremo alcune applicazioni che già si rivolgono a bambini con disturbi specifici dell'apprendimento nell'ambito dell'educazione.*

Esistono numerose applicazioni che promettono di migliorare i processi cognitivi degli utenti e parte di esse sono specifiche per la dislessia.

Alcune di queste applicazioni sono pensate per essere dei giochi che cercano di aiutare i bambini a migliorare le abilità con le quali si trovano in difficoltà.

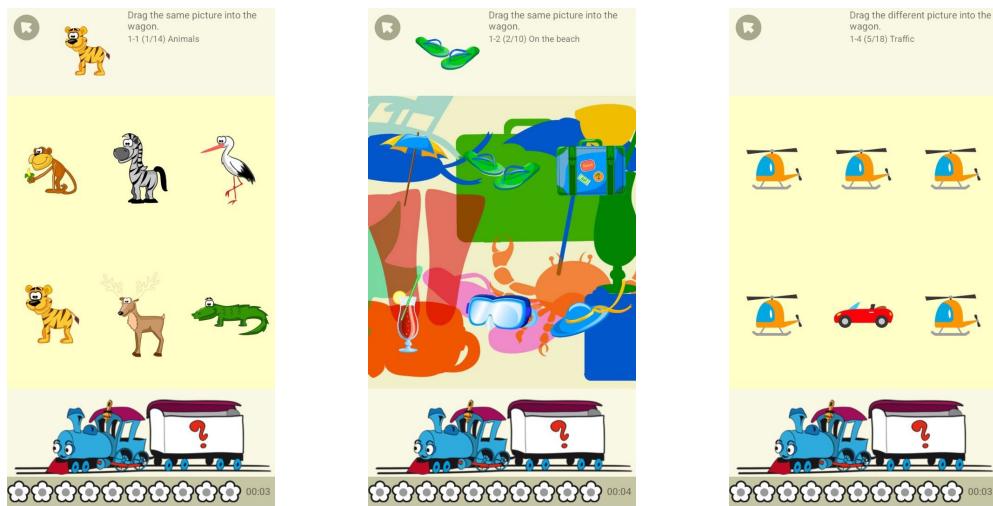
### 2.1 Eye games

Eye games è un'applicazione Android che ha come scopo esercitare i bambini nel riconoscere le forme. Gli esercizi sono suddivisi per età e, in base a questa, vengono sottoposte ai bambini forme diverse, cominciando da disegni di animali ed oggetti fino ad arrivare alle lettere. La lingua dell'applicazione è l'inglese, ma se ai bambini è chiaro l'obiettivo dei minigiochi, la lingua non risulta limitante.

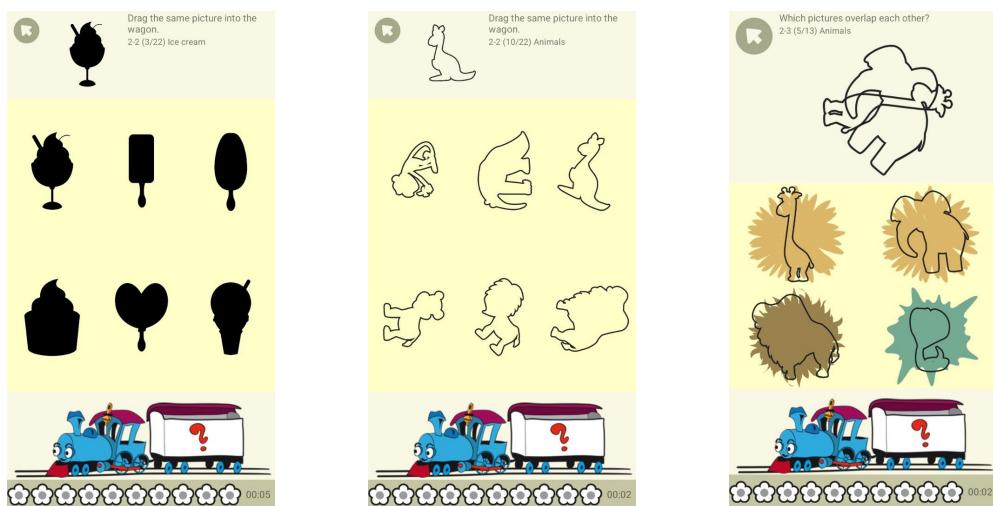
L'applicazione presenta esercizi diversi a seconda della fascia d'età:

- 3-4 anni: le forme da riconoscere sono dei disegni, che possono essere colorati o solo delle sagome nere. In figura 2.1 sono riportate le schermate di alcuni esercizi. La prima immagine a sinistra mostra uno degli esercizi nei quali si chiede di trovare, tra sei immagini, quella uguale ad un'immagine di riferimento. Nell'immagine centrale viene aggiunta una difficoltà: lo sfondo non è monocromatico ma contiene a sua volta dei disegni. Nell'immagine a destra, l'esercizio chiede ai bambini di trovare l'immagine diversa rispetto ad altre cinque, tutte uguali, presenti sullo schermo;
- 4-5 anni: come si può vedere in figura 2.2, questa sezione ha esercizi simili al punto precedente. La differenza principale consiste in un esercizio nel quale i disegni sono composti solo dai contorni della sagoma. In quest'ultimo vengono presentate due sagome sovrapposte e i bambini devono distinguere le due sagome e selezionarle tra le sei presenti sotto;
- 5-6 anni: gli esercizi sono simili ai precedenti, ma viene aggiunta come difficoltà che i disegni e le sagome da riconoscere hanno gradi di rotazione differenti. Si possono vedere alcuni di questi esercizi in figura 2.3;

- da 6 anni in su: da come si può vedere in figura 2.4, le forme da riconoscere sono lettere. I bambini devono toccare tutte le lettere presenti sullo schermo che sono uguali a quella di riferimento. Possono essere lettere singole, a coppie, o in trio. Come negli altri esercizi, lo sfondo varia, da monocromatico a sfondo con texture. Anche qui, come difficoltà aggiuntiva, alle lettere può essere applicata una rotazione. Alcuni esercizi presentano un’ulteriore difficoltà: le lettere sono visibili a intermittenza in intervalli di circa mezzo secondo. I bambini quindi hanno circa mezzo secondo di tempo per riconoscere se la forma delle lettere corrisponde alle lettere di riferimento.



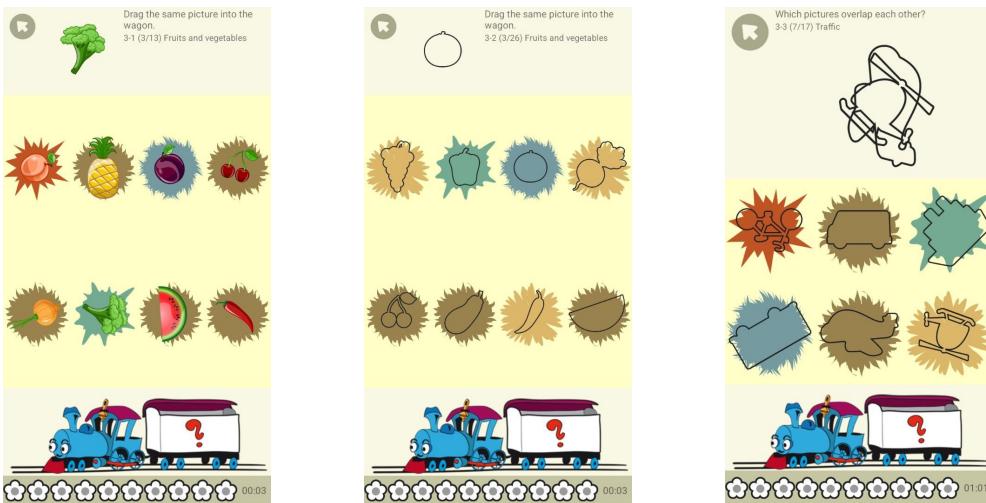
**Figura 2.1:** Alcuni livelli da 3 a 4 anni



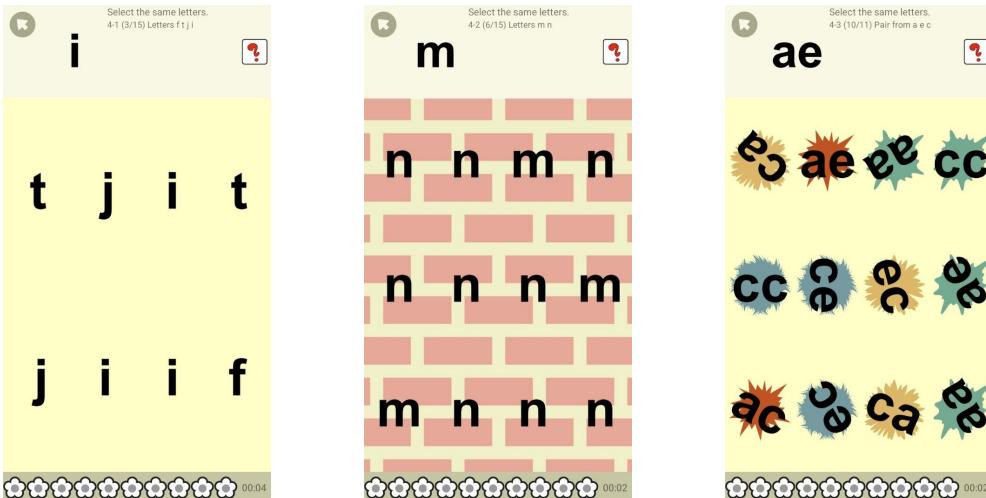
**Figura 2.2:** Alcuni livelli da 4 a 5 anni

## 2.1. EYE GAMES

9



**Figura 2.3:** Alcuni livelli da 5 a 6 anni



**Figura 2.4:** Alcuni livelli da 6 anni in su

Come si può vedere nelle immagini, nella parte inferiore sono presenti dei fiorellini. Essi corrispondono alle prove che i bambini devono affrontare per terminare il livello. Lo scopo di questa applicazione è diverso da quello del nostro prototipo: Eye games ha lo scopo di insegnare a riconoscere le forme, mentre il nostro prototipo quello di riconoscere le parole. In Eye games le parole non sono contemplate, infatti la lingua inglese dell'applicazione non ostacola i bambini nel raggiungimento degli obiettivi se viene spiegato loro da un adulto come superare i livelli.

Anche il target è diverso. Eye games punta ad una fascia di età molto giovane, per la quale lavorare con le parole potrebbe essere prematuro.

## 2.2 Percivals Help For Dyslexia

Percivals Help For Dyslexia, la cui pagina iniziale è raffigurata in figura 2.5, è un'applicazione inglese pensata specificamente per bambini con dislessia. Si concentra su 107 parole di uso comune e utilizza delle tecniche divertenti per aiutarne il memorizzazione. All'utente vengono presentate delle liste di parole (figura 2.6), ordinate alfabeticamente, dalla “a” alla “o”. Le altre parole non sono disponibili nella versione gratuita dell'applicazione. L'utente sceglie quale parola vuole imparare e, una volta selezionata, inizia l'allenamento. Il contenuto dell'allenamento varia leggermente da parola a parola, ma la struttura rimane la stessa:

1. lettura della parola e lettura di una frase contenente la parola scelta (figura 2.7);
2. lettura di un'ulteriore frase contenente la parola. Questa lettura, a differenza della prima, è accompagnata da animazioni, enfasi diverse, acronimi della parola stessa, o filastrocche che riconducono al significato della parola. Questo è fatto in modo, a mio parere, che i bambini ricordino la parola più facilmente (figure 2.8 e 2.9);
3. scrittura della parola. I bambini hanno un solo tentativo per scriverla correttamente (figura 2.10).

Terminato l'allenamento, l'utente viene riportato alla lista di parole per scegliere la successiva.

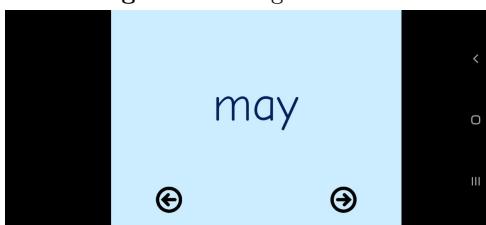
Alcune fasi dell'allenamento possono essere riconducibili a quelle del nostro prototipo, ma in questa applicazione la lingua inglese è indubbiamente un ostacolo.



**Figura 2.5:** Pagina iniziale



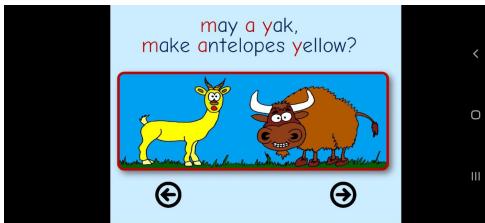
**Figura 2.6:** Lista parola dalla “m” alla “o”



**Figura 2.7:** Lettura parola



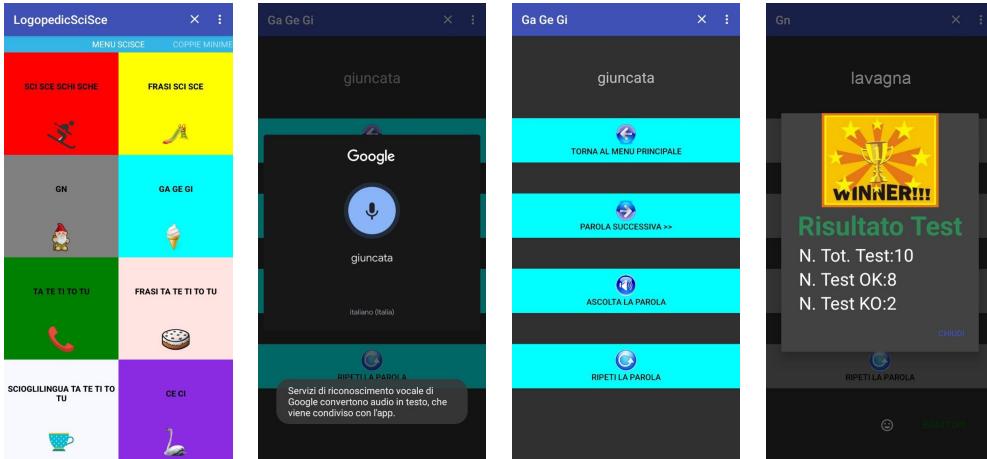
**Figura 2.8:** Frase per ricordare la parola (1)

**Figura 2.9:** Frase per ricordare la parola (2)**Figura 2.10:** Scrittura con risultato

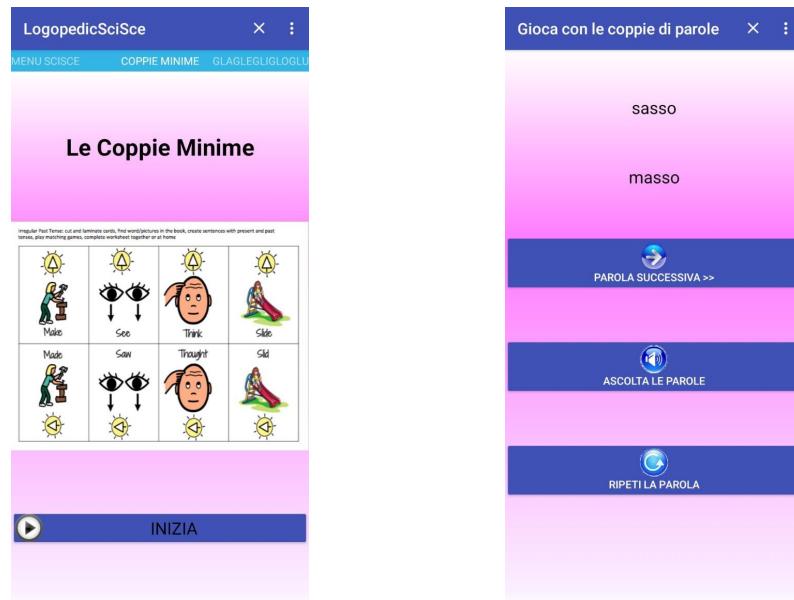
### 2.3 LogopedicSciSce

LogopedicSciSce è un'applicazione in italiano con diversi tipi di allenamento. Contrariamente da quanto lascia intuire il titolo, le parole ivi presenti non contengono solo “sci” e “sce”, ma anche altri gruppi ortografici, come ad esempio “gn” e “ce-ci”. Lo scopo di questa applicazione è insegnare a leggere ed è divisa in tre sezioni.

Nella prima sezione (“MENU SCISCE”), scegliendo uno degli argomenti presenti nella schermata principale (prima immagine a sinistra della figura 2.11) si aprirà una schermata nella quale i bambini dovranno leggere la parola, senza però che venga fornita loro una dimostrazione. Dopo aver tentato di leggerla è possibile ascoltarne la lettura, ma anche riprovare a leggerla, e passare alla parola successiva. Il test prevede 10 parole e alla fine dei tentativi mostra il risultato dell'esercizio.

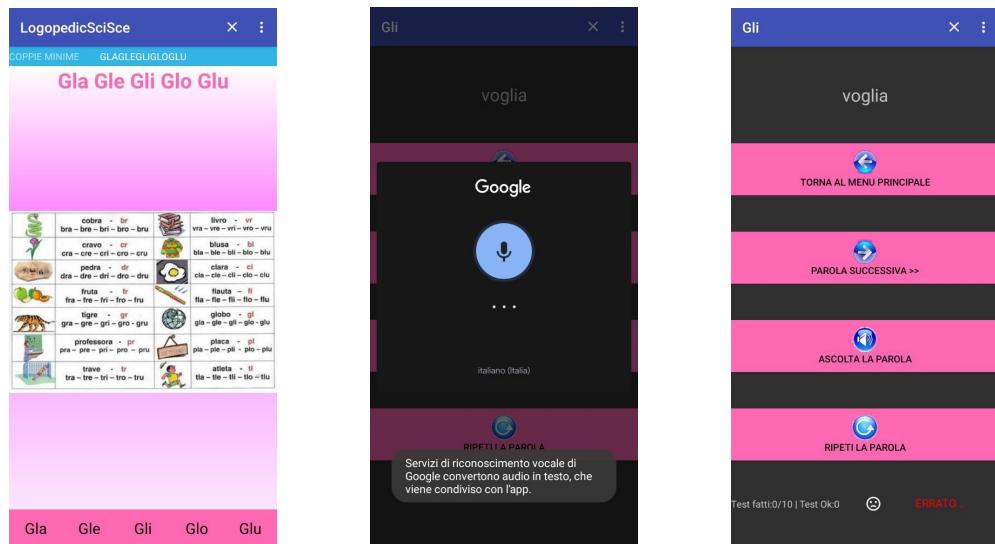
**Figura 2.11:** Svolgimento esercizi “MENU SCISCE”

La seconda sezione dell'applicazione (“COPPIE MINIME”) mostra ai bambini delle coppie di parole con suoni simili. Per ogni coppia, gli utenti possono ascoltare la lettura delle parole o procedere con la coppia successiva (figura 2.12). Questo non è un test come il precedente, ma fa solo ascoltare ai bambini le letture delle parole, cercando di far cogliere loro la differenza tra parole con suono simile.



**Figura 2.12:** Svolgimento esercizi “COPPIE MINIME”

La terza sezione (“Gla Gle Gli Glo Glu”), in figura 2.13, completa la prima con i gruppi ortografici “gla”, “gle”, “gli”, “glo”, e “glu”. Per accedere ai test è necessario selezionare il gruppo ortografico desiderato nel footer della pagina.



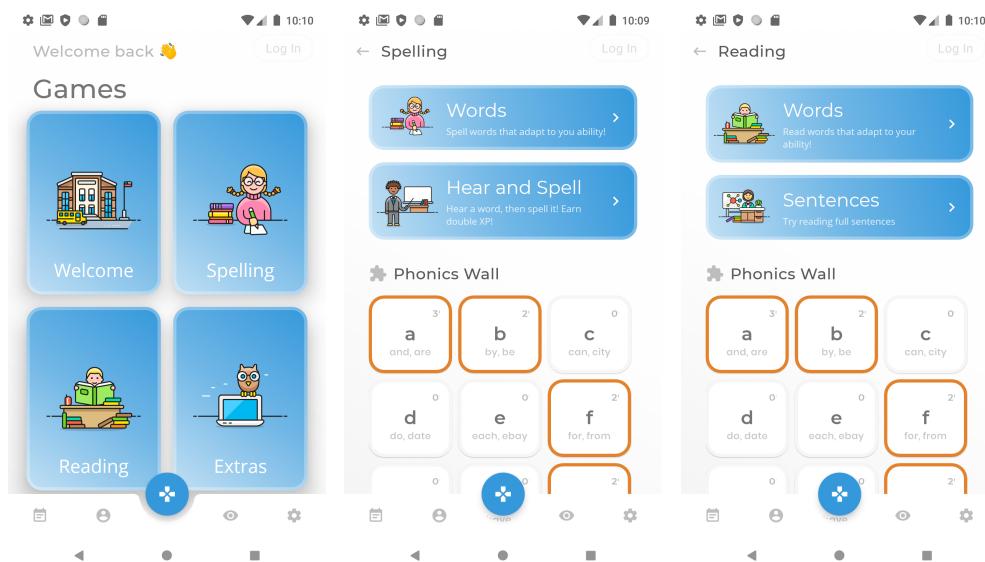
**Figura 2.13:** Svolgimento esercizi “Gla Gle Gli Glo Glu”

L’approccio all’allenamento è diverso rispetto a quello del nostro prototipo.

## 2.4 Dyslexia.ai

Dyslexia.ai è un'applicazione in inglese pensata per migliorare le abilità di lettura e di scrittura. Le funzionalità presenti sono numerose, consideriamo solo quelle più vicine all'obiettivo della tesi.

L'utente può decidere, nella home page (figura 2.14), se allenarsi nella lettura o nella scrittura. In entrambi i casi, l'applicazione presenta una pagina con gli esercizi che può svolgere e il “muro dei fonemi”, vale a dire una lista di tutti i fonemi presenti nella lingua inglese. Nel muro dei fonemi, i fonemi con cui l'utente si è esercitato sono contraddistinti da un bordo arancione.



**Figura 2.14:** Home page di Dyslexia.ai

**Figura 2.15:** Schermata “Spelling”

**Figura 2.16:** Schermata “Reading”

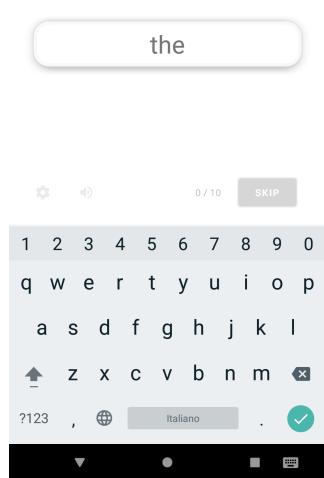
Dalla home page, selezionando il riquadro “Spelling” (figura 2.15), si apre la pagina per esercitarsi nella scrittura. Qui sono disponibili due esercizi e il muro dei fonemi:

- **Words:** vengono presentate consecutivevamente 10 parole. Ogni parola appare sullo schermo per circa 2 secondi (figura 2.17), dopodiché scompare e l'utente deve cercare di scriverla correttamente (figura 2.18). Ad ogni parola esatta guadagna dei punti esperienza. Alla prima parola corretta i punti guadagnati sono 2 e ad ogni parola successiva scritta correttamente il punteggio incrementa finché non viene scritta un parola in modo errato. Al termine dell'esercizio vengono mostrati i risultati: l'elenco delle parole distinguendo tra quelle scritte in modo corretto e quelle scritte in modo errato. Un esempio è riportato in figura 2.19;

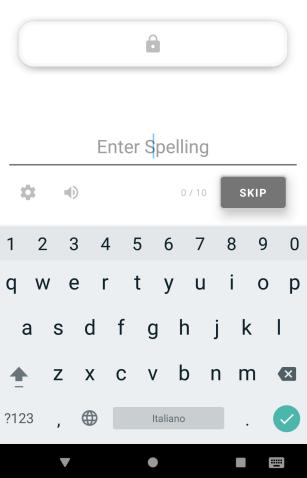
Spelling Words  
X

Spelling Words  
X

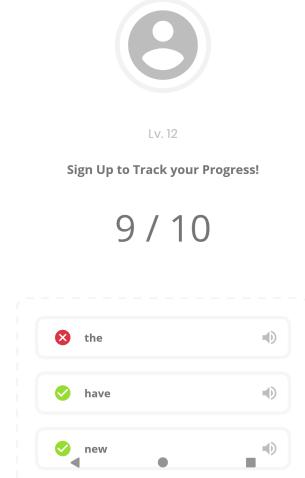
Spelling Words  
X



**Figura 2.17:** Fase lettura



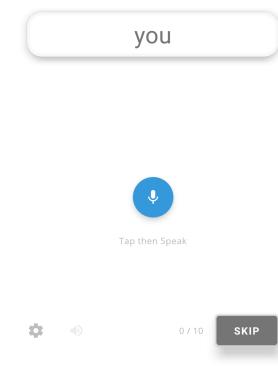
**Figura 2.18:** Fase scrittura



**Figura 2.19:** Risultati

- **Hear and Spell:** simile all'esercizio precedente, con la differenza che la parola non viene mostrata ma ne viene riprodotta la lettura (figura 2.20). In questo caso quindi l'utente non ha un input visivo ma bensì uditivo.

Reading Words  
X



**Figura 2.20:** Lettura della parola da parte dell'utente

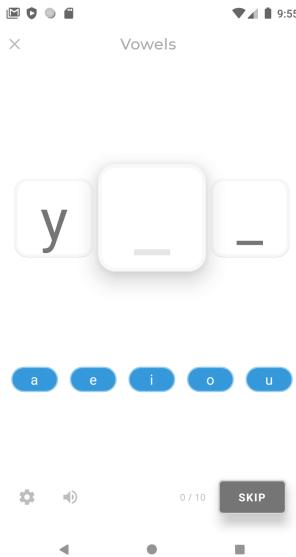
Oltre gli esercizi, abbiamo detto che è presente anche il muro dei fonemi. Non solo vengono indicati quali sono i fonemi con cui ci si è già esercitati, ma permette anche di esercitarsi con un fonema in particolare. Cliccando su uno di essi, infatti, viene riproposto l'esercizio "Words" che presenta però solo parole col fonema selezionato.

Selezionando il riquadro "Reading" dalla home page (figura 2.16) ci si può invece esercitare nella lettura. Anche qui sono presenti due esercizi e il muro dei fonemi.

- **Words:** uguale come struttura all'esercizio per la scrittura, in questo caso le parole appaiono sullo schermo e l'utente, attivando il microfono del dispositivo, deve leggerle. Se la parola riconosciuta dall'intelligenza artificiale che analizza l'audio in ingresso dal microfono corrisponde a quella apparsa sullo schermo, il test viene considerato corretto;
- **Sentences:** simile al primo esercizio, non appare una sola parola ma un'intera frase che l'utente deve leggere.

Anche qui, oltre gli esercizi, è presente il "muro dei fonemi". Premendo sul riquadro di un fonema, l'applicazione ripropone l'esercizio "Words" nel quale però appariranno solo parole con il fonema selezionato.

Altro esercizio presente in Dyslexia.ai è "Vowels", raggiungibile dal riquadro "Extras". In questo esercizio vengono presentate all'utente delle parole alle quali mancano le vocali. Lo scopo dell'esercizio è inserire le lettere mancanti, come si può vedere nella schermata riportata in figura 2.21.



**Figura 2.21:** Completare la parola con le lettere mancanti

L'applicazione risulta per certi aspetti simile a quella pensata per questa tesi. Difficilmente però potrà essere utilizzata, in quanto pensata per bambini di madre lingua inglese. Potrebbe invece essere utile, a mio parere, per esercitarsi nella lingua inglese.



# Capitolo 3

## Serious Game e Gamification

*Spieghiamo in questo capitolo la classificazione “serious game” e “gamification”, la differenza tra i due, e come si classifica Librania.*

### 3.1 Serious Game

I serious game sono dei giochi, o videogiochi, progettati a scopo educativo. Nonostante si presentino come dei giochi, l'intrattenimento non è il loro fine, ma lo è invece la formazione.

Una definizione di *serious game*, tratta da *Serious Games* di Clark Abt [6], è:

*“Games may be played seriously or casually. We are concerned with serious games in the sense that these games have an explicit and carefully thought-out educational purpose and are not intended to be played primarily for amusement. This does not mean that serious games are not, or should not be, entertaining”*

Ciò che contraddistingue un serious game quindi è l'obiettivo di istruire i giocatori e l'intrattenimento non è componente essenziale del gioco.

Come descritto in *Serious Games* [6], da sempre l'uomo cerca di accrescere la propria esperienza per prepararsi a diverse circostanze. Per farlo, simula la realtà tramite dei modelli in scala ridotta e più semplice da comprendere. Così facendo si possono capire quali sono le regole che governano la circostanza reale, in modo poi da essere in grado di applicarle correttamente nella realtà.

Si pensa che i primi veri e propri serious game siano nati per questioni belliche, ma gli ambiti in cui questi sono applicati sono i più diversi: commerciali, formazione aziendale, attività promozionali, campagne di sensibilizzazione, e formazione educativa.

Il primo serious game riconosciuto, *Kriegsspiel*, risale agli inizi del XVIII secolo, un gioco di simulazione di guerra dell'esercito prussiano.

Un classico esempio è invece *Monopoly* (figura 3.1), un gioco da tavolo il cui nome deriva ma dal termine “monopolio”. Lo scopo del gioco è il dominio del mercato da parte di un singolo venditore. Si tratta quindi di un serious game in ambito economico, che insegna a gestire le proprie risorse nel modo più proficuo.



**Figura 3.1:** Gioco da tavolo Monopoly

### 3.2 Gamification

La gamification è un insieme di tecniche e regole, derivanti dal mondo dei videogiochi, che vengono applicate in contesti non ludici.

Approcci di questo tipo sono studiati ormai da anni e continuano ad attirare interesse. La loro efficacia sembra ormai essere comprovata, tant'è che gli ambiti in cui vengono utilizzati sono dei generi più diversi.

Cerchiamo di dare una definizione al termine “gamification” basandoci sulla letteratura attualmente disponibile.

Secondo Kai Huotari e Juho Hamari, in *Defining Gamification - A Service Marketing Perspective* [4], col termine “gamification” si intende:

*“A process of enhancing a service with affordances for gameful experiences in order to support user’s overall value creation.”*

È dunque un processo che migliora un servizio introducendo caratteristiche per esperienze di gioco, al fine di incrementarne il valore per l’utente.

Una definizione più semplice è di Sebastian Deterding, et al. in *Gamification: Toward a Definition* [5]:

*“Gamification is the use of game design elements in non-game contexts.”*

Si può indicare come “gamification” l’utilizzo di elementi di progettazione di un gioco all’interno di contesti non ludici.

Altra definizione, proveniente dal sito [Growth Engineering](#) [7]:

*“Gamification is about taking something that is not a game and applying game mechanics to increase user engagement, happiness and loyalty!”*

Ogni definizione presa in considerazione indica l'inserimento di elementi pensati per giochi e videogiochi in contesti che non lo sono. Il motivo per il quale questi elementi funzionano "al di fuori del contesto" risiede nel fatto che migliorano l'esperienza dell'utente.

Affinché non perdano la loro efficacia e che addirittura risultino pesanti e dannosi per il prodotto o servizio offerto, rivelandosi degli ostacoli per l'utente nel raggiungimento dell'obiettivo, devono essere pensati e adattati consapevolmente, in modo che si integrino senza diventare essi stessi il cuore del prodotto o servizio.

Prova della loro popolarità, le tecniche di gamification sono utilizzate in diversi ambiti, come ad esempio:

- formazione;
- finanza;
- educazione;
- processi aziendali;
- politiche pubbliche;
- ricerca scientifica;
- salute e benessere.

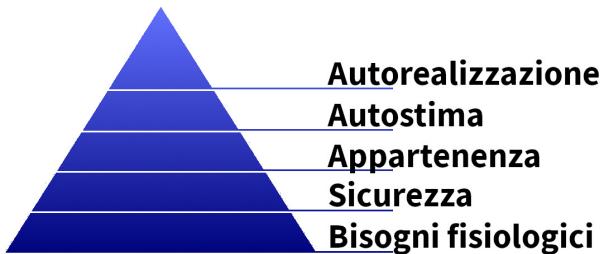
Il loro scopo è far sì che l'utente percepisca piacevole un task che altrimenti potrebbe risultare poco attraente o che potrebbe addirittura abbandonare per scarso interesse. Il modo più efficace è far leva sui desideri e i bisogni delle persone con:

- obiettivi da raggiungere;
- ricompense da guadagnare;
- competizione;
- riconoscimento di uno status.

Si tratta quindi di racchiudere le informazioni che si vogliono trasmettere all'utente o il task da svolgere in un contesto più ludico. Questo non vuol dire banalizzare il contenuto, ma renderlo solo più appetibile.

Consideriamo la piramide di Maslow. La piramide di Maslow è una teoria psicologica proposta da Abraham Maslow [6] contenente una gerarchia dei bisogni umani. Alla base della piramide si trovano i bisogni necessari alla sopravvivenza del genere umano, i livelli superiori contengono i bisogni che rendono migliore l'esperienza della vita. Ogni livello è una base necessaria per il livello superiore. Come possiamo vedere nell'immagine 3.2, alla base della piramide si trovano elementi necessari e primitivi, senza i quali la vita non esisterebbe. Il secondo livello contiene comunque bisogni molto importanti, ma meno fondamentali dei precedenti. Salendo la piramide, gli elementi di ogni livello sono sempre meno fondamentali rispetto a quelli dei livelli sottostanti, ma permettono di vivere un'esistenza migliore.

La piramide di Maslow non è esclusiva al mondo della psicologia, ma può essere adattata a numerosi settori, formando una gerarchia alla base della quale si pongono gli elementi fondamentali e necessari, e nei livelli superiori quelli che permettono di



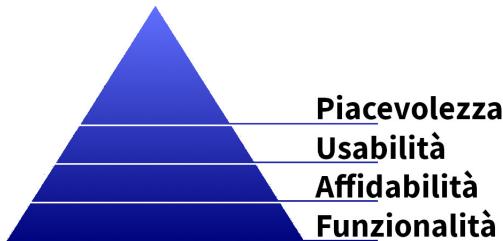
**Figura 3.2:** Piramide di Maslow

dare una qualità maggiore al contesto nel quale sono applicati.

La piramide di Maslow può essere rimappata nell'informatica, o nel nostro caso, per essere più specifici, sui bisogni degli utenti (figura 3.3). I primi tre livelli della piramide, partendo dal basso, contengono caratteristiche riguardo il funzionamento del prodotto, che deve:

- far raggiungere gli obiettivi designati all'utente;
- essere utilizzabile e funzionante ogni volta che l'utente ne richiede l'utilizzo, oltre ad essere sicuro;
- avere un'interazione semplice e intuitiva.

La punta della piramide comprende tutto ciò che rende piacevole l'utilizzo del prodotto. La gamification si trova in questo livello, perché il suo obiettivo consiste esattamente nel rendere l'esperienza dell'utente migliore. Affinché la gamification abbia effetto, è necessario che i livelli sottostanti siano tutti presenti: se il prodotto non funziona, non è disponibile quando l'utente vuole usufruirne, o è difficile da utilizzare, la gamification non avrà alcun effetto in quanto l'utente abbandonerà l'applicazione per il suo mancato funzionamento di base.



**Figura 3.3:** Piramide di Maslow (informatica)

### 3.3 Classificazione di Librania

Librania è un'applicazione a confine tra serious game e applicazione web a cui è applicata la gamification. Questo perché, nata per essere principalmente una web app che applica il “The Mind’s Ear and Eye Training”, la gamification aveva lo scopo di renderla più attraente agli utenti. Ma per essere percepita ancora più gradevole ai bambini, è stato aggiunto un semplice contesto e una grafica che possa risultare più attraente ai nostri piccoli utenti. Per questi motivi, l'applicazione web si avvicina ad un serious game.

### 3.4 Elementi utilizzati nel prototipo

La gamification contestualizza elementi e regole tipiche dei videogiochi in ambienti non ludici. I videogiochi hanno obiettivi da raggiungere, sfide da superare, e oggetti da conquistare. Spesso gli utenti sono classificati in base ai loro risultati e hanno la possibilità di mostrare il proprio status, sfoggiare i loro premi, e mettersi in competizione con altri. Esistono diversi modi, più o meno esplicativi a seconda del contesto del prodotto, in cui questi elementi possono essere inseriti.

Gli elementi utilizzati nella tesi sono:

- **livelli:** la web app è strutturata in livelli. Le parole con cui i bambini si possono esercitare sono divise in livelli ed ogni livello è giocabile solo dopo aver terminato i precedenti. In questo modo viene premiato l'impegno, in quanto dopo aver superato un livello, il successivo viene sbloccato e l'utente può proseguire;
- **medaglie / badge:** si ottengono generalmente dopo aver raggiunto determinati obiettivi, come superare livelli, vincere sfide, ottenere un certo punteggio. Rappresentano la bravura e le capacità di un utente;
- **status:** appartenenza ad una cerchia ristretta di utenti. Un esempio molto semplice riguarda servizi o prodotti con account a pagamento, dove si distingue tra “utenti standard” e “utenti premium”. Appartenere ad una certa categoria di utenti, non solo può fornire un’esperienza migliore ma può essere motivo di orgoglio. A seconda dello status, gli utenti possono godere di più o meno privilegi, come ad esempio un maggiore spazio di archiviazione in un servizio cloud, strumenti migliori in un videogioco, o maggiore disponibilità di contenuti in un servizio streaming. Addirittura, soprattutto nei videogiochi, l’obiettivo di alcuni utenti può consistere proprio nel raggiungere un determinato status, perché è motivo d’orgoglio e vanto nei confronti degli altri giocatori;
- **competizione:** una sfida competitiva è un’esperienza di gioco nella quale due o più giocatori misurano le proprie capacità al fine di riconoscere chi tra essi è il migliore. Fa leva sulla necessità che l’essere umano ha fin dall’antichità di competere con altri soggetti per garantirsi la sopravvivenza.  
Per completezza, anche se non adottato nella tesi, esiste un altro tipo di sfida: la sfida collaborativa, nella quale più utenti collaborano per raggiungere lo stesso obiettivo.



# Capitolo 4

## Librania

*Descriviamo in questo capitolo la progettazione e il funzionamento dell'applicazione web oggetto della tesi.*

### 4.1 Scopo

Lo scopo della tesi è l'implementazione digitale di “The Mind’s Ear and Eye Training”: un metodo che aiuta ad insegnare a leggere e a scrivere, tramite insegnamento implicito, ai bambini con DSA.

Con “implementazione digitale” si intende lo sviluppo di un sito web, di un programma, o di un’applicazione, che permetta ai bambini di seguire “The Mind’s Ear and Eye Training” senza che vi sia una figura umana di riferimento.

Il prototipo realizzato dovrà essere usufruibile dal maggior numero di utenti possibile. Per questo motivo, è stato deciso di svilupparlo come applicazione web, in modo che possa essere eseguito tramite browser sulla maggior parte dei dispositivi (desktop, laptop, tablet, e smartphone) senza necessità di installazione. Tuttavia, come vedremo in seguito, il framework utilizzato permette di compilare il prototipo come codice direttamente eseguibile su alcuni sistemi operativi, permettendo così agli utenti, se lo desiderano, di installare l’applicazione invece che accedervi tramite browser.

### 4.2 Tecnologie utilizzate

Vediamo in questa sezione quali tecnologie sono state utilizzate per lo sviluppo e il mantenimento del prototipo.

Il progetto non presenta particolari requisiti, infatti le tecnologie impiegate non sono molte. Inseriamo in questa sezione solo le tecnologie principali, tralasciando gli strumenti per il versionamento e per l’editing di immagini e audio, accennando il loro utilizzo solo se necessario nelle sezioni successive.

L’applicazione è stata sviluppata con il linguaggio Dart, utilizzando il framework Flutter. La scelta è ricaduta su Flutter perché permette di sviluppare applicazioni per cellulari ma anche web, accessibili da browser come classici siti web. Esso permette di compilare il codice in eseguibili specifici per diversi sistemi operativi: è in grado quindi

di compilare applicazioni Android e iOS, dando la possibilità di inserirle negli appositi store.

Uno dei punti forti di Flutter è proprio quello di confezionare prodotti per diversi sistemi operativi partendo dallo stesso codice. La scelta di tecnologie diverse, specifiche per ogni sistema operativo, non sarebbe vantaggiosa per il nostro scopo. Possibili alternative a Flutter potevano essere Cordova e Ionic, ma la scelta è ricaduta su Flutter per la curiosità di testare una tecnologia nata solo nel 2017.

Scegliere un framework come Flutter, potrebbe comportare delle complicazioni nell'utilizzo di funzionalità native del dispositivo utilizzato dall'utente, come ad esempio la fotocamera e i sensori, dato che non tutti i dispositivi ne sono provvisti. Questo progetto tuttavia non richiede tali funzionalità, quindi la scelta di Flutter risulta opportuna, nonostante in alcune occasioni sia necessario distinguere tra sistemi operativi o dispositivi diversi.

Tutto ciò che concerne autenticazione, memorizzazione cloud, e deploy, sfrutta invece la tecnologia di Firebase. Appartenenti entrambi all'azienda Google, Flutter è predisposto per l'interfacciamento a Firebase. La scelta è dipesa sia da quest'ultima affermazione, sia dalla grande quantità di strumenti che mette a disposizione per lo sviluppo di progetti.

#### 4.2.1 Flutter

Flutter (figura 4.1) è un framework open-source sviluppato da Google per la creazione di applicazioni Web, Android, iOS, Windows, Linux, e macOS. Nato nel 2017, la prima versione stabile risale al 2018. Si tratta dunque di una tecnologia recente e con una community in continua crescita.



**Figura 4.1:** Logo di Flutter

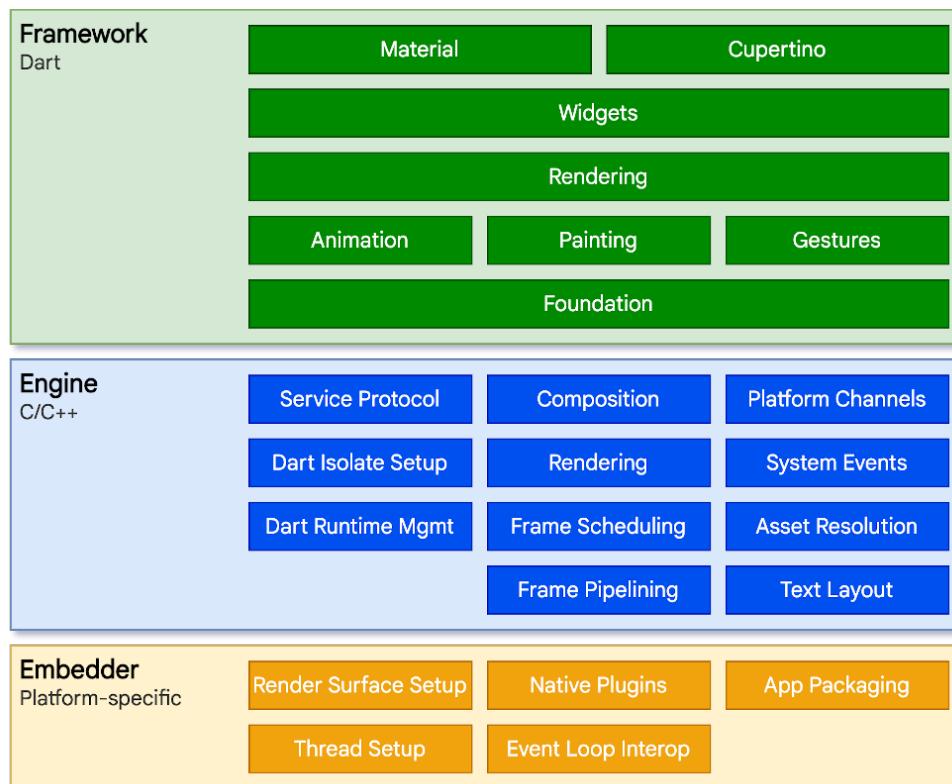
Prendendo a riferimento la classificazione descritta in *An empirical analysis of energy consumption of cross-platform frameworks for mobile development* di Ombretta Gaggi e Matteo Ciman [1], Flutter appartiene ai framework cross-compiled. Questo significa che un'applicazione Flutter, scritta in linguaggio Dart, viene compilata per ogni piattaforma nel linguaggio macchina specifico. La lista delle piattaforme supportate è riportata nella tabella 4.1.

Per quanto riguarda il web, non è propriamente corretto usare il termine “compilare”, sarebbe più corretto dire che il framework “traduce” il linguaggio Dart in HTML, CSS, e Javascript.

Flutter è stato progettato come un sistema estensibile e a layer (figura 4.2). Ogni layer interagisce con i suoi adiacenti tramite interfaccia e nessuno ha accesso privilegiato ai layer sottostanti.

Per il sistema operativo, le applicazioni Flutter sono identiche ad applicazioni native.

Piattaforma	Versione
Android	API 16 (Android 4.1) +
iOS	iOS 9 +
Linux	Debian, 64-bit
macOS	El Capitan (10.11) +
Web (Chrome)	84 +
Web (Firefox)	72.0 +
Web (Safari)	El Capitan +
Web (Edge)	1.2.0 +
Windows	7 +

**Tabella 4.1:** Piattaforme supportate dal framework Flutter**Figura 4.2:** Architettura di Flutter

Riportiamo di seguito una breve descrizione dei layer che compongono un'applicazione Flutter.

### **Embedder**

Il layer “embedder” è il punto di incontro tra l'applicazione e il sistema operativo. Esso è scritto nel linguaggio specifico della piattaforma che ospita l'applicazione:

- Java e C++ per Android;
- Objective-C / Objective-C++ per iOS e macOS;
- C++ per Windows e Linux.

### **Engine**

Il layer “engine” è il cuore di Flutter ed è scritto principalmente in C++. Esso è responsabile della costruzione delle schermate visibili agli utenti e contiene le API principali di Flutter:

- grafica;
- rendering del testo;
- I/O di file e di rete;
- supporto all'accessibilità;
- architettura dei plugin;
- toolchain per la compilazione e il runtime.

### **Framework**

Il layer “framework” è quello col quale interagiscono principalmente gli sviluppatori. Esso è scritto in Dart e contiene il codice vero e proprio dell'applicazione Flutter. Si interfaccia al layer “engine” tramite la libreria `dart:ui`, che raggruppa le funzionalità del layer “engine”, scritte in C++, in classi Dart utilizzabili dallo sviluppatore.

Ogni oggetto renderizzato è una sottoclasse di `Widget`. Possono essere personalizzati per far sì che siano visualizzati nello stesso modo in ogni sistema operativo. Le classi `Material` e `Cupertino` permettono invece di renderizzare gli oggetti primitivi, come pulsanti e caselle di testo, nel design standard del sistema operativo, rispettivamente Android e iOS.

Nonostante la sua giovane età, Flutter si avvale di una base di utenza molto ampia, che si confronta in numerosi forum. Alcuni di essi, ai quali partecipano anche gli sviluppatori Google che hanno contribuito alla nascita di questo framework, fanno da riferimento.

La sua popolarità è dovuta sia alla facilità con la quale l'utente può creare applicazioni per diversi dispositivi, sia per i numerosi plugin a disposizione, sempre forniti dai membri della community. I plugin sono molto semplici da installare ed utilizzare, e la maggior parte di essi è fornita di documentazione ed esempi.

Altro punto di forza è la disponibilità di un'ampia documentazione:

- guida passo passo per installare Flutter ed integrarlo con degli editor (Visual Studio Code, IntelliJ, Android Studio, ed Emacs);
- guide per compilare, eseguire, e testare l'applicazione su diverse piattaforme;
- documentazione approfondita sul linguaggio Dart;
- documentazione sui Widget presenti nel framework con esempi interattivi;
- guide che agevolano la comprensione del linguaggio per sviluppatori che derivano da sviluppo web, Android, iOS;
- documentazione e guide per l'integrazione con Firebase;
- documentazione per l'integrazione di plugin;
- e molto altro.

Peculiarità di Flutter è l'“hot reload”. Esso è uno strumento che agevola la fase di scrittura del codice e di debugging, permettendo di modificare velocemente l'interfaccia e il codice. La sua particolarità consiste nel fatto che è in grado di aggiornare l'applicazione senza perdere il suo stato.

L'hot reload funziona inserendo i file di codice aggiornati all'interno della macchina virtuale che ospita l'applicazione in esecuzione, la quale aggiorna le classi modificate. Dopodiché, il framework ricostruisce l'albero di Widget (gli oggetti visualizzati nell'interfaccia) consentendo di visualizzare gli effetti delle modifiche.

Un esempio che può chiarire l'affermazione appena riportata è il seguente.

*Testando una pagina dell'applicazione contenente dei dati inseriti “a mano” o derivanti da una procedura svolta durante il test, lo sviluppatore nota un errore nell'interfaccia. Lo sviluppatore può aggiornare il codice dell'interfaccia, mantenendo l'applicazione in esecuzione, ed eseguendo l'hot reload l'interfaccia dell'applicazione si aggiorna mantenendo intatti i dati già presenti.*

Questo risulta essere molto comodo, soprattutto se per raggiungere la pagina in questione è necessaria una lunga procedura.

### 4.2.2 Firebase

Firebase (figura 4.3) è una piattaforma online progettata per la creazione di applicazioni web e dispositivi mobili, comprata da Google nel 2014. Essa fornisce numerosi servizi che facilitano sviluppo, rilascio e monitoraggio, e analisi. L'ampio utilizzo di questa tecnologia è dovuto anche alla disponibilità delle sue librerie per diversi linguaggi, tra i quali Dart, ovviamente, Python, Javascript, Java, C++, permettendo il loro utilizzo in tecnologie Android, Apple, Web, Unity, e Flutter.

Ad ogni progetto Firebase è associata una chiave e l'accesso ai suoi servizi da parte delle applicazioni è possibile soltanto tramite essa. In questo modo i dati non sono accessibili a chiunque, ma solo alle applicazioni registrate con tale chiave.



**Figura 4.3:** Logo di Firebase

Di seguito ci limitiamo a citare i servizi utilizzati in questo progetto, rimandando la loro implementazione alle sezioni successive dove descriviamo come ne usufruiamo:

- **Authentication:** sistema di autenticazione. Questo servizio permette di diversificare gli utenti che richiedono di leggere e scrivere dati in Firebase. In questo modo, è possibile fornire privilegi diversi in base all'utente che accede. L'autenticazione avviene all'interno dell'applicazione e serve a tracciare e identificare le richieste dell'utente ai servizi Firebase utilizzati dall'applicazione. L'autenticazione può avvenire nelle seguenti modalità:
  - **email/password:** si può registrare un utente inserendo un'email e una password nel sistema di autenticazione;
  - **numero di telefono:** un utente può essere registrato tramite il suo numero di telefono;
  - **account di altre piattaforme:** l'accesso può avvenire tramite l'account di un'altra piattaforma, come ad esempio Google, Facebook, e GitHub, per citarne alcune;
  - **anonimo:** l'utente non utilizza credenziali per accedere a Firebase e usa un account temporaneo.

Identificatore	Provider	Data di creazione	Accesso eseguito	UID utente
manager@trainingapp.it	✉️	10 feb 2022	15 giu 2022	khw8ZPwpz9gUVZi2pell9kctFVQ2
elio.costa@trainingapp.it	✉️	22 set 2021	28 giu 2022	KuL3w50QBHWoXmqAQIL4PPPrB...
vera.rossi@trainingapp.it	✉️	22 set 2021	30 giu 2022	jC4F5kyNyOTOfeiDoupKgmXHLie2

**Figura 4.4:** Esempio pagina di autenticazione

Nella figura 4.4 sono presenti tre utenti la cui autenticazione avviene tramite email e password (unica modalità utilizzata in questo progetto).

Firebase rende disponibili delle API con le quali è possibile, dall'applicazione client, effettuare la registrazione e il login. Al momento, la registrazione di nuovi utenti avviene per opera di un amministratore che fornirà poi le credenziali all'utente, dato che il contesto di utilizzo prevede un ambiente controllato. Per quanto riguarda il login, invece, esso avviene tramite API;

- **Firestore Database:** database contenente i dati dell'applicazione. È un database cloud NoSQL suddiviso in *raccolte*, ognuna delle quali contiene una lista di *documenti*. Un documento ha a sua volta una lista di raccolte e dei campi dati associati. I campi dati possono assumere diversi tipi: stringhe, numeri, booleani, mappe, liste, timestamp, coordinate geografiche, e riferimenti ad altri documenti o raccolte del database.

In figura 4.5 si può vedere la struttura del database del progetto, che può funzionare da esempio. La raccolta `users` contiene due documenti: `elio.costa` e `vera.rossi`, due utenti dell'applicazione.

The screenshot shows the Firebase Firestore interface. On the left, there's a sidebar with a tree view of collections: 'training-dsa' (with 'challenge', 'data', 'families', 'levels', 'training\_info', and 'users' children), 'users' (with 'elio.costa' and 'vera.rossi' children), and 'words'. In the center, under the 'users' collection, there's a list of documents: 'elio.costa' and 'vera.rossi'. The 'vera.rossi' document is selected and expanded, showing its fields: 'age' (8), 'first\_time' (false), 'name' ('Vera'), 'starting\_training\_message\_date' ('30 giugno 2022 12:28:52 UTC+2'), 'surname' ('Rossi'), and 'username' ('vera.rossi'). On the right, there are two sections: '+ Avvia raccolta' (with 'known\_words', 'levels', 'training\_info', and 'tutorial') and '+ Aggiungi campo' (with a placeholder for a new field). At the top, there are navigation icons: a home icon, a back arrow, a forward arrow, and a search bar.

**Figura 4.5:** Esempio pagina del Firestore Database

Il documento `vera.rossi` ha sei campi dato: `age`, `first_time`, `name`, `starting_training_message_date`, `surname`, e `username`; e quattro raccolte: `known_words`, `levels`, `training_info`, e `tutorial`, ognuna delle quali contiene una lista di documenti.

Ogni elemento contenuto nel database è identificabile da un percorso, univoco, nel formato

/[Raccolta]/[Documento]/[Campo | Raccolta] ...

Ad esempio, il percorso che identifica l'utente `vera.rossi` è

`/users/vera.rossi`

Il seguente invece è un percorso più lungo:

`/users/vera.rossi/known_words/acquitrino/views`

e identifica il campo dati `views` del documento `acquitrino` che fa parte della raccolta `known_words` del documento `vera.rossi`.

Al database possono essere associate delle regole, che permettono di gestire i permessi di lettura e di scrittura al database in base al tipo di accesso e all'utente loggato. Qui, tramite delle funzioni, è possibile specificare il permesso di ciascun utente per accedere ad ogni documento e raccolta.

Proponiamo in figura 4.6 un esempio molto semplice, nel quale chiunque (anche se non autenticato) può leggere l'intero database, ma solo chi ha effettuato il login può compiere operazioni di scrittura. Infatti, la condizione che permette la lettura è sempre vera e quella che permette la scrittura assume valore `true` solo se la richiesta avviene da un utente autenticato.

```
service cloud.firestore {
  match /databases/{database}/documents {
    match /{document=**} {
      allow read: if true;
      allow write: if request.auth != null;
    }
  }
}
```

**Figura 4.6:** Esempio di regole del Firestore Database

Tramite un grafico, come quello in figura 4.7, è possibile monitorare il numero di accessi al database, sia in lettura che in scrittura. Questo può essere utile ai fini della fatturazione, per controllare se viene raggiunta o meno la soglia imposta dal contratto sottoscritto, ma soprattutto è un modo per sapere quando l'applicazione viene usata maggiormente e se è opportuno cercare una soluzione diversa per gestire il carico di lavoro;



**Figura 4.7:** Esempio di utilizzo Firestore Database

- **Storage:** spazio web, strutturato in directory, nel quale è possibile memorizzare file. I file possono essere di qualunque tipo (quelli rilevanti in questo progetto sono file audio e immagine) e possono essere letti e caricati anche da client, tramite apposite API.  
In figura 4.8 è riportato il contenuto di una cartella dello storage. Nello specifico quella contenente i file audio e immagine utilizzati in Librania per la parola “disegno”.

gs://training-dsa.appspot.com > words > disegno			
	Nome	Dimensioni	Tipo
<input type="checkbox"/>	DISEGNO parola.mp3	30.38 KB	audio/mpeg
<input type="checkbox"/>	DISEGNO sillabe.mp3	41.34 KB	audio/mpeg
<input type="checkbox"/>	DISEGNO spelling.mp3	81.28 KB	audio/mpeg
<input type="checkbox"/>	image.jpg	35.71 KB	image/jpeg

**Figura 4.8:** Esempio di Storage. I quattro file visualizzati sono contenuti nella cartella `disegno`, corrispondente al percorso `/words/disegno/`

Anche in questo caso, come per Firestore Database, si possono specificare delle regole per gestire i permessi degli utenti. In questo semplice esempio (figura 4.9), ogni utente che ha effettuato il login può leggere tutti i file, ma nessun utente può modificarli o caricarne di nuovi.

```
service firebase.storage {
  match /b/{bucket}/o {
    match /{allPaths=**} {
      allow read: if request.auth != null;
      allow write: if false;
    }
  }
}
```

**Figura 4.9:** Esempio delle regole dello Storage

Esattamente come per Firestore Database, è possibile monitorare tramite un grafico temporale le richieste effettuate per gli elementi residenti nello Storage.

- **Hosting:** l’Hosting permette di rendere raggiungibile al pubblico la propria applicazione web. Dopo aver installato gli strumenti di Firebase nel proprio computer, è possibile, tramite linea di comando, eseguire il deploy del progetto. Terminata l’esecuzione del processo, apparirà sul terminale l'url che permette di accedere all’applicazione web. Tale indirizzo è visibile anche nella pagina “Hosting” nel sito di Firebase, nella sezione del progetto in questione. È possibile inoltre condividere in anteprima il proprio progetto sul web tramite un “canale di anteprima”: esso formulerà un link apposito, diverso da quello definitivo e “ufficiale”, utile per far testare l’applicazione ad un numero ristretto di persone. L'url definitivo dell’applicazione web è:

<https://training-dsa.web.app/>

L'url del canale di anteprima è nel formato:

<https://training-dsa-test-xxxxxx.web.app/>

dove le “x” corrispondono a caratteri alfanumerici.

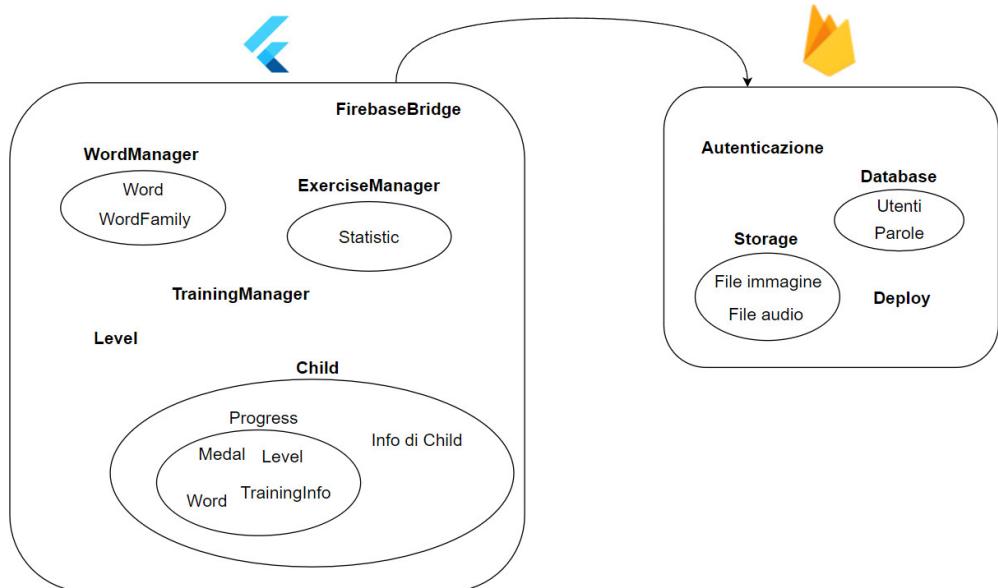
### 4.3 Struttura

L’applicazione è stata pensata per poter essere eseguita su dispositivi con sistemi operativi diversi, in modo da raggiungere il maggior numero di utenti possibile. Ma anche uno stesso utente deve poter accedere da più di un dispositivo (con lo stesso sistema operativo o meno) e l’applicazione dovrebbe rimanere *consistente*. Essere consistente significa che i progressi di un utente sono caricati correttamente ovunque, su qualunque dispositivo. A questo scopo, tutti i dati sono salvati online (figura 4.10), memorizzando nel dispositivo solo ciò che riguarda strettamente l’applicazione e non ciò che è legato all’utente.



**Figura 4.10:** Ogni dispositivo esegue l'applicazione sviluppata con Flutter ed è connesso a Firebase

Non solo ciò che riguarda l'utente è memorizzato online, ma anche ciò che riguarda le parole. Questa scelta è stata fatta per non appesantire l'applicazione, cosa importante per un dispositivo mobile sempre a corto di memoria, a discapito però di una maggiore quantità di dati scambiati con il cloud. Ma il motivo principale è che in questo modo è possibile modificare, aggiungere, e rimuovere parole senza dover aggiornare l'applicazione.



**Figura 4.11:** Struttura dell'applicazione con componenti principali

In figura 4.11 è riportato, in modo molto sintetico, come sono suddivisi i componenti e i servizi tra Flutter e Firebase, nella speranza che possa aiutare la comprensione della struttura del progetto. La sezione successiva, [Componenti](#), contiene i componenti di Flutter, quelli progettati per l'applicazione e riportati nel riquadro a sinistra della figura 4.11.

## 4.4 Componenti

L'applicazione, lato client, è composta da diversi componenti che interagiscono tra loro per il corretto funzionamento del sistema. Elenchiamo di seguito quali sono tali componenti, come funzionano, e come interagiscono tra loro.

### 4.4.1 FirebaseBridge

Il componente fa da collegamento tra l'applicazione e Firebase. Lo scopo del componente è fare da interfaccia dei servizi cloud, questo in previsione, se necessario, di sostituire Firebase con un altro servizio.

Il suo compito è proprio quello di rendere disponibili delle funzioni, in linguaggio Dart utilizzabili nel resto dell'applicazione web, per interagire col database online, leggendo e scrivendo dati. Queste funzioni sfruttano le API di Firebase specifiche per Dart. I tipi dei parametri in input alle funzioni e i valori di output nulla hanno a che fare con i tipi di Firebase, ma sono comuni tipi di Dart. Così facendo, se in futuro si decidesse di modificare il sistema di memorizzazione, sarebbe sufficiente adattare il contenuto di **FirebaseBridge**, lasciando inalterata l'interazione tra esso e il resto dell'applicazione web.

### 4.4.2 Word e WordFamily

Word è ciò che all'interno dell'applicazione rappresenta una parola. Ogni parola ha associate le seguenti informazioni:

- **spelling**: una stringa contenente la parola stessa;
- **sillabe**: una lista ordinata di stringhe contenente la sillabazione della parola: ogni stringa della lista è una sillaba della parola. Da notare che la sillabazione (fornita dalla professoressa Arfé) non è quella tradizionale, ma è adattata al contesto per facilitare l'apprendimento da parte dei bambini. Infatti, in questo caso, le lettere doppie non vengono divise;
- **temporizzazione sillabe**: durante la fase di allenamento, avviene la riproduzione di un file audio contenente la lettura della parola sillaba per sillaba. Questa informazione contiene gli intervalli temporali nei quali viene detta ogni sillaba. Si tratta di una lista di numeri interi, nella quale ogni numero rappresenta in millisecondi il momento nel quale la sillaba corrispondente viene detta;
- **significato**: testo contenente il significato della parola.

Per chiarire le informazioni appena riportate, prendiamo come esempio la parola “ghiacciolo” e mostriamo il contenuto di ciascuna variabile:

- **spelling**: “ghiacciolo”;
- **sillabe**: [“ghia”, “ccio”, “lo”];
- **temporizzazione sillabe**: [500, 1600, 2600];
- **significato**: “Gelato a base di ghiaccio colorato, arricchito con aromi vari.”

In figura 4.12 è riportato un asse temporale, che ha come unità di misura i millisecondi. Sopra l'asse sono poste le sillabe che compongono la parola “ghiacciolo”, disposte in base



**Figura 4.12:** Divisione in sillabe di una parola con temporizzazione

alla loro lettura nel file audio riprodotto durante l’allenamento. La sillaba “ghia” viene letta a circa 500 millisecondi di riproduzione, la sillaba “ccio” a circa 1600 millisecondi, e la sillaba “lo” a circa 2600 millisecondi.

The screenshot shows the Firebase Realtime Database interface. The path is `words > acciughe`. The database structure is as follows:

- training-dsa** (collection):
  - + Avvia raccolta
  - challenge
  - data
  - families
  - levels
  - training\_info
  - users
  - words** (collection)
    - + Aggiungi documento
    - acciughe** (document)
      - + Avvia raccolta
      - + Aggiungi campo
      - explanation: "Piccolo pesce commestibile di acqua salata."
      - syllables
        - 0 "a"
        - 1 "cciu"
        - 2 "ghe"
      - syllables\_times
        - 0 "700"
        - 1 "1700"
        - 2 "2700"
      - word: "acciughe"

**Figura 4.13:** Memorizzazione di una parola in Firebase

Come si può vedere in figura 4.13, le parole vengono memorizzate online in modo molto semplice. Esiste una raccolta `words` che contiene un documento per ogni parola. Ogni documento ha a sua volta i campi contenenti le informazioni della parola associata, che sono gli stessi elencati sopra.

Ogni parola fa parte di una famiglia di parole. Le famiglie previste sono:

- GH;
- CH;
- GN;
- GL;
- CQ-QU;
- SC;

- Doppie;
- Eccezioni.

Nell'applicazione esiste, per ogni famiglia, un oggetto `WordFamily` con una lista contenente tutte le parole associate alla famiglia stessa.

Path	Type	Value
/families/CH	Document	
/families/CH/CQ-QU	String	CQ-QU
/families/CH/Doppie	String	Doppie
/families/CH/Eccezioni	String	Eccezioni
/families/CH/GH	String	GH
/families/CH/GL	String	GL
/families/CH/GN	String	GN
/families/CH/SC	String	SC
/families/CH/Eccezioni	List	<ul style="list-style-type: none"> <li>0 "scherzare"</li> <li>1 "parrucchiere"</li> <li>2 "lucchetto"</li> <li>3 "orecchino"</li> <li>4 "inchino"</li> <li>5 "tacchino"</li> <li>6 "eschimese"</li> <li>7 "chiacchierare"</li> <li>8 "scricchiolare"</li> <li>9 "chilogrammo"</li> </ul>

**Figura 4.14:** Memorizzazione di una famiglia di parole in Firebase

Lato online, invece, come si può vedere in figura 4.14, in Firebase esiste una raccolta `families` con un documento per ogni famiglia prevista nell'applicazione. Ogni famiglia ha un campo `list`: una lista delle parole che compongono la famiglia.

Ho scelto di usufruire di questo approccio, piuttosto che aggiungere un campo ad ogni parola nel quale specificare la famiglia di appartenenza, principalmente per due motivi:

- una parola potrebbe appartenere a più famiglie;
- diminuire le operazioni lato utente: per sapere da quali parole è composta una certa famiglia, l'applicazione dovrebbe scorrere l'intera lista di parole (potenzialmente molto lunga) per determinare i componenti di ciascuna famiglia. In questo modo, a discapito di un po' di ridondanza nei server, si velocizzano le operazioni lato client, con maggiori vantaggi soprattutto per i dispositivi mobili che risparmiano più batteria.

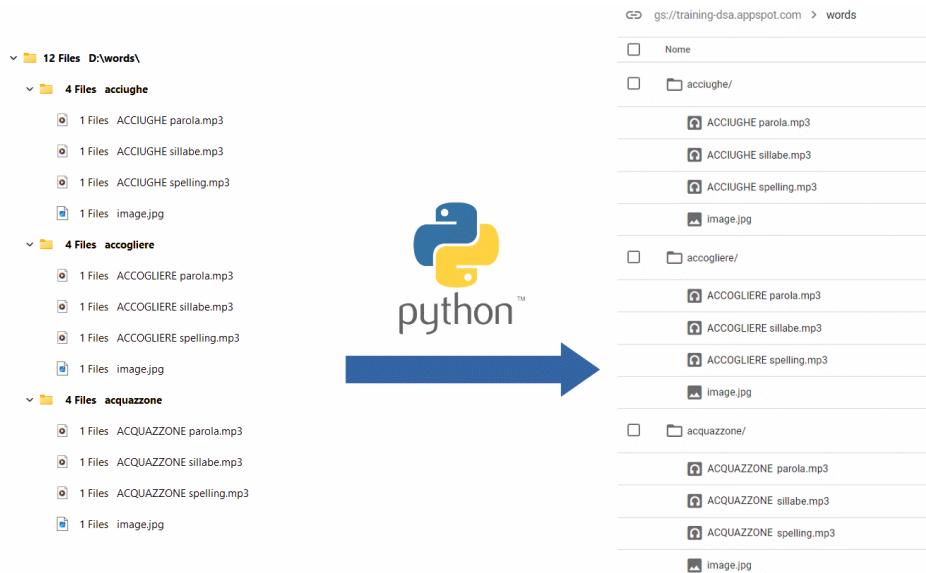
Per inserire le parole in Firebase si può disporre dell'interfaccia grafica della piattaforma. Tuttavia l'operazione richiede molto tempo considerando anche solo il numero attuale di parole e di famiglie. Per semplificare l'inserimento ho creato un semplice script in Python che, usufruendo delle API di Firebase, esegue l'upload di tutte le informazioni in pochi secondi. Esso utilizza come input un file nel quale tali informazioni sono inserite, esegue un parsing del file per determinarne il contenuto e suddividerlo appositamente, poi scrive le informazioni online. In [Appendice B](#) è riportata la struttura dei file e il loro contenuto.

Anche per i file audio e i file immagine l'inserimento nello Storage di Firebase tramite

interfaccia grafica richiede molto tempo. La struttura in directory prevede una cartella per ogni parola e ogni cartella contiene quattro file:

- audio con lettura parola;
- audio con spelling;
- audio con lettura per sillabe;
- immagine descrittiva della parola.

L’interfaccia non permette l’upload di cartelle o di più file contemporaneamente. È necessario quindi creare le cartelle una ad una e caricare i file singolarmente. Il procedimento risulta molto lungo quindi, anche in questo caso, ho creato uno script Python che crea in automatico la struttura delle cartelle nello Storage di Firebase ed esegue l’upload dei file in pochi secondi. Esso prende come riferimento una directory nel dispositivo nel quale viene eseguito lo script e replica nello Storage le cartelle e i file previsti (figura 4.15). Nel caso si vogliano aggiungere nuove parole, è sufficiente aggiornare la struttura delle directory e i file in esse contenute ed eseguire lo script per aggiornare i file online.



**Figura 4.15:** Trasferimento file da dispositivo locale (a sinistra) allo Storage di Firebase (a destra)

L’interfaccia dello Storage di Firebase non permette una visualizzazione come quella riportata in figura 4.15, è stata modificata appositamente per visualizzare meglio i file memorizzati e la loro struttura.

#### 4.4.3 Level

La classe `Level` rappresenta i livelli del gioco. I livelli sono ordinati e hanno:

- un nome;
- un numero che indica la loro posizione all'interno dell'ordinamento;
- una lista di famiglie che compongono il livello.

Un livello può non contenere solo le parole di una famiglia (ad esempio solo le parole della famiglia “GH”). Infatti il primo livello contiene sia le parole della famiglia “GH”, sia quelle della famiglia “CH”. Sarebbe bene che le parole visualizzate durante l'esecuzione di un livello siano scelte omogeneamente tra le diverse famiglie. Per questo motivo, `Level` mette a disposizione un metodo che ritorna una lista di parole bilanciata: se le famiglie associate al livello sono due, il metodo ritorna una lista contenente tre parole per famiglia.

Al momento le parole sono scelte in modo casuale, ma in futuro, tenendo in considerazione i dati raccolti durante gli allenamenti, si potrebbero mostrare le parole viste di meno.

I livelli previsti per il gioco sono, in ordine, i seguenti cinque:

1. “GH-CH”  
famiglie: GH, CH  
parole: *acciughe, ghiacciole, margherita, spaghetti, ghirigoro, cinghiale, righello, ringhiera, singhiozzo, ghirlanda, scherzare, parrucchiere, lucchetto, orecchino, inchino, tacchino, eschimese, chiacchierare, scricchiolare, chilogrammo*;
2. “GN-GL”  
famiglie: GN, GL  
parole: *ragnatela, disegno, ingegnoso, sostegno, magnete, sognare, compagna, segnalibro, usignolo, falegname, scogliera, quadrifoglio, artiglieria, coniglio, padiglione, accogliere, germogliare, groviglio, ventaglio, migliorare*;
3. “CQ-QU”  
famiglia: CQ-QU  
parole: *annacquare, acquitrino, acquazzone, risciacquare, obliquo, quadrifoglio, liquirizia, cinquanta, acquolina, soquadro*;
4. “SC-Doppie”  
famiglie: SC, Doppie  
parole: *pastasciutta, scienziato, sciopero, moscerino, sciabola, sciropo, sgusciare, pesciolino, sciatore, ascensore, serratura, sussurrare, assaggiare, cassetiera, passerotto, sotterrare, giubbotto, zattera, luccicante, spadaccino*;
5. “Eccezioni”  
famiglia: Eccezioni  
parole: *taccuino, igiene, aquilone, ciliegia, cuocere, aiuola, familiare, percuotere, riunione, allievo*.

I livelli numero 1, 2, e 4 contengono le parole di due famiglie.

L'ordine e la suddivisione dei livelli è stata decisa dalla professoressa Arfè.

I livelli, così come le parole e le famiglie, sono memorizzati in Firebase.

Come si può osservare in figura 4.16, la raccolta `levels` contiene un documento per ogni livello. I documenti hanno come identificativo l'ordinamento dei livelli ed ognuno di essi contiene due campi dati:

- `families`: lista delle famiglie che compongono il livello;
- `name`: nome del livello.

Path	Value
training-dsa / levels / 1	+ Aggiungi documento 1 2 3 4 5
training-dsa / levels / 1 / families	+ Avvia raccolta + Aggiungi campo ▾ families 0 "GH" 1 "CH" name : "GH-CH"

**Figura 4.16:** Memorizzazione dei livelli in Firebase

Come per le parole, ho scelto per semplicità di modifica della base dati, di creare uno script che, leggendo un file in locale contenente tutte le informazioni riguardo i livelli, aggiorna autonomamente il contenuto di Firebase. Nella sezione B.4 dell'[Appendice B](#), ho inserito la struttura e il contenuto del suddetto file.

#### 4.4.4 WordManager

`WordManager` è un componente fondamentale che gestisce tutto ciò che ha a che fare con le parole. Esso costruisce tutti gli oggetti `Word`, `WordFamily`, e `Level`, rendendoli pronti per l'utilizzo. Tramite `FirebaseBridge`, scarica dal cloud i dati necessari e con questi inizializza:

- le parole con relativi significati, sillabe, e tempistiche per le sillabe, reperendoli dai documenti presenti nella raccolta `words` di Firebase (figura 4.13);
- le famiglie delle parole, presenti nella raccolta `families` di Firebase (figura 4.14);
- i livelli, dai documenti della raccolta `levels` di Firebase. I nomi dei documenti, composti da un numero, indicano l'ordinamento dei livelli. I campi contengono invece il nome di ciascun livello e la lista delle famiglie che lo compongono (figura 4.16).

Esso è il punto di riferimento per quanto riguarda queste informazioni, infatti rende disponibili le liste delle parole, delle famiglie, e dei livelli all'interno dell'applicazione web. Oltre a ciò, `WordManager` contiene anche una funzione che ritorna una parola casuale, selezionandola, se necessario, da una famiglia specifica oppure tra tutte le parole presenti. Questa funzione si rende molto utile nella sua preparazione dei livelli

sia nella preparazione delle sfide, quando viene creata la lista delle parole da presentare all’utente. In particolare:

- nella preparazione dei livelli si specifica la famiglia a cui deve appartenere la parola casuale;
- nella preparazione delle sfide la parola viene scelta casualmente tra tutte.

#### 4.4.5 Child

**Child** è il componente più complesso del nostro progetto e contiene tutto ciò che riguarda l’utente, dalle informazioni personali ai progressi fatti. Descriviamo di seguito le parti che compongono **Child**. Va ricordato che tutte le informazioni sono reperite da Firebase ad ogni accesso, nulla viene memorizzato nel dispositivo, in modo che le informazioni siano consistenti tra dispositivi diversi.

I primi dati inseriti di un utente riguardano l’anagrafica:

- nome;
- cognome;
- età;
- username;
- email.

Non tutte queste informazioni sono necessarie. L’età, ad esempio, non è rilevante nell’applicazione ma, essendo questo un prototipo, sono state inserite perché potrebbero essere utili per ulteriori funzionalità delle quali accenneremo nella sezione [Sviluppi futuri](#).

Oltre ciò, **Child** contiene lo stato dell’utente. Questo comprende sia i progressi (livelli svolti, parole scritte correttamente, ...) descritti nella sezione [Progress](#), sia informazioni utili all’applicazione.

Vediamo queste ultime più nel dettaglio:

- **ultimo accesso:** viene memorizzato il timestamp dell’ultimo accesso alla schermata della scelta del livello. Questo riguarda la visualizzazione di un reminder, per fare in modo che venga mostrato non prima di un certo periodo di tempo. È stato richiesto che appaia solo alcune volte e, in questo modo, è possibile sapere da quanto tempo non viene visualizzato, per poter scegliere se mostrarlo o meno. L’accesso viene memorizzato online per far sì che l’intervallo di tempo rimanga consistente anche se l’utente accede da un dispositivo diverso;
- **visualizzazione dei tutorial:** se è la prima volta che l’utente raggiunge un punto dell’applicazione (ad esempio una pagina) nel quale è previsto un tutorial, viene memorizzato online se il tutorial è già stato visto, in modo da non riproporlo ogni volta.

Queste sono informazioni che riguardano principalmente il funzionamento dell’applicazione. Ciò che riguarda invece i progressi dell’utente nell’applicazione si trova nella classe [Progress](#), che vedremo nella sezione seguente, [Progress](#). Le informazioni degli utenti si trovano nella raccolta [users](#). Come si può vedere in figura 4.17, la quale contiene uno screenshot della struttura del database contenente i dati degli utenti, il documento di ogni utente, oltre ai campi dati sopra citati, contiene quattro raccolte che contengono i progressi.

The screenshot shows the Firebase Realtime Database interface. The path is 'users > vera.rossi'. The left sidebar shows a tree structure with 'challenge', 'data', 'families', 'levels', 'training\_info', and 'users' (which has a child node 'words'). The middle section shows a list of documents under 'users': 'android\_a22', 'android\_web', 'chiara', 'elio.costa', 'francesca', 'iphone', 'krizia', 'mac\_web', and 'vera.rossi' (which has children: 'web\_chrome\_win', 'web\_edge\_win', 'web\_firefox\_win', and 'web\_opera\_win'). The right section shows the details of the 'vera.rossi' document, including fields like 'known\_words', 'levels', 'training\_info', 'tutorial', 'age: 8', 'first\_time: false', 'name: "Vera"', 'starting\_training\_message\_date: 28 agosto 2022 00:40:08 UTC+2', 'surname: "Rossi"', and 'username: "vera.rossi"'. There are also '+' buttons for 'Aggiungi documento' and 'Aggiungi campo'.

**Figura 4.17:** Memorizzazione di un utente in Firebase

#### 4.4.6 Progress

Sono diverse le informazioni da memorizzare che riguardano i progressi fatti dall’utente. Ho scelto di creare una classe separata da `Child` per mantenere la struttura più ordinata. `Child`, quindi, contiene un’istanza di `Progress`.

I progressi riguardano le parole incontrate dall’utente, i livelli svolti, le medaglie conquistate, e le statistiche riguardo gli allenamenti.

Ogni informazione qui descritta è memorizzata sia nella memoria locale (e temporanea) dell’applicazione, ossia nelle variabili, sia online.

Vediamole ora nel dettaglio: sia come sono strutturate all’interno dell’applicazione, sia come sono salvate in Firebase.

##### WordProgress

Ai fini di valutare l’apprendimento del bambino, è necessario sapere quante volte scrive correttamente ciascuna parola. `WordProgress` memorizza, per ogni parola, tre informazioni:

- **timestamp:** data e ora di quando il bambino ha affermato di conoscere il significato della parola. Il significato viene chiesto non appena la parola viene visualizzata durante la fase di allenamento. In questo caso sarebbe sufficiente memorizzare un valore booleano, ma ho scelto di utilizzare un timestamp perché ritengo possa essere utile per uno sviluppo futuro (esso è descritto in [Sviluppi futuri](#)). Quando l’utente incontra nuovamente una stessa parola, se il timestamp è già stato memorizzato, non viene più chiesto all’utente se ne conosce il significato;
- **views:** numero di volte in cui il bambino ha incontrato la parola durante un allenamento, indipendentemente dal fatto che l’abbia scritta in modo corretto o errato;
- **errors:** numero di volte in cui il bambino ha scritto la parola in modo sbagliato.

Nel cloud, come si può vedere in figura 4.18, tali informazioni vengono memorizzate per ogni parola all’interno della raccolta `known_words` dell’utente e vengono aggiornate durante la fase di allenamento. `date`, invece, è aggiornato quando l’utente afferma di

The screenshot shows the Firebase Realtime Database interface. The left sidebar shows a tree structure for the path: users > vera.rossi > known\_words > acquazzone. Under 'known\_words', there is a list of words: acciughe, accogliere, acquazzone, acquitrino, acquolina, aiuola, allievo, annacquare, aquilone, artiglieria, ascensore, assaggiare, cassetiera, chiacchierare. On the right, there is a summary for the 'acquazzone' node: date: 19 maggio 2022 11:06:45 UTC+2, errors: 0, views: 2.

**Figura 4.18:** Memorizzazione delle informazioni riguardanti le parole, per ciascun utente in Firebase

conoscere il significato della parola, sia che questo avvenga durante l’allenamento, sia che avvenga durante una sfida.

### LevelProgress

LevelProgress tiene traccia dei progressi sui livelli e, nello specifico, contiene due informazioni:

- **timestamp**: data e ora di quando il livello è stato sbloccato. Questa informazione non è fondamentale, ma un suo possibile utilizzo viene accennato nel capitolo [Sviluppi futuri](#);
- **phase**: numero da 0 a 2 che indica qual è lo stato del livello:
  - 0: livello mai terminato;
  - 1: livello terminato (ma non completato);
  - 2: livello completato.

Gli aggettivi “terminato” e “completato” hanno significati diversi:

- “terminato”: il livello è stato giocato e il bambino ha scritto tutte e sei le parole arrivando alla schermata dei risultati;
- “completato”: il bambino ha scritto correttamente almeno una volta tutte le parole previste per il livello.

Un esempio che può chiarire le idee è il seguente.

*Un bambino gioca il primo livello e sblocca il secondo. Il primo livello, in questo momento, è stato “terminato”, perché il bambino ha tentato di scrivere le 6 parole previste per l’allenamento, mentre il secondo si trova ancora alla fase 0. Il primo livello prevede in totale 20 parole, ma il bambino può averne scritte correttamente al massimo 6 finora. Il bambino, anche dopo aver sbloccato il secondo livello, potrà giocare nuovamente al primo. Il primo livello sarà “completato” solo dopo che il bambino avrà scritto correttamente almeno una volta tutte e 20 le parole previste per il primo livello.*

Come si può vedere in figura 4.19, in Firebase esiste un **LevelProgress** per ogni livello e sono memorizzati tra le raccolte di ciascun utente. Il loro percorso è:

```
/users/[username]/levels/[level_number]/
```

Nel percorso riportato, **[username]** è il nome dell’utente per il quale sono memorizzati i dati e **[level\_number]** è un numero che indica la posizione del livello (ricordo che i livelli in questo prototipo sono ordinati da 1 a 5).

**Figura 4.19:** Memorizzazione delle informazioni riguardanti i livelli, per ciascun utente in Firebase

## Medals

Nell’applicazione sono previsti degli obiettivi che l’utente può raggiungere. Al raggiungimento di ciascun obiettivo, l’utente riceve una medaglia. Questo dovrebbe incoraggiare i bambini ad impegnarsi per raggiungere tutti gli obiettivi, in modo da aggiudicarsi più medaglie possibile.

Ogni medaglia ha:

- un nome identificativo: per identificarla in modo univoco nell’applicazione web;
- un nome visualizzato nella web app;
- una descrizione dell’obiettivo da raggiungere al fine di ottenere la medaglia;
- un’immagine;
- una funzione di validazione: una funzione che ritorna un valore booleano. Essa esegue dei controlli, per la maggior parte tra i progressi dell’utente, per verificare se l’obiettivo è stato raggiunto. Se ritorna valore positivo, significa che il bambino si è meritato la medaglia.

Le medaglie conquistate non sono memorizzate su cloud, ma vengono validate ad ogni esecuzione. Dato che le medaglie sono poche e la loro validazione non ha un grosso peso computazionale, ho preferito non occupare ulteriore memoria nei server. Se in futuro si dovessero prevedere nuovi obiettivi che hanno bisogno di maggiori risorse per essere verificati o ai quali saranno associati dei dati (come ad esempio la data di

conseguimento), potrebbe convenire una soluzione diversa da quella adottata ora. Vedremo più nel dettaglio quali sono gli obiettivi da raggiungere e le medaglie conquistabili nella sezione riguardante gli [Elementi di gamification utilizzati](#).

### Statistics

Obiettivo del prototipo è anche raccogliere informazioni sull'utilizzo giornaliero dell'applicazione web. Ai fini attuali, è sufficiente conoscere il numero di parole con cui un bambino si è esercitato, quindi le informazioni raccolte sono:

- **timestamp**: giorno di riferimento delle informazioni;
- **correct**: numero di parole scritte correttamente;
- **error**: numero di parole scritte in modo errato (una parola si considera errata se l'utente ha sbagliato entrambi i tentativi di scrittura disponibili);
- **levels**: elenco dei livelli giocati.

Le informazioni raccolte sono principalmente a scopo dimostrativo per impostare una struttura dati che possa raccogliere informazioni riguardanti l'allenamento giornaliero. Oltre al **timestamp**, nel prototipo sono utilizzati anche **correct** e **error**. Sommando questi due valori è possibile stilare una classifica degli utenti basata sul numero di parole esercitate.

Se l'utente gioca con l'applicazione, viene creato in Firebase un documento relativo a quel giorno, come riportato in figura 4.20. Qui verranno memorizzate queste informazioni. Esse sono salvate in Firebase al seguente percorso:

```
/users/[username]/training_info/[identificativo]/
```

dove:

- **[username]** è il nome utente del bambino a cui appartengono le informazioni;
- **[identificativo]** è un numero identificativo del giorno in cui sono stati raccolti i dati.

The screenshot shows the Firebase Realtime Database interface. The path is /users/vera.rossi/training\_info/20220419. The document structure is as follows:

```

{
  "vera.rossi": {
    "known_words": 0,
    "levels": [
      "training_info": [
        "20220415": 0,
        "20220419": 5,
        "20220425": 0,
        "20220427": 0,
        "20220503": 0,
        "20220507": 0,
        "20220508": 0,
        "20220509": 0,
        "20220515": 0,
        "20220518": 0,
        "20220519": 0,
        "20220525": 0,
        "20220526": 0
      ],
      "tutorial": 0
    ],
    "age": 8,
    "first_time": false,
    "name": "Vera",
    "starting_training_message_date": null,
    "surname": "Rossi",
    "username": "vera.rossi"
  },
  "20220419": {
    "correct": [
      0: "liquirizia",
      1: "acquitriño",
      2: "cinquanta",
      3: "risciacquare",
      4: "acquolina",
      5: "annacquare"
    ],
    "errors": 0,
    "levels": [
      0: "CQ-QU"
    ],
    "timestamp": "19 aprile 2022 14:31:28 UTC+2"
  }
}

```

**Figura 4.20:** Memorizzazione delle informazioni riguardanti le statistiche in Firebase

#### 4.4.7 ExerciseManager

**ExerciseManager** gestisce lo svolgimento di un livello. Entra in azione quando l'utente sceglie a che livello giocare e la sua attività termina quando il livello è finito. Per essere più specifici, è il componente che gestisce le fasi del metodo “The Mind’s Ear and Eye Training” e raccoglie le statistiche del livello che l'utente sta giocando.

**ExerciseManager** stabilisce innanzitutto:

- numero di parole da presentare all'utente durante un livello (attualmente impostato a 6);
- il numero di tentativi che hanno i bambini per scrivere correttamente una parola (attualmente impostato a 2).

Vediamo ora qual è il workflow di **ExerciseManager**, riportato brevemente anche in figura 4.21:

1. Quando l'utente sceglie il livello, **ExerciseManager** lo inizializza:
  - (a) imposta il numero di parole da presentare durante il livello;
  - (b) imposta il numero di tentativi che i bambini hanno a disposizione per scrivere correttamente ciascuna parola;
  - (c) reperisce da **Level** la lista di parole che saranno presentate durante l'esecuzione del livello. Le parole sono selezionate tra quelle associate al livello. Se ad un livello sono associate più famiglie (in questo prototipo al massimo 2), la lista sarà composta in modo equilibrato da parole di entrambe le famiglie;
  - (d) imposta la prima parola da esercitare.
2. fase di lettura della parola:
  - (a) comunica la parola alla Route;
  - (b) permette alla Route di sapere se è la prima volta che l'utente incontra quella parola, in modo che la pagina chieda, se necessario, se l'utente ne conosce il significato.
3. fase di lettura della parola (spelling):
  - (a) comunica alla Route la parola da mostrare.
4. fase di lettura delle parole per sillabe:
  - (a) comunica alla Route la parola da mostrare.
5. fase della “fotografia”:
  - (a) comunica alla Route la parola da mostrare.
6. fase di scrittura:
  - (a) comunica alla Route la parola corrente;
  - (b) memorizza se l'utente ha scritto la parola in modo corretto o errato.
7. schermata risultato parziale:
  - (a) stabilisce il contenuto della schermata in base al risultato della fase precedente;

- (b) sulla base del risultato precedente stabilisce lo step successivo dell’allenamento.

Il contenuto della schermata del risultato dipende infatti da **ExerciseManager**. In base alla performance dell’utente possono verificarsi i seguenti tre casi:

- se l’utente ha scritto la parola in modo corretto verranno mostrati dei complimenti;
- se l’utente ha scritto la parola in modo errato ma ha ancora dei tentativi a disposizione, verrà incoraggiato a riprovare;
- se l’utente ha scritto la parola in modo errato e non ha più tentativi a disposizione, la schermata mostrerà la parola scritta correttamente e l’utente viene invitato a proseguire.

Come scritto nel punto 7.b, **ExerciseManager** decide anche la prosecuzione o la terminazione dell’allenamento nella schermata del risultato parziale. I casi sono i seguenti:

- l’utente ha scritto la parola correttamente:
  - devono essere presentate altre parole:
    1. stabilisce la nuova parola;
    2. imposta come prossima fase la fase di lettura della parola (punto 2).
  - non devono essere presentate altre parole:
    1. notifica al **TrainingManager** che il livello è stato terminato;
    2. imposta come prossima schermata la Route con i risultati del livello.
- l’utente ha scritto la parola in modo errato:
  - l’utente ha ancora tentativi:
    1. decrementa i tentativi a disposizione;
    2. imposta come prossima fase la fase di lettura della parola (punto 2), mantenendo la stessa parola.
  - l’utente non ha più tentativi:
    - \* ci sono altre parole da presentare:
      1. stabilisce la nuova parola;
      2. imposta come prossima fase la lettura della nuova parola (punto 2).
    - \* non ci sono altre parole da presentare:
      1. notifica al **TrainingManager** che il livello è stato terminato;
      2. imposta come prossima schermata la Route con i risultati del livello.

**ExerciseManager** raccoglie alcune statistiche durante l’esecuzione di un livello. Al momento, le statistiche sono molto limitate, ossia tengono traccia solo di:

- numero di volte che una parola è stata scritta correttamente;
- numero di volte che una parola è stata scritta in modo errato.

In realtà, potrebbero raccogliere molte più informazioni che potrebbero essere utili a chi ha il compito di monitorare l’addestramento dei bambini. Ne parleremo brevemente nel capitolo [Sviluppi futuri](#).

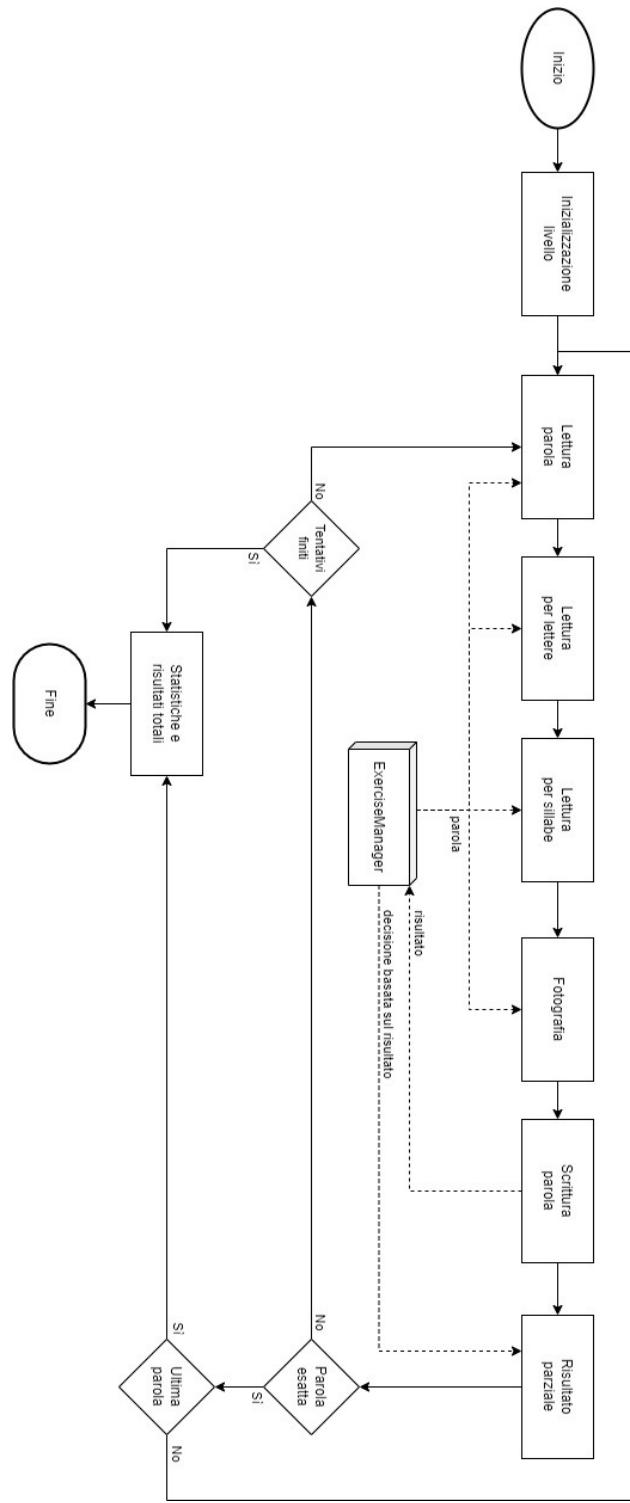


Figura 4.21: Workflow di ExerciseManager

#### 4.4.8 TrainingManager

**TrainingManager** gestisce l'allenamento complessivo, tenendo conto dei progressi dell'utente:

- tiene traccia dell'ultimo livello sbloccato, caricandolo da Firebase all'accesso dell'utente e aggiornandolo, anche online, nel caso venga sbloccato un nuovo livello;
- dà un limite all'allenamento giornaliero tenendo in memoria il numero di esercizi svolti nella giornata. Se l'utente ha raggiunto il numero massimo consigliato, notificherà alla Route della scelta dell'esercizio di chiedere all'utente se ha intenzione di continuare. Un'alternativa al numero massimo consigliato di esercizi è impostare un limite consigliato di tempo e, terminato questo tempo, si chiede all'utente se desidera continuare. Alla fine, è stato preso come riferimento il numero di livelli giocati, ma se nell'utilizzo dell'applicazione da parte dei bambini risulta migliore il secondo approccio, è sufficiente modificare la funzione di **TrainingManager** che verifica se è stata svolta la quantità massima di lavoro consigliata.

Oltre a questo, memorizza online i livelli terminati dai bambini. Al momento questa informazione non è utilizzata nell'applicazione, ma potrebbe essere utili a fini statistici (vedere la sezione [Sviluppi futuri](#)).

Quando l'utente finisce un livello, **TrainingManager** si occupa di:

- salvare le statistiche online;
- determinare se il livello è “terminato” o “completato”. Se necessario, aggiorna questo dato online.

## 4.5 Applicazione Web

Descriviamo in seguito come la web app si presenta all’utente e quali sono i meccanismi che le permettono di funzionare. Per il semplice motivo di non appesantire la tesi con troppe immagini, la maggior parte di esse sono screenshot della versione web dell’applicazione eseguita su un browser nel sistema operativo Windows. Laddove ci fossero delle differenze con altri sistemi, verranno mostrati gli screenshot delle versioni rilevanti, affinché risultino visibili al lettore.

La grafica della web app è pensata per ricordare la lettura e la scrittura. Gli elementi ripresi maggiormente sono:

- pergamene: tutte la pagine hanno come sfondo la texture di una pergamena, che ricorda vecchi libri e papiri.  
I titoli delle pagine o dei paragrafi non solo hanno una dimensione del font maggiore, ma sono scritti all’interno di una pergamena parzialmente arrotolata;
- matite: utilizzate per la maggior parte dei pulsanti. Molti pulsanti infatti hanno la forma di una matita e al loro centro riportano il testo del pulsante;
- gomma da cancellare: utilizzata nei form che richiedono una risposta affermativa o negativa.

Altri pulsanti sono invece composti da immagini. Un esempio è il pulsante per accedere alle impostazioni, rappresentato in figura 4.22: è un ingranaggio stilizzato, perché ricordi il tratto di una matita, e dunque della scrittura.

Altri pulsanti ancora sono composti da un’immagine e una descrizione, come quello in figura 4.23. Solo l’immagine è cliccabile e la descrizione ha lo scopo di dare un significato al pulsante, nel caso in cui l’immagine possa non essere sufficientemente esplicativa.

Tutti gli elementi cliccabili sono dotati di animazione: all’evento di **tap down**, il pulsante si rimpicciolisce leggermente, per poi tornare alla dimensione originale quando viene rilasciato o quando l’utente sposta la pressione del mouse o del dito al di fuori dell’area cliccabile.

La gomma da cancellare, riportata in figura 4.24, è composta da due pulsanti adiacenti che, come tutti gli altri, sono dotati di animazione.



**Figura 4.22:** Pulsante senza descrizione



Livelli

**Figura 4.23:** Pulsante con descrizione



**Figura 4.24:** Pulsante “Sì / No”

### 4.5.1 Login

La prima schermata che appare è quella di login (figura 4.25). Qui l'utente deve inserire il proprio username e la propria password negli appositi campi per poter accedere. A differenza dei classici form di login, la password è visibile all'utente, per rendere più semplice il suo inserimento da parte dei bambini con disturbi specifici dell'apprendimento.

Per procedere oltre, l'utente deve premere il pulsante “Accedi”.

Nel caso in cui il lo username inserito non esista nel sistema, l'utente viene notificato affinché lo corregga. La stessa cosa vale per la password: se non corrisponde allo username, viene chiesto all'utente di inserire quella corretta.

Affinché il login funzioni, l'account dell'utente deve essere presente in Firebase. Tutti gli account esistenti sono visibili nella sezione Authentication di Firebase. Il tipo di autenticazione utilizzato è “email/password”. Gli account vengono creati e gestiti dagli amministratori e le email hanno il formato [username]@trainingapp.it. Ulteriori informazioni riguardo la creazione di nuovi utenti si possono trovare nell'[Appendice B](#). Gli utenti inseriscono solo lo username, mentre la chiocciola e il dominio vengono aggiunti in automatico dall'applicazione. Tramite API di Firebase, le credenziali vengono trasferite e viene ritornato un valore che specifica se:

- il login ha avuto successo;
- lo username non esiste nel sistema;
- lo username esiste ma la password non corrisponde.

Dopo aver inserito correttamente le credenziali e aver cliccato (o premuto nel caso si utilizzi un dispositivo dotato di touch screen) il pulsante “Accedi”, la web app dà il benvenuto all'utente visualizzando il suo nome.

Se è la prima volta che l'utente accede, verrà mostrato un brevissimo tutorial (descritto nella successiva sezione [4.5.2](#)) che presenta l'applicazione e l'idea di aiutare la mascotte a mangiare le banane svolgendo gli esercizi.



**Figura 4.25:** Pagina di login

#### 4.5.2 Tutorial

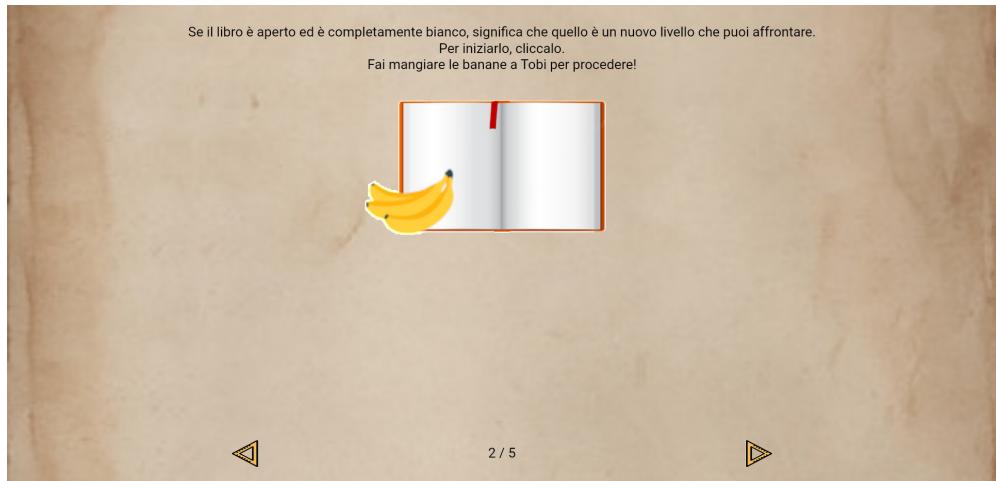
I tutorial sono delle schermate che forniscono agli utenti informazioni aggiuntive che aiutano la loro esperienza con l'applicazione.

Essi sono divisi in pagine, una di queste è riportata in figura 4.26. La parte superiore della schermata contiene la pagina vera e propria, con testo e immagini che saranno utili ai bambini per comprendere meglio la web app. La parte inferiore invece contiene:

- delle squadre, posizionate come fossero delle frecce, che permettono all'utente di spostarsi da una pagina all'altra.  
Nel caso della prima pagina, è presente solo la freccia per mostrare la pagina successiva.  
Nel caso dell'ultima pagina, la freccia a destra non ha più la funzione di cambiare pagina, ma bensì di chiudere il tutorial;
- il numero della pagina corrente e il numero totale di pagine che compongono il tutorial.

Al momento ci sono solo quattro tutorial che sono ritenuti fondamentali per la comprensione della web app. In futuro, come vedremo nella sezione [Sviluppi futuri](#), se ne potranno aggiungere altri.

Ogni utente, in Firebase, ha memorizzato una variabile booleana per ciascun tutorial dell'applicazione, il cui valore determina se l'utente lo ha già visualizzato. Quando è il momento di mostrare un tutorial, l'applicazione verifica prima se il tutorial è già stato visualizzato e, se l'esito della verifica è negativo, appare il tutorial. In questo caso, dopo che l'utente lo ha visualizzato per intero, la variabile online viene aggiornata, cosicché la volta successiva, non venga più mostrato.



**Figura 4.26:** Esempio di tutorial con 5 pagine

I tutorial ora presenti sono:

- presentazione dell'applicazione: espone lo scopo del gioco e com'è composta la web app;
- schermata selezione del livello: spiega come iniziare a giocare ad un livello e quali sono i significati dei componenti presenti nella schermata;

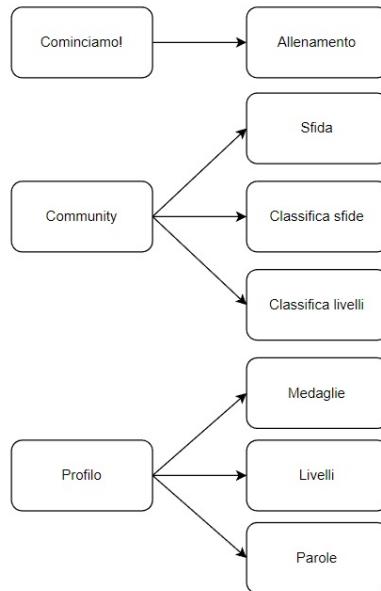
- allenamento: spiega come si svolge un livello, ossia la versione adattata del metodo “The Mind’s Ear and Eye Training”;
- pulsante “play” per riprodurre gli audio durante le diverse fasi di allenamento: questo tutorial viene mostrato solo se il sistema operativo è iOS. Il pulsante play infatti appare solo in questo caso, perché il sistema operativo richiede un’interazione da parte dell’utente affinché l’audio venga riprodotto.

#### 4.5.3 Home

La home è la schermata principale dell’applicazione ed è riportata in figura 4.28. Qui l’utente trova quattro pulsanti, che gli permettono di postarsi in sezioni diverse dell’applicazione web:

- **“Cominciamo!”**: iniziare l’allenamento (se è la prima volta che l’utente gioca verranno mostrate le istruzioni);
- **“Community”**: accedere alla community per misurarsi in sfide con gli altri utenti o visualizzare le classifiche;
- **“Profilo”**: accedere al proprio profilo per visualizzare i progressi e gli obiettivi raggiunti;
- **impostazioni** (ingranaggio in basso a destra): visualizzare la pagina delle impostazioni.

In figura 4.27 sono riportate le sezioni a cui ciascun pulsante porta.



**Figura 4.27:** Sezioni a cui conducono i pulsanti

Nell’applicazione Android di Librania, la home presenta un pulsante aggiuntivo, come si può vedere in basso a sinistra nella figura 4.29. Lo scopo del pulsante è uscire dall’applicazione: una volta premuto apparirà una dialog che chiede conferma all’utente se vuole uscire (figura 4.30).

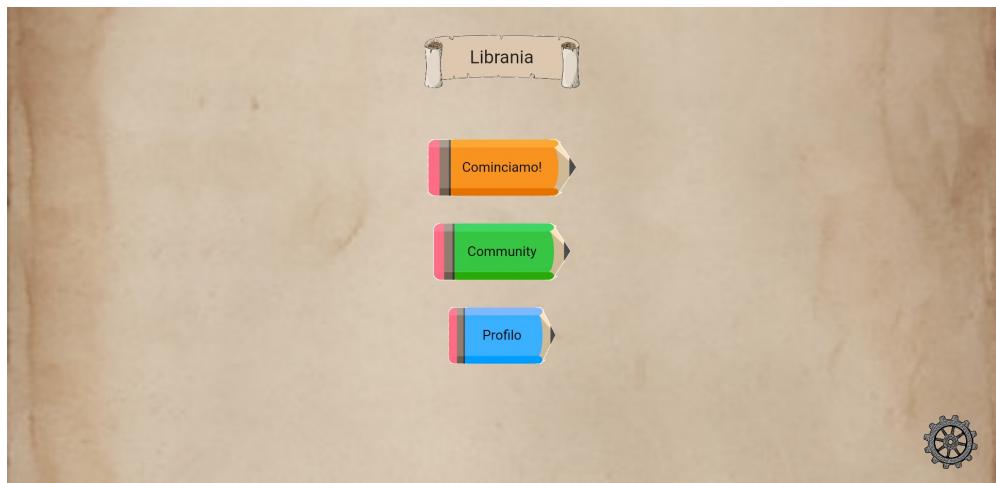


Figura 4.28: Home page di Librania



Figura 4.29: Home page di Librania  
(applicazione Android)



Figura 4.30: Conferma di uscita  
(applicazione Android)

#### 4.5.4 Istruzioni

Per iniziare l'allenamento, l'utente deve premere nella home il tasto “Cominciamo!”. Se è la sua prima volta, vengono mostrate delle istruzioni che illustrano il metodo da seguire, vale a dire il “The Mind’s Ear and Eye Training”. Queste istruzioni sono le stesse che vengono proposte ai bambini durante una sessione frontale e, come accenneremo nella sezione [Sviluppi futuri](#), dovrebbero essere adattate alla versione digitale. Nel caso in cui invece fossero già state illustrate, viene chiesto all'utente se le ricorda e, in base alla sua risposta, la web app mostra o meno le istruzioni. Per determinare se è la prima volta, viene preso in considerazione lo stato del primo livello: se il primo livello non è mai stato terminato, le istruzioni vengono mostrate senza chiedere all'utente se già le ricorda; l'utente potrebbe averle già viste se la volta precedente non ha terminato il primo livello. Questo significa che potrebbe non essersi cimentato a sufficienza con l'esercitazione, per questo potrebbe essere opportuno forzarlo a rivederle.

Le istruzioni sono divise in pagine. In ogni pagina (una di queste è visibile nella figura 4.31), nella parte inferiore dello schermo si può leggere la parte di istruzioni contenuta nella pagina; nella parte superiore, invece, è presente un’immagine legata al testo riportato nella parte inferiore. L’utente può leggere autonomamente il testo, ma una voce inizia a leggere le istruzioni in modo automatico all’apertura della pagina. Solo dopo che il file audio contenente la lettura è stato riprodotto completamente, in basso a destra appare una matita che invita il bambino a proseguire alla pagina successiva. La matita è animata e si muove orizzontalmente, rimbalzando sul lato destro.

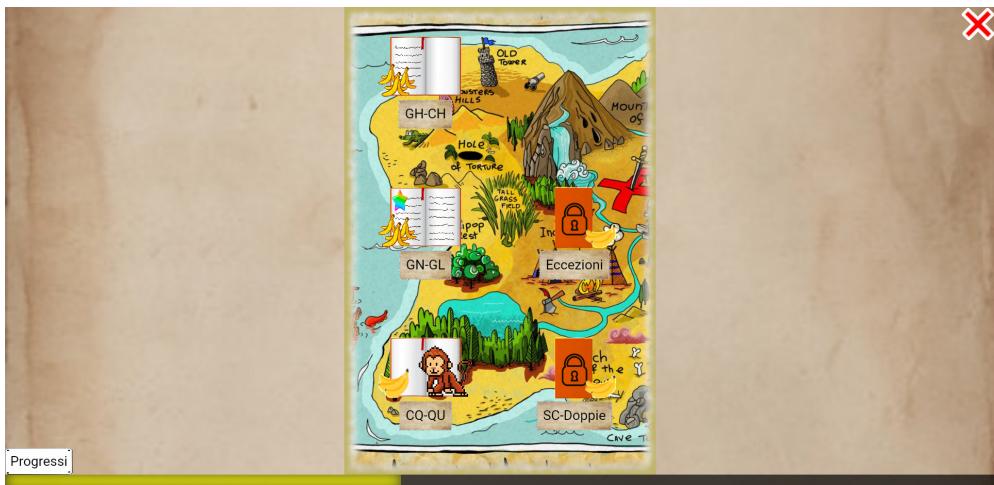
Le istruzioni riportate attualmente sono tratte dalla tesi *Le abilità ortografiche nella scuola primaria: Il training ortografico in italiano e in inglese* di Tamara Zancato [2], a pagina 57.



**Figura 4.31:** Schermata con una parte delle istruzioni che spiegano come svolgere gli esercizi

#### 4.5.5 Allenamento

La schermata della scelta del livello (figura 4.32) permette all'utente di scegliere con quale tipologia di parole cimentarsi per imparare a leggerle e scriverle. La prima volta che l'utente accederà a questa schermata, gli verrà mostrato un messaggio che dovrà motivarlo e / o ricordargli le fasi principali del metodo di allenamento (un esempio si trova in figura 4.33). È stato richiesto che questo messaggio appaia qualche volta e non sempre: al momento appare non prima di mezz'ora dall'ultimo accesso alla schermata. Per fare questo, la data e l'ora nella quale il messaggio è stato mostrato l'ultima volta viene memorizzata. In questo modo, ad ogni accesso, l'applicazione controlla il timestamp dell'ultima apparizione del messaggio e se è trascorsa più di mezz'ora, il messaggio viene mostrato e la variabile con la data e l'ora aggiornata.

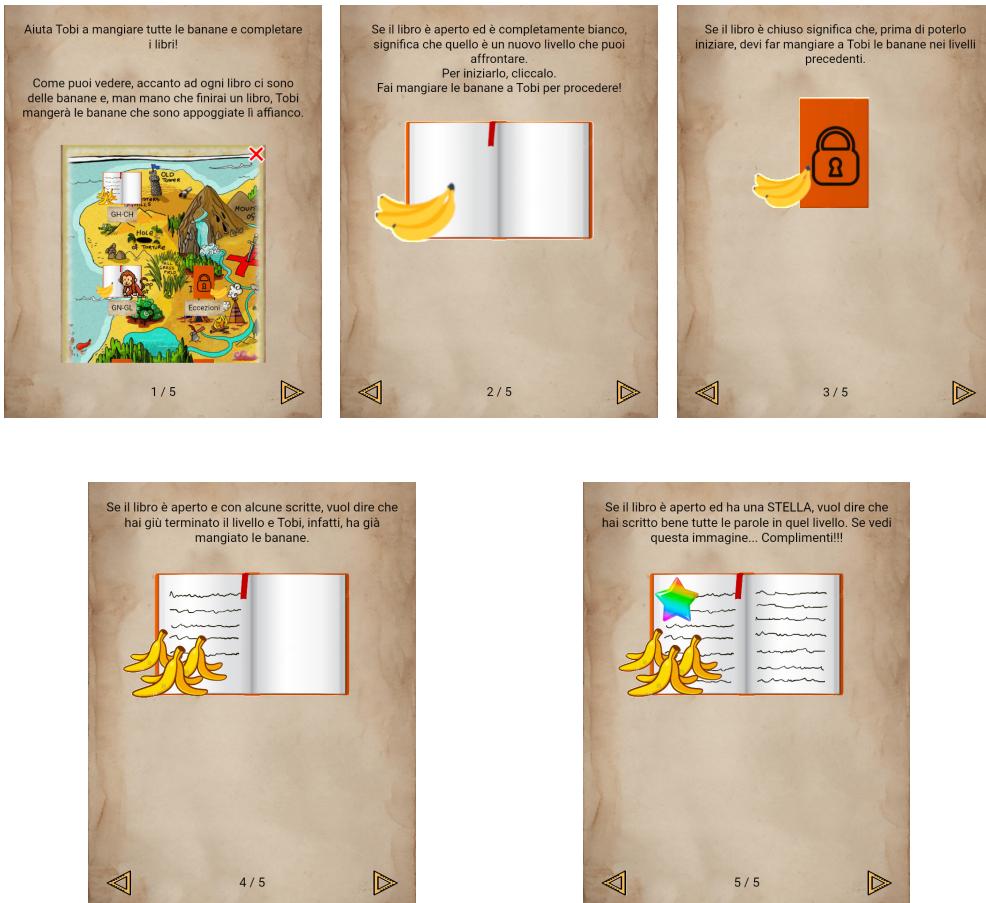


**Figura 4.32:** Pagina in cui si sceglie il livello da giocare



**Figura 4.33:** Pagina in cui si sceglie il livello da giocare con messaggio di reminder

La prima volta che l'utente accede a questa pagina, appare il tutorial che chiede di aiutare la mascotte Tobi a mangiare tutte le banane presenti nell'isola. La figura 4.34 è composta da 5 immagini, che corrispondono alle pagine del tutorial in questione. Esso spiega il significato degli elementi all'interno della schermata e come far mangiare le banane alla mascotte.



**Figura 4.34:** Tutorial della pagina della scelta del livello

Esistono cinque livelli ed ognuno di essi è affiancato da tre banane:

1. “GH-CH”;
2. “GN-GL”;
3. “CQ-QU”;
4. “SC-Doppie”;
5. “Eccezioni”.

Una volta terminato un livello, la mascotte Tobi mangia le tre banane ad esso associate e si sposta sul livello successivo (nella sezione [Sviluppi futuri](#) ho inserito alcune possibili migliorie).

Ogni livello è rappresentato da un libro e sotto di esso si trova una pergamena col nome del livello. Per quanto riguarda l'accessibilità, il contrasto tra la pergamena e la scritta al suo interno è in media 8.75:1, il che vuol dire che supera i test di accessibilità WCAG AAA.

La mascotte si trova all'ultimo livello sbloccato. Come descritto precedentemente, ogni livello può trovarsi in quattro stati: bloccato, nuovo, terminato, e completato; in base allo stato, l'immagine del libro cambia:

- **bloccato:** il libro è chiuso e sulla copertina è disegnato un lucchetto chiuso. Se l'utente preme su un livello bloccato, apparirà la scritta “Questo esercizio non è ancora stato sbloccato”, spiegando che il livello non può ancora essere giocato, come mostrato in figura 4.35;



**Figura 4.35:** Messaggio livello bloccato

- **nuovo:** il libro è aperto e le sue pagine sono bianche;
- **terminato:** il libro è aperto e ha alcune scritte. Questo sta ad indicare che in quel livello l'utente ha imparato a leggere e a scrivere alcune parole;
- **completato:** il libro è aperto ed entrambe le pagine sono completamente scritte. Questo indica che l'utente ha imparato a leggere e a scrivere tutte le parole previste in quel livello. Oltre a questo sul libro è presente anche una stellina.

Inizialmente, come si può vedere in figura 4.36, solo il livello “GH-CH” ha il libro aperto, mentre gli altri sono tutti chiusi. Infatti, all'inizio, solo il primo livello può essere giocato. Le banane sono tutte integre, perché non è stato terminato ancora alcun livello.



**Figura 4.36:** Solo il primo livello è giocabile

In figura 4.37, il giocatore ha “terminato” il primo livello (non ha dunque ancora scritto correttamente tutte le parole previste, ma ha comunque giocato il livello), sbloccando così il secondo. Come si può vedere, il secondo livello è stato “completato”, il che significa che l’utente ha giocato più volte questo livello fino a scrivere correttamente tutte le parole contenute in esso. Il terzo livello è sbloccato, ma non è ancora mai stato giocato, o meglio, “terminato”.

Nella parte inferiore della pagina si può vedere la barra dei progressi. Essa si riempie man mano che i livelli vengono terminati. Infatti, in figura 4.36 la barra è vuota perché nessun livello è ancora stato terminato. In figura 4.37, invece, due livelli sono stati terminati e la barra si è riempita per due quinti della sua lunghezza.



**Figura 4.37:** Primo livello “terminato”, secondo livello “completato”, terzo livello “sbloccato”, quarto e quinto livello “bloccati”

Man mano che l’utente gioca, la mascotte mangia le banane dei livelli terminati. Nell’immagine 4.37 infatti si può vedere che le banane dei primi due livelli sono state mangiate, mentre quelle degli altri livelli sono ancora intatte.

Cliccando su un livello giocabile, verrà eseguita la versione digitale del metodo “The Mind’s Ear and Eye Training”. Vediamo di seguito ciascuna fase, descrivendo sia l’interfaccia sia il suo funzionamento.

Le fasi sono:

1. lettura della parola;
2. lettura della parola per lettere;
3. lettura della parola per sillabe;
4. fotografia della parola;
5. scrittura della parola;
6. risultato.

### Lettura parola

La prima fase prevede la lettura della parola. La parola viene scritta al centro dello schermo e sopra di essa è posta un’immagine che rappresenta il significato della parola. Un esempio è riportato in figura 4.38.



**Figura 4.38:** Lettura parola

In questa schermata, viene riprodotto il file audio nel quale la parola è letta per intero. Nel caso il sistema operativo sia iOS, la riproduzione dell’audio non inizia automaticamente, ma ha bisogno dell’interazione dell’utente quindi, solo nel caso di iOS, è presente un pulsante che ricorda un tasto di “play” (figura 4.39), che fa cominciare la riproduzione dell’audio.



**Figura 4.39:** Lettura parola in iOS

Appena il pulsante “play” viene premuto, scompare e comincia la riproduzione del file.

Al termine della riproduzione, si verifica uno dei seguenti due scenari:

- è la prima volta che l'utente incontra questa parola e verrà chiesto se ne conosce il significato. In questo caso apparirà una finestrella che ha come titolo “Conosci questa parola?” nella quale si può rispondere “sì” oppure “no”. La finestrella è visibile nella figura 4.40 in basso. Contemporaneamente, viene riprodotto un file audio nel quale una voce chiede “Conosci il significato di questa parola?”. Se risponde in modo affermativo passa alla fase successiva, altrimenti l'utente viene rimandato alla pagina con la spiegazione della parola.

Quest'ultima pagina è molto semplice e un suo esempio è riportato in figura 4.41. La parola in esame fa da titolo. Il corpo centrale della pagina è scorrevole e contiene un'immagine che aiuta a comprendere il significato del termine, seguita dal significato riportato testualmente.



**Figura 4.40:** Richiesta significato parola



**Figura 4.41:** Spiegazione significato parola

Entrambe le ultime due informazioni (immagine e spiegazione testuale) non sono memorizzate nell'applicazione, ma si trovano in Firebase: l'immagine in Storage e il testo in Firestore Database. Questa scelta permette di aggiornare i dati senza dover aggiornare l'applicazione. In fondo alla pagina è presente il pulsante “Capito!” per tornare all'allenamento e, premendolo, in Firebase viene memorizzato che l'utente ha incontrato la parola corrente e che ne conosce il significato. In particolare, viene memorizzato il timestamp di quando preme il pulsante. Un semplice booleano sarebbe più che sufficiente per questo scopo, ma questa scelta potrebbe facilitare alcuni sviluppi futuri (accennati nella sezione **Sviluppi futuri**). Una parola viene considerata “non conosciuta” se il timestamp corrisponde al timestamp di default di Firebase (1 gennaio 2016), data che non verrà mai memorizzata in un utilizzo odierno dell'applicazione web.

Dopo aver premuto il pulsante nella pagina contenente il significato della parola, viene ricaricata la prima fase dell'allenamento, in modo che il bambino possa associare fin dalla prima fase la parola al suo significato. A questo punto, lo scenario risulterà equivalente a quello descritto nel punto successivo;

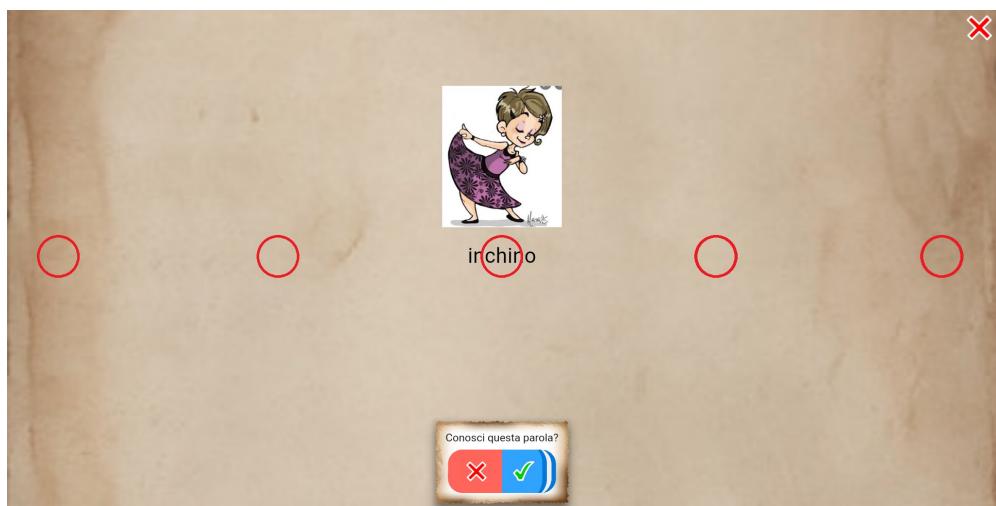
- l'utente ha già incontrato la parola e ne conosce il significato (il timestamp memorizzato è successivo al 1 gennaio 2016). In questo caso, dopo la lettura della parola, appare il pulsante “Avanti” per proseguire alla fase successiva. Il pulsante

appare solo alla fine affinché l'utente non si distraiga tentando di premerlo durante la lettura.

Per quanto concerne l'accessibilità, il contrasto tra la il colore della scritta e il colore dello sfondo è sufficientemente alto da rendere la scritta leggibile, superando i test WCAG per il testo normale. Il punto più scuro individuato sul quale può essere posizionata la scritta ha un contrast ratio di circa 7:1, un valore che permette di superare i test WCAG AAA. Quello appena descritto tuttavia è il caso peggiore. Per individuare un valore medio del contrasto tra la parola e lo sfondo, ho selezionato cinque punti e i rispettivi contrasti. Nell'immagine 4.42 si possono vedere cinque cerchi rossi disposti orizzontalmente. Essi spaziano sull'intero asse per coprire tutta l'area sulla quale può essere disposta la parola prendendo in considerazione diverse dimensioni dello schermo. Ho selezionato il colore del pixel centrale a tali cerchi e i contrasti ottenuti, partendo da sinistra, sono:

1. 8.8:1;
2. 11.35:1;
3. 12.32:1;
4. 11.79:1;
5. 7.97:1.

Ognuno di essi supera i test WCAG AAA. La media del contrasto della parola con lo sfondo, seppur approssimativa, risulta essere pari a 10.4:1 che, naturalmente, supera entrambi i test.



**Figura 4.42:** Selezione colore per test accessibilità

### Spelling

La seconda fase è lo spelling, in cui la parola viene letta lettera per lettera (anche qui è presente un tasto “play” per i dispositivi iOS). In questa schermata (figura 4.43), come richiesto, le lettere della parola sono visualizzate con una spaziatura maggiore, in modo che l’utente possa distinguerle meglio. In basso è presente il pulsante per iniziare la fase successiva, ma all’inizio non è visibile. Appena la pagina è pronta, viene riprodotto il file audio contenente lo spelling della parola e solo al termine della riproduzione il pulsante diventa visibile.

Anche in questo caso, il file audio non risiede tra gli asset applicazione, ma risiede nello storage di Firebase.

La leggibilità rimane invariata rispetto alla fase precedente, in quanto la parola copre i punti in cui i contrasti sono già stati misurati, quindi anche in questo caso i test WCAG AAA sono superati.



**Figura 4.43:** Fase di spelling

### Lettura per sillabe

La terza fase comprende la lettura della parola per sillabe. Come mostrato in figura 4.44, la parola compare nuovamente al centro dello schermo con la classica spaziatura tra le lettere, al contrario della fase precedente. A schermata pronta, viene riprodotto il file audio (anch’esso memorizzato in Firebase Storage) contenente la lettura per sillabe. Se il browser è installato in un dispositivo con iOS, anche qui apparirà il tasto “play”. Contemporaneamente alla riproduzione, una mano segue la lettura delle sillabe, indicando quella letta in quel preciso istante. Oltre ad essere indicata dalla mano, la sillaba corrente è anche evidenziata. Al termine della lettura, l’immagine della mano scompare e appare invece il pulsante per proseguire l’allenamento.

Nel database di Firebase ho memorizzato le tempistiche con le quali vengono lette le sillabe (descritto nella sezione [Word](#) e [WordFamily](#)). Durante la lettura viene monitorato il tempo di riproduzione e, in base a questo, la sillaba associata all’intervallo che comprende il tempo di riproduzione attuale viene evidenziata.

Inizialmente, l’evidenziazione delle sillabe consisteva nel colorare le lettere in modo

diverso ma, per superare i test WCAG, il colore delle lettere deve essere scuro, questo però rende la sillaba evidenziata difficilmente distinguibile dalle altre. Per questo motivo, attualmente, l'evidenziazione consiste nel colorare lo sfondo della sillaba corrente in giallo, come se fosse colorata con un evidenziatore. Il contrasto tra il nero della sillaba e il giallo dello sfondo è 17.19:1, un valore che supera abbondantemente le soglie dei test WCAG AAA.



**Figura 4.44:** Lettura delle sillabe

### Fotografia

In questa fase l'utente deve “fotografare” la parola nella propria mente.

La schermata è molto semplice: al centro della pagina si trova la parola e una macchina fotografica, con tanto di animazione e flash, simula una fotografia (figura 4.45). Al termine dell'animazione, l'utente può premere il pulsante “Avanti” per procedere.



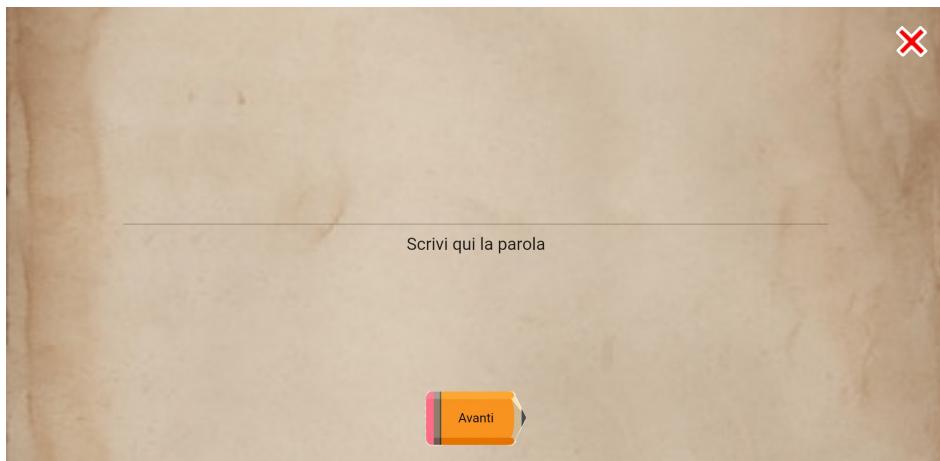
**Figura 4.45:** Fase della fotografia

### Scrittura

Questa è la fase decisiva, nella quale si verifica se l'utente ha imparato a scrivere correttamente la parola.

La figura 4.46 contiene uno screenshot dell'applicazione web durante questa fase. Al centro si trova una casella di testo, nella quale l'utente deve scrivere la parola. Una volta scritta, può premere il tasto “Invio” della tastiera oppure il pulsante “Avanti” in basso sullo schermo.

Sui dispositivi mobile, le tastiere virtuali generalmente correggono le parole scritte. Per questo motivo, la casella di testo è stata impostata come se fosse la casella di inserimento di una password con testo visibile, cosicché non ci sia alcuna assistenza da parte della tastiera virtuale. Prima di adottare questa soluzione, ho tentato di utilizzare alcuni plugin che hanno lo scopo di disabilitare questa funzione, ma purtroppo non mi hanno permesso di raggiungere il risultato sperato.



**Figura 4.46:** Fase di scrittura

### Risultato parola

La route di questa fase riceve come parametro la parola scritta dall'utente durante la fase precedente. Qui possono verificarsi tre scenari:

- la parola è stata scritta correttamente: indipendentemente che sia il primo o il secondo tentativo, l'utente riceve i complimenti della web app per aver scritto correttamente la parola. I complimenti sono rappresentati da un'immagine animata composta da un'emoji sorridente circondato da coriandoli e da una frase di congratulazioni (scelta casualmente da un insieme di messaggi). In basso si trova un pulsante che permette di procedere oltre:

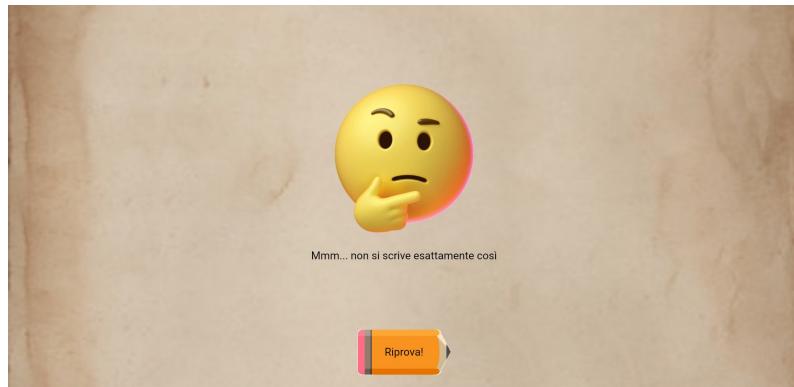
- se questa era l'ultima parola, la schermata successiva contiene i risultati del livello;
- se invece ci sono altre parole, l'utente viene reindirizzato alla prima fase con la nuova parola.

Riportiamo in figura 4.47 lo screenshot di questo scenario.



**Figura 4.47:** Parola scritta correttamente

- la parola è stata scritta in modo errato ma è il primo tentativo: l'utente ha due tentativi per scrivere correttamente la parola, quindi viene mostrato solo un messaggio che lo invita a continuare e un'immagine animata di un'emoji che riflette (figura 4.48). Cliccando il pulsante “Riprova!”, l'utente riprende dalla prima fase, con la lettura della parola. Tenere il conto dei tentativi è compito di `ExerciseManager`, tramite un'apposita variabile. Prima di spostarsi sulla route della prima fase, la variabile viene aggiornata;



**Figura 4.48:** Parola sbagliata al primo tentativo

- la parola è stata scritta in modo errato anche al secondo tentativo: l'utente non ha più tentativi e la parola viene considerata errata. Ora la schermata (figura 4.49) contiene l'emoji pensierosa del caso precedente ma, al posto del messaggio, viene mostrata la parola scritta in modo corretto.

Premendo il pulsante sottostante si procede come nel primo caso:

- se questa era l'ultima parola, la schermata successiva contiene i risultati del livello;
- se invece ci sono altre parole, l'utente viene reindirizzato alla prima fase con la nuova parola.



**Figura 4.49:** Parola sbagliata al primo tentativo

In questo momento, mentre l’utente visualizza il risultato della parola appena scritta, vengono aggiornate online le statistiche di tale parola, cioè:

- numero di volte che l’utente ha incontrato la parola durante gli allenamenti;
- numero di volte che l’utente ha scritto la parola in modo errato durante gli allenamenti (la parola è considerata errata solo nel caso in cui l’utente dovesse fallire entrambi i tentativi).

### Statistiche del livello

Terminato il livello, i risultati vengono salvati online:

- vengono aggiornate le informazioni riguardanti l’allenamento del giorno:
  - parole scritte correttamente;
  - parole scritte in modo sbagliato;
  - livelli giocati.
- viene aggiornato lo stato del livello:
  - se il livello non è mai stato giocato viene classificato come “terminato”;
  - se il livello è classificato come “terminato”, l’applicazione controlla se con le parole nuove scritte correttamente durante l’ultima esecuzione il livello può essere classificato come “completato” e, in tal caso, aggiorna il suo stato online;
  - se il livello è classificato come “completato”, non è necessario eseguire controlli, in quanto lo stato non può più essere modificato.

La prima schermata mostrata all’utente contiene la lista di parole nuove che ha incontrato durante il livello (figura 4.50). Qui viene effettuato il controllo di uno degli obiettivi che l’utente può raggiungere: conoscere il significato di tutte le parole.

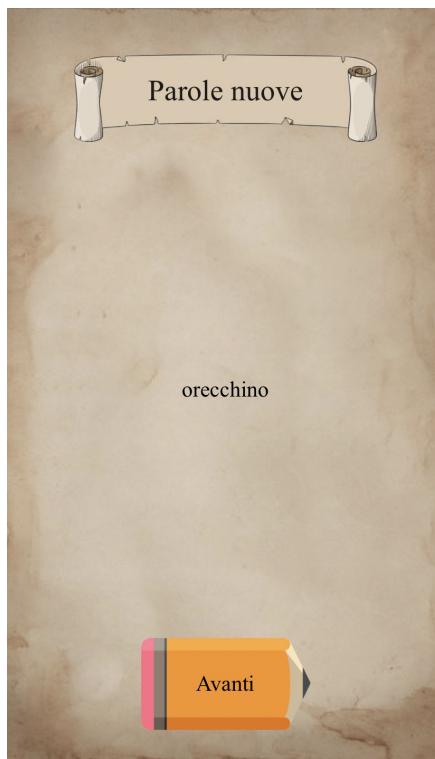
Indipendentemente dal fatto che abbia scritto correttamente le parole, l'utente dovrebbe averne imparato il significato. In questa schermata, quindi, se dal controllo risulta che l'utente conosce il significato di tutte le parole presenti nell'applicazione, viene assegnata la medaglia che certifica il raggiungimento di questo obiettivo.

È una schermata molto semplice, che ha come titolo “Parole nuove”, come corpo centrale la lista delle parole, e in basso un pulsante per passare alla pagina successiva. Nell'esempio di figura 4.50, l'utente ha incontrato solo una parola nuova (“orecchino”) durante lo svolgimento del livello.

Se l'utente non ne ha incontrate di nuove, prosegue direttamente con la schermata successiva, che riassume i risultati del livello (figura 4.51). Qui c'è una lista di tutte e sei le parole affrontate nel livello e ciascuna parola, se è stata scritta correttamente (è considerata corretta anche se scritta bene al secondo tentativo), è affiancata da una stellina. Nell'esempio riportato in figura l'utente ha scritto correttamente quattro parole su sei infatti, osservando le stelline, le ultime due sono state scritte in modo errato in entrambi i tentativi.

Anche qui viene effettuato il controllo per gli obiettivi raggiunti. In questo caso si contano le parole scritte correttamente: ogni dieci parole corrette l'utente guadagna una medaglia. Il conteggio riguarda la lista totale delle parole, vale a dire che scrivere correttamente più volte la stessa parola non fa incrementare il conteggio infatti, l'ultima medaglia ottenibile corrisponde al numero di parole nell'applicazione.

La struttura di questa pagina, molto semplice, è analoga alla precedente. È l'ultima pagina del livello e, premendo il pulsante in basso, si ritorna alla schermata della scelta del livello.



**Figura 4.50:** Parole nuove



**Figura 4.51:** Risultati livello

#### 4.5.6 Community

Per incoraggiare gli utenti e renderli più partecipi è stato creato uno spazio nel quale condividere i propri risultati. L'idea è che possano anche interagire tra loro tramite delle chat, nelle quali ognuno dispone di un avatar. Gli avatar si possono comprare all'interno di un negozio, scambiandoli per un certo numero di banane che sono conquistabili nei vari livelli. Al momento questa parte di interazione però non è presente, in quanto esula dallo scopo principale del nostro prototipo. Ora l'unico modo per interagire con gli altri utenti è tramite le sfide e le classifiche.

Le sfide mettono alla prova l'utente chiedendo di scrivere parole che ha già incontrato e altre che ancora gli sono nuove. Il vero obiettivo delle sfide è permettere di capire se l'utente ha appreso il metodo “The Mind’s Ear and Eye Training” e se riesce a generalizzarlo con parole mai incontrate.

Nella community sono presenti delle classifiche, basate sull'impegno dei bambini. Esse, su indicazione della professoressa Arfè, tengono conto del numero totale di parole affrontate e non solo di quelle scritte correttamente, questo per non svalorizzare lo sforzo dei bambini più in difficoltà.

La pagina della community (figura 4.52) permette di:

- cominciare una nuova sfida;
- visualizzare le classifiche relative alle sfide (della settimana corrente o totale);
- visualizzare le classifiche relative ai livelli (della settimana corrente o totale);



**Figura 4.52:** Pagina community

#### Sfida

Per cominciare una sfida, l'utente deve premere le due matite incrociate sopra la scritta “Comincia la sfida!!” nella pagina della community (figura 4.52).

La sfida, come accennato sopra, ha lo scopo di capire se i bambini riescono a generalizzare il metodo “The Mind’s Ear and Eye Training” anche su parole mai incontrate. L'ideale quindi sarebbe avere una lista di parole diverse da quelle utilizzate per l'allenamento, ma al momento sono le stesse.

Quando comincia la sfida, vengono scelte casualmente sei parole, indipendentemente dalla famiglia o dal livello a cui appartengono.

La sfida si svolge in modo molto simile a un livello:

1. lettura della parola (se è la prima volta che l'utente incontra questa parola viene chiesto se conosce il significato);
2. lettura dello spelling della parola;
3. lettura della parola per sillabe, evidenziandole;
4. “fotografia” della parola;
5. scrittura della parola;
6. risultato.

La differenza consiste nel fatto che, mentre in un livello i bambini hanno due tentativi per scrivere correttamente la parola (in questo caso devono imparare a scriverla), in una sfida hanno un solo tentativo (qui lo scopo è verificare il loro apprendimento). Nella fase del risultato, quindi, se la parola è corretta viene mostrata una pagina con i complimenti, se invece è scritta in modo errato viene mostrata la parola scritta correttamente.

Alla fine della sfida, quando l'utente ha tentato di scrivere tutte e sei le parole, una pagina mostra quante parole è riuscito a scrivere correttamente.

Le informazioni di una sfida che vengono memorizzare sono diverse da quelle memorizzate dallo svolgimento di un livello. Qui non viene memorizzato quante volte un utente ha visto una certa parola e quante volte l'ha scritta correttamente. L'unica informazione che al momento ci interessa è il *numero* di parole scritte corrette. Essa non viene memorizzata insieme alle altre informazioni all'interno dei documenti relativi a ciascun utente, ma è salvata in nella raccolta **Challenge** per semplificare l'accesso e permettere all'applicazione web di scaricare meno dati dal cloud di Firebase. Per ogni sfida, quindi, viene creato un nuovo documento che contiene:

- utente;
- numero parole scritte correttamente;
- timestamp della sfida.

### Classifiche

Le classifiche hanno lo scopo di accendere la competizione tra i bambini. Esistono classifiche relative a:

- sfide;
- livelli.

Entrambi gli ambiti hanno sia una classifica totale che tiene conto di tutti i dati presenti nel database, sia una classifica relativa alle attività svolte nell'ultima settimana.

I pulsanti per visualizzare le classifiche sono quattro, suddivisi in due righe, come si può vedere in figura 4.52. La prima riga rimanda alle classifiche riguardanti le sfide. La seconda riga, invece, rimanda alle classifiche riguardante i livelli. Il pulsante a sinistra di ogni riga mostra la classifica completa, che tiene conto cioè di tutti i dati raccolti fino

a quel momento. Il pulsante a destra mostra la classifica della settimana corrente. Per ricordare all'utente che questi ultimi pulsanti riguardano solo una settimana, l'icona contiene un calendario nel quale sono stati colorati di rosso i giorni che compongono una settimana.

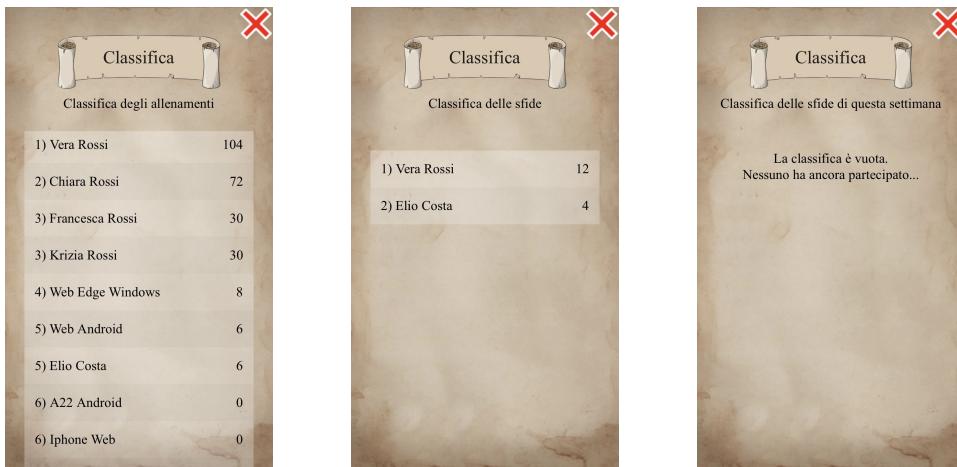
Il punteggio associato ad ogni utente nelle classifiche dei livelli corrisponde al numero di parole che ha incontrato. Al momento, per calcolare il punteggio, in Firebase vengono scorsi per ogni utente i documenti di ciascuna parola per poi sommare le volte in cui l'utente le ha incontrate. Questa però è un'operazione costosa in termini di tempo e richiede numerosi accessi al database (o anche solamente uno ma dovrebbe scaricare in locale tutti i dati di ciascun utente). Una soluzione migliore potrebbe consistere, invece, nel prevedere che ogni utente mantenga aggiornato un valore, al di fuori del proprio documento riservato, che corrisponda al suo punteggio. Il valore memorizzato sarebbe ridondante e occuperebbe maggiore spazio all'interno di Firebase, però questo permetterebbe di ridurre notevolmente gli accessi al database e le operazioni eseguite in locale dalla web app per calcolare i punteggi. Inoltre, in questo modo sarebbe possibile far sì che ogni utente possa accedere solo al proprio documento personale e non a quello degli altri.

Per le classifiche settimanali dei livelli, viene applicato un filtro che considera solo gli ultimi livelli giocati.

Per le classifiche delle sfide, invece, è stata adottata un'altra soluzione, con lo scopo di testare approcci diversi. In questo caso tutte le informazioni sono raccolte nel documento **Challenge**, esterno agli utenti, il che semplifica le operazioni da svolgere per stilare la classifica. Nel caso della classifica settimanale, è sufficiente aggiungere un filtro nella query di Firebase che seleziona solo i documenti delle sfide che hanno un timestamp uguale o seguente alla data di inizio della settimana.

In Librania, le classifiche sono visualizzate in righe che contengono:

- numero della posizione in classifica;
- nome e cognome dell'utente;
- punteggio.



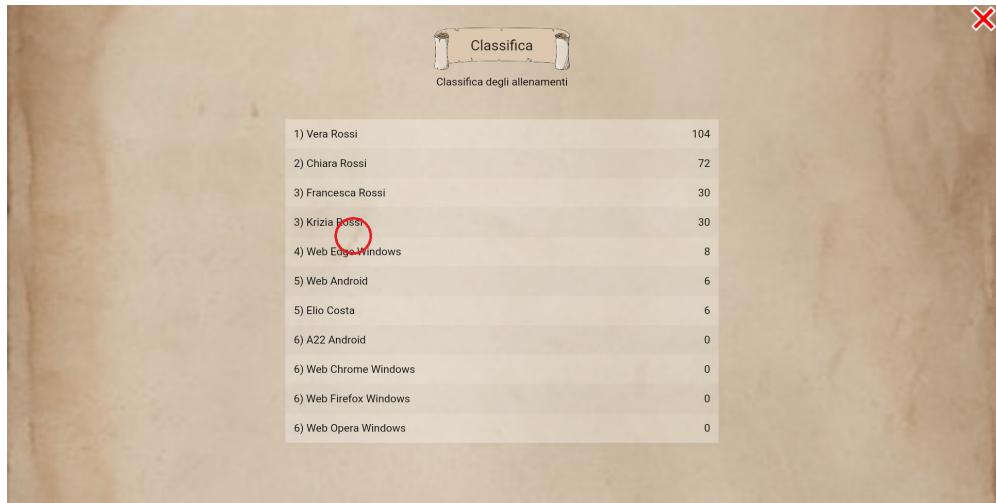
**Figura 4.53:** In ordine da sinistra: classifica livelli, sfide, e sfide settimanali

In figura 4.53, sono riportati gli screenshot di tre classifiche. La prima immagine a sinistra rappresenta una classifica totale degli allenamenti. L'immagine centrale contiene la classifica generale delle sfide. L'ultima immagine a destra dovrebbe contenere invece la classifica settimanale delle sfide però, in quella settimana, nessun utente ha terminato una sfida, quindi la classifica risulta essere vuota.

Per aumentare ulteriormente la leggibilità della classifica, le righe alternano leggermente il colore degli sfondi. Il colore in realtà è bianco per tutte, ciò che cambia è la trasparenza. In questo modo il colore risultante dipende dal colore di sfondo della pagina. Essendo il contrasto tra il colore del testo e lo sfondo della pagina già sufficiente per superare i test WCAG, aggiungere un po' di bianco può solo migliorare il risultato. Prendendo in considerazione il punto più scuro, evidenziato con un cerchio rosso in figura 4.54, riportiamo qui sotto i valori del contrasto degli sfondi con trasparenza diversa:

- 12.14:1;
- 12.39:1.

Entrambi i valori superano i test WCAG AAA.



**Figura 4.54:** Selezione colore per test accessibilità delle classifiche

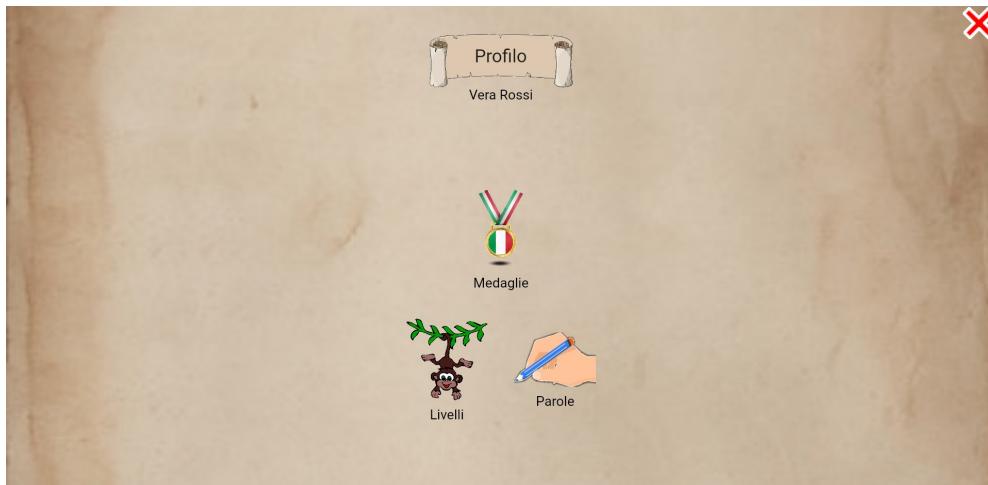
### 4.5.7 Profilo

La pagina del profilo (figura 4.55) permette all’utente di verificare i propri progressi. Può vedere:

- le medaglie ottenute;
- la completezza di ogni livello;
- le parole scoperte fin’ora.

Per visualizzare i progressi sopraccitati, l’utente deve premere il pulsante corrispondente. La grafica pensata inizialmente per questa pagina non era questa, vedremo meglio le cause del cambiamento nella sezione **Problematiche**.

I valori dei contrasti tra il colore della scritta e quelli della pagina sono gli stessi presenti nelle altre sezioni, quindi anche qui, secondo i test WCAG, le scritte risultano leggibili.



**Figura 4.55:** Pagina profilo

#### Medaglie

La schermata delle medaglie contiene tutte le medaglie guadagnate dell’utente (figura 4.56). Esse sono rappresentate da delle carte in rilievo rispetto allo sfondo, per dare un’idea di tridimensionalità. Le carte ricordano delle pergamene, per rimanere in tema con la scrittura, il cui bordo è più scuro. All’interno di ogni carta si trova l’immagine della medaglia e, sotto l’immagine, il nome della medaglia. Le carte sono disposte in una griglia, che viene riempita per righe, partendo dalla prima in alto, e ogni riga viene riempita cominciando dalla prima colonna a sinistra. Il numero di righe e colonne non è fisso, ma dipende dalla dimensione dello schermo o dalla dimensione della finestra, nel caso si utilizzi la versione web tramite un browser su un PC desktop o su un portatile. Ogni carta è cliccabile. Cliccando su una di esse apparirà un’altra carta, più grande e in primo piano rispetto tutto il resto, contenente anche la descrizione della medaglia, vale a dire qual è l’obiettivo che è stato raggiunto per ottenere quella medaglia (figura 4.57). Mentre questa è aperta, non è possibile cliccare su altre medaglie finché non si preme il pulsante “Chiudi”.



Figura 4.56: Pagina medaglie



Figura 4.57: Descrizione medaglia

### Livelli

Come detto precedentemente, la web app distingue tra livelli “terminati” e livelli “completati”. In questa schermata l’utente può vedere quante parole mancano per completare ciascun livello.

La pagina contiene una lista e ogni riga è associata ad un livello (figura 4.58). Nel caso in cui lo schermo fosse troppo piccolo per mostrare contemporaneamente tutti i livelli e nel caso in cui il loro numero dovesse aumentare, il corpo della pagina è già costruito per essere scrollabile. I livelli sono ordinati: l’ordine in cui appaiono è lo stesso con cui essi vengono sbloccati. Ogni riga contiene:

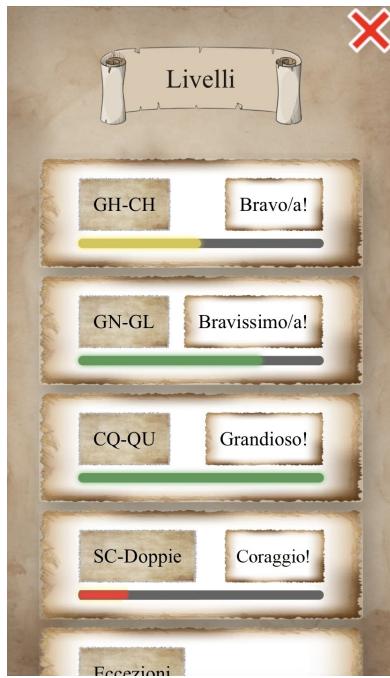
- nome del livello;
- un complimento che dipende dal grado di completezza del livello:
  - 0%: nessun messaggio, deve ancora iniziare;
  - fino a 33%: “Coraggio!”;
  - tra 33% e 50%: “Bravo/a!”;
  - tra 50% a 80%: “Bravissimo/a!”;
  - superiore a 80%: “Grandioso!”.
- una barra che indica la percentuale di completezza del livello: se è vuota significa che l’utente non ha mai giocato il livello; se è riempita a metà vuol dire che l’utente ha scritto correttamente la metà delle parole totali previste dal livello; se è piena significa che l’utente ha scritto correttamente tutte le parole. In ogni caso è una rappresentazione percentuale del numero di parole scritte correttamente

sul numero di parole totali del livello. La barra assume tre colori diversi in base alla percentuale che rappresenta:

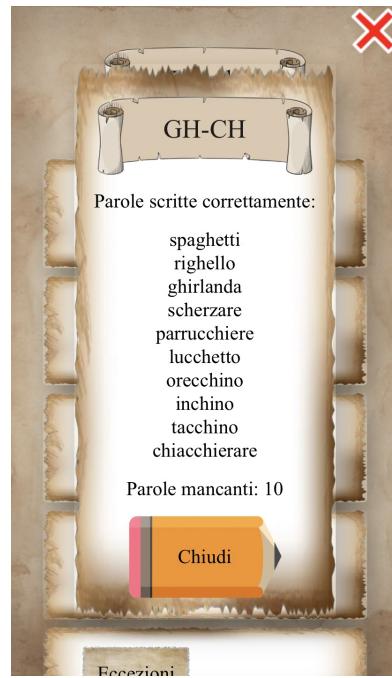
- minore di 33%: rosso;
- compresa tra 33% e 66%: giallo;
- superiore di 66%: verde.

Cliccando una riga si possono avere più informazioni circa la completezza del livello (figura 4.59): appare una finestrella in primo piano, nella quale sono elencate le parole scritte correttamente ed è riportato il numero di parole mancanti. Nel caso un livello abbia un numero tale di parole che l'elenco ha un'altezza maggiore dello schermo, il corpo della finestrella è scrollabile. Per chiudere la finestrella è necessario premere il pulsante “Chiudi”.

Per quanto riguarda l'accessibilità, i test WCAG AA e WCAG AAA sono superati. I nomi dei livelli sulla sinistra di ogni riga hanno un contrasto sufficiente a superare i test: il contrasto più basso riscontrato è 7.81:1, che è sufficiente sia per WCAG AA che per WCAG AAA. Il testo che si trova a destra, invece, ha contrasto massimo, essendo una scritta nera su sfondo bianco.



**Figura 4.58:** Pagina livelli



**Figura 4.59:** Progressi livello

### Parole

La pagina delle parole contiene l'elenco di tutte le parole presenti nel database. In figura 4.60 si può vedere uno screenshot di questa schermata.

Ogni riga dell'elenco è composta, partendo da sinistra, da:

- un semaforo;
- la parola;
- due numeri.



The screenshot shows a table titled "Parole" with a red "X" in the top right corner. The table lists words with their status (represented by a colored circle) and two numerical values (likely correct and incorrect attempts). The words and their entries are:

●	annacquare	0 / 0
●	artiglieria	3 / 3
●	ascensore	0 / 0
●	assaggiare	3 / 3
●	cassettera	0 / 0
●	chiacchierare	2 / 2
●	chilogrammo	0 / 0
●	cinghiale	1 / 1
●	cinquanta	0 / 0
●	coniglio	0 / 0
●		3 / 3
●		0 / 1
●		0 / 0

**Figura 4.60:** Pagina parole

Per ogni utente, le parole possono essere classificate come “conosciute” o “non conosciute”, in base al fatto se ha dichiarato o meno di conoscerne il significato (o se ha premuto “Capito!” nella pagina con la spiegazione della parola). In base a questa classificazione, le parole non conosciute vengono annerite: sopra di esse viene posta un’immagine simile ad uno scarabocchio, che ricorda la cancellazione di una parola fatta con un pennarello. L’immagine copre la maggior parte della scritta e lascia intravedere, a volte, le prime e le ultime lettere, per permettere ai bambini, se è di loro gusto, di provare ad indovinare la parola nascosta.

Il semaforo è uno strumento indicativo per capire se l’utente ha più difficoltà o meno a scrivere determinate parole. Per fare questo si tiene conto del rapporto tra il numero di volte in cui l’utente ha scritto correttamente la parola e il numero di volte che ha incontrato la stessa parola durante l’allenamento. Queste informazioni risiedono nel database di Firebase: ogni utente ha un documento relativo a ciascuna parola e qui sono riportati il numero di volte che ha incontrato la parola e il numero di volte in cui l’ha scritta in modo errato. Nelle righe, per scrivere il numero di volte in cui ha scritto la parola correttamente, viene semplicemente sottratto il numero di errori dagli incontri. Il semaforo può assumere quattro colori:

- nero: l’utente non ha mai tentato di scrivere la parola. Il semaforo è “spento”. Nel database risulta quindi che il numero di volte che l’utente ha incontrato la parola è 0;
- rosso: ha incontrato la parola almeno una volta e il rapporto è inferiore a 0.33;

- giallo: ha incontrato la parola almeno due volte e il rapporto è compreso tra 0.33 e 0.66;
- verde: ha incontrato la parola almeno una volta e il rapporto è superiore a 0.66.

Ai semafori è stata aggiunta un'ombra nera per aumentare la visibilità dei colori. Senza l'ombra, il rapporto di contrasto tra i colori del semaforo e lo sfondo sono:

- nero: 12.48:1;
- rosso: 2.24:1;
- giallo: 1.34:1;
- rosso: 1.69:1.

Tra questi rapporti, solo quello del nero supera i test WCAG.

Aggiungendo l'ombra nera, invece, i rapporti che prima non superavano i test ora diventano:

- rosso: 5.7:1;
- giallo: 17.19:1;
- rosso: 7.55:1.

Tutti i valori ora superano i test WCAG AA. I colori nero, giallo, e verde superano anche il test WCAG AAA.

Alla destra di ogni riga sono scritti due numeri, il primo a sinistra è il numero di volte che l'utente ha scritto correttamente la parola e il secondo, invece, il numero di volte che l'utente ha affrontato quella parola. Questa informazione è analoga al colore del semaforo, ma esplicita i numeri da cui dipendono il suo colore.

Il contrasto tra il colore dei numeri e lo sfondo è equivalente a quello della parola, dunque anch'esso supera i test WCAG AAA.

#### 4.5.8 Impostazioni

La pagina delle impostazioni (figura 4.61) è raggiungibile dalla home di Librania. Le impostazioni sono pensate per:

- apportare modifiche all'applicazione;
- rivedere i tutorial;
- uscire dall'account.

Le modifiche suggerite per l'applicazione non sono al momento disponibili perché non indispensabili per il funzionamento del prototipo. Esse sono accennate nella sezione [Sviluppi futuri](#).

I tutorial presenti ora sono solo quattro (considerando anche quello per il tasto “play” nei dispositivi con iOS), ma nell'applicazione finale dovrebbero essere di più, per spiegare nel dettaglio all'utente come funziona. Qui è presente un pulsante per ciascun tutorial e, premendolo, è possibile rivederlo.

L'ultimo pulsante disponibile serve per effettuare il logout. Se premuto, si viene reindirizzati alla pagina di login.



**Figura 4.61:** Pagina delle impostazioni

## 4.6 Elementi di gamification e per la piacevolezza della web app

Al fine di rendere l'applicazione web più accattivante e attraente per i bambini, sono stati inseriti degli elementi tipici dei contesti ludici, con la speranza che gli utenti percepiscano Librania più come un intrattenimento che come l'ennesimo compito scolastico per casa.

Per raggiungere questo obiettivo mi sono affidato a concetti che tentino di umanizzare l'applicazione web e di immergere l'utente in un semplice contesto. L'ambientazione che si vuole creare è quella di un'avventura su un'isola, nella quale vive la scimmia Tobi.

Vediamo di seguito quali elementi ho utilizzato.

### Mascotte

La mascotte è un elemento che aggiunge personalità all'applicazione web e le fornisce un'identità. La mascotte scelta è una scimmietta di nome Tobi, raffigurata in figura 4.62. L'idea è che Tobi sia l'elemento di congiunzione tra l'utente e l'applicazione web.



**Figura 4.62:** La mascotte Tobi

Infatti, lo scopo dell'utente è aiutare la scimmietta a mangiare più banane possibile, creando in questo modo un legame tra esso e la mascotte. Questo potrebbe far sì che l'utente si senta più coinvolto nell'avventura della mascotte e, di conseguenza, anche nell'applicazione web.

### Livelli

L'utilizzo dei livelli è un classico dei videogiochi. L'utente è incoraggiato a completarli e passare ai livelli successivi spinto dalla voglia di superare i propri limiti e dal riconoscimento dei suoi successi.

Generalmente, i livelli successivi sono più difficili e impegnativi rispetto ai precedenti per mettere maggiormente alla prova le abilità del giocatore. Non è però questo il caso di Librania. Qui i livelli non hanno difficoltà diverse in base al loro ordinamento, ma sono semplicemente un modo per suddividere l'allenamento dei bambini. Nonostante ciò, gli effetti che i livelli hanno sugli utenti non vengono meno. La soddisfazione del loro superamento rimane, perché i punteggi in classifica aumentano e il livello raggiunto è certificato dalla posizione della mascotte nella schermata della selezione del livello: se due utenti mettono a confronto le proprie schermate, possono determinare chi è riuscito ad avanzare maggiormente. Un altro metodo di confronto si trova nella pagina dedicata ai livelli all'interno della sezione del profilo, dove sono mostrati i progressi di ciascun livello e le parole mancanti per completarlo.

### Competizione, classifiche, e status

La competizione accende lo spirito di rivalità tra i giocatori. Essa è implementata in Librania attraverso delle classifiche. Esistono in totale quattro classifiche che possono essere divise in due categorie:

- totali: tengono traccia di tutte le attività svolte senza limiti di tempo;
- settimanali: considerano solo delle attività dell'ultima settimana.

Il primo tipo di classifica può rappresentare una competizione a lungo termine per determinare chi si allena di più. La seconda, invece, è una competizione a breve termine, che può riguardare sfide occasionali.

Nei giochi che prevedono l'interazione tra giocatori, essi sono generalmente contraddistinti da uno status sociale, che dipende dalle loro capacità, dai risultati ottenuti, dalle sfide superate. Nel nostro prototipo non è stato implementato al momento un vero e proprio status, ma una sua versione parziale può essere derivata dalle classifiche, basandosi sul posizionamento di ogni utente. Parliamo di una versione di status "parziale" perché non viene tenuto conto delle medaglie ottenibili nel prototipo. Uno status vero e proprio dovrebbe tenere conto sia della classifica sia delle medaglie ottenute.

### Medaglie

Le medaglie riconoscono il raggiungimento di un obiettivo, il superamento di una sfida, e contribuiscono alla reputazione e allo status di un giocatore.

Librania prevede delle medaglie ottenibili tramite l'assolvimento di alcuni compiti, ma sono visibili solo all'utente che le ottiene. Affinché esse possano agire come status, sarebbe opportuno che gli utenti possano verificare quali sono le medaglie in possesso di altri rivali. Questo attualmente non è previsto nel prototipo, ma due bambini possono confrontarsi affiancando i loro dispositivi ed esaminando le pagine contenenti le medaglie di ciascuno.

Le medaglie presenti attualmente nel prototipo hanno obiettivi poco originali e semplici da verificare:

- medaglia di inizio del primo allenamento;

- medaglia per aver affermato di conoscere il significato di tutte le parole presenti nel prototipo;
- una medaglia per la terminazione di ogni livello;
- una medaglia ogni dieci parole scritte correttamente, fino al numero totale di parole presenti.

Ma le medaglie possono essere più accattivanti, come ad esempio:

- terminare più livelli consecutivamente senza commettere errori;
- finire una sfida o un livello entro un limite di tempo scrivendo correttamente un numero minimo di parole;
- terminare almeno un livello al giorno per un certo numero di giorni consecutivi.

Più le medaglie avranno obiettivi complessi da raggiungere, più saranno considerate di valore. Quindi ogni medaglia acquisirà un peso diverso e tutto ciò contribuirà allo status dell'utente.

## 4.7 Problematiche

La prima difficoltà riscontrata riguarda la fase di scrittura, descritta nella sezione [Scrittura](#). La versione per PC non presenta alcun problema, ma la versione per cellulari e tablet al contrario sì. Quando l'utente seleziona la casella di testo per scrivere la parola, essa potrebbe venir coperta dalla tastiera. Questo dipende dalle dimensioni dello schermo. Adattare efficacemente la grafica quando la tastiera è presente non è semplice. Una possibile soluzione potrebbe prendere spunto dalla figura [2.10](#) di [Percivals Help For Dyslexia](#), nella quale si può vedere una tastiera costruita ad hoc. Così facendo la versione per cellulari e tablet non presenterebbe più il problema e, tramite un'API fornita da Flutter, questa tastiera potrebbe apparire solo se Librania è eseguita su mobile e non su PC.

Un altro problema riscontrato riguarda l'orientamento del dispositivo che esegue l'applicazione. L'applicazione web è pensata per essere usata su dispositivo mobile in modalità portrait e, impostando i corretti parametri, la versione nativa dell'applicazione Android mantiene la modalità selezionata nonostante il dispositivo venga orientato diversamente. Se però l'applicazione web è eseguita in un browser (sempre su dispositivo mobile) e il dispositivo si trova in modalità landscape, anche il browser passerà in modalità landscape, forzando a sua volta l'interfaccia dell'applicazione web. Questo purtroppo compromette l'usabilità, in quanto molti elementi fuoriescono dallo schermo.

Sono state poi riscontrate delle difficoltà riguardanti le performance. Sono da tenere in particolare attenzione le risorse a disposizione dei dispositivi sui quali può essere eseguita Librania e delle risorse richieste da ciascun widget di cui usufruiamo. Riporto ora un esempio di quando le risorse impiegate erano adeguate nell'esecuzione da PC e in eccesso nell'esecuzione da dispositivo mobile.

La pagina del profilo utente (figura [4.55](#)) ora contiene tre pulsanti, tramite i quali si può accedere alle sezioni “Medaglie”, “Livelli”, e “Parole”. Inizialmente, però, la pagina era pensata per contenere tutte queste informazioni. L'idea era raggruppare ogni

sezione all'interno di uno spoiler e se l'utente desiderava vedere le medaglie ottenute, i progressi dei livelli, o quelli delle parole, poteva espandere lo spoiler associato per esporne i dettagli.

Questa soluzione però si è dimostrata avida di risorse, soprattutto per la visualizzazione delle parole: ad ogni parola era associata una card (figura 4.63), contenente una progress bar che indicava il rapporto tra volte in cui la parola era scritta correttamente e scorrettamente, cambiando il colore della barra in funzione del rapporto.



**Figura 4.63:** Layout iniziale delle parole

Sebbene questa soluzione funzionasse su un dispositivo desktop, l'esecuzione su cellulare e tablet si trovava evidentemente in difficoltà nel gestire tutti gli elementi grafici. Questo portava ad una pagina molto poco reattiva, che risultava quasi inutilizzabile. Per questo ho scelto di suddividere le sezioni “Medaglie”, “Livelli”, e “Parole” in tre pagine diverse ma, nonostante ora gli elementi da gestire fossero minori, la pagina contenente i progressi delle parole risultava sempre poco reattiva: non solo i tempi di caricamento erano lunghi ma mancava anche la fluidità dello scroll. Mi sono visto costretto dunque a ripensare il modo di rappresentare graficamente i progressi, utilizzando elementi grafici molto semplici. Il risultato è quello attuale del prototipo, nel quale il rapporto tra volte in cui una parola è scritta in modo corretto ed errato è riportato testualmente sulla destra, mentre il colore che prima riempiva la barra ora rappresenta un semaforo, posto alla sinistra di ogni parola.



# Capitolo 5

## Test

*In questo capitolo sono riportati i risultati dei test effettuati sul progetto sviluppato.*

### 5.1 Test funzionali e di usabilità

La web app è stata testata su diversi dispositivi per garantirne il corretto funzionamento. Oltre alla versione web è stata testata anche la versione compilata in apk esclusiva per i dispositivi Android.

Durante i test è stata prestata attenzione a:

- funzionamento delle feature dell'applicazione, come ad esempio:
  - fasi del metodo “The Mind’s Ear and Eye Training” e della sfida;
  - correttezza nel calcolo del punteggio delle classifiche;
  - conquista delle medaglie;
  - corretta memorizzazione e lettura dal database.

Valutando anche il corretto funzionamento dei plugin sulle diverse piattaforme e sistemi operativi;

- grafica, verificando che si adatti a dimensioni diverse dello schermo e della finestra.

Elenchiamo quali sono i sistemi operativi su cui è stato testato il prototipo, specificando la modalità di esecuzione (web o nativo):

- Web:
  - Windows:
    - \* Edge;
    - \* Chrome;
    - \* Firefox;
    - \* Opera.
  - Linux (Ubuntu):
    - \* Firefox

- MacOS:
  - \* Safari;
  - \* Chrome.
- Android:
  - \* Chrome.
- iPhone:
  - \* Safari.
- Nativo:
  - Andoid.

Per ogni combinazione di sistema operativo e browser da testare è stato creato un utente apposito, per poter testare tutte le funzionalità da zero senza influenze dalle sperimentazioni precedenti.

Riportiamo di seguito i risultati dei test effettuati per ogni sistema operativo.

## Windows

Windows (figura 5.1) è un sistema operativo dell'azienda Microsoft, per computer desktop, laptop, e tablet. È il sistema operativo più diffuso al mondo.



**Figura 5.1:** Logo Windows

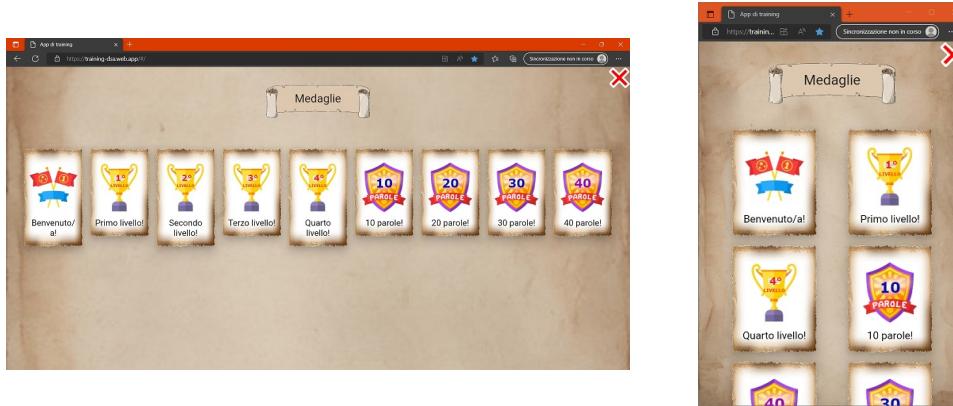
I test sono stati effettuati su Windows 10 e Windows 11, accedendo alla web app solo tramite browser, più precisamente con Edge, Chrome, Firefox, ed Opera. Non stati riscontrati problemi di compatibilità, nemmeno per i plugin Flutter utilizzati.

Il browser, non eseguito in schermo intero ma all'interno di una finestra, può essere ridimensionato, costringendo l'interfaccia ad adattarsi alle nuove dimensioni, ma nemmeno la grafica riscontra problematiche.

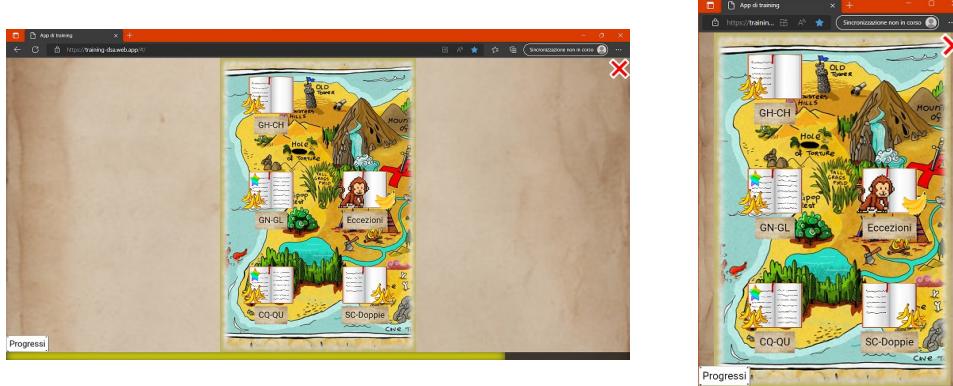
Inseriamo in figura 5.2 alcune immagini della web app eseguita su dimensioni differenti della finestra che la ospita. Come si può vedere l'interfaccia si adatta coerentemente e, se necessario, con dei compromessi affinché continui ad essere usabile ed accessibile.

Riportiamo anche altre immagini (figure 5.3 e 5.4) di screenshot della web app in esecuzione sul browser Edge di Windows. Anziché proporre pagine diverse, affiancheremo alcune immagini della stessa pagina ma con dimensioni della finestra differenti.

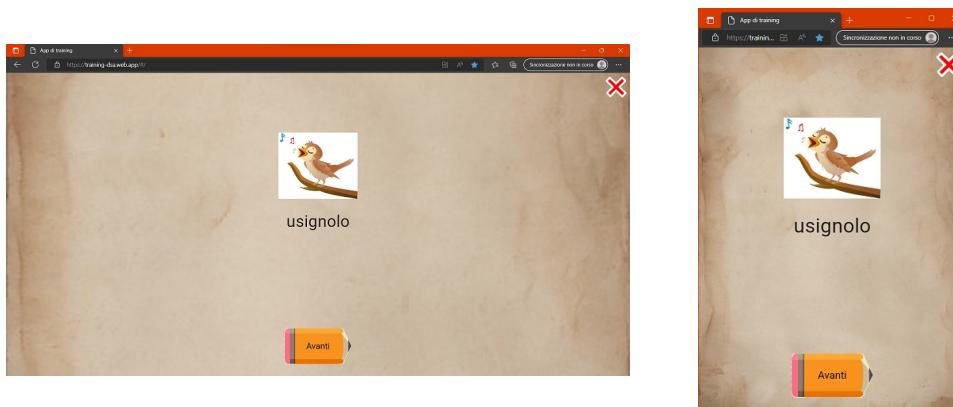
Le immagini fanno riferimento solo ad Edge, in quanto non esiste differenza con gli altri browser.



**Figura 5.2:** Schermate della pagina “Medaglie” con dimensioni diverse della finestra



**Figura 5.3:** Schermate della pagina dei livelli con dimensioni diverse della finestra



**Figura 5.4:** Schermate della fase di lettura della parola dell’allenamento con dimensioni diverse della finestra

### MacOS

MacOS (figura 5.5) è il sistema operativo sviluppato da Apple per i computer Macintosh, prodotti dalla stessa azienda. È il secondo sistema operativo più diffuso per computer dopo Windows.



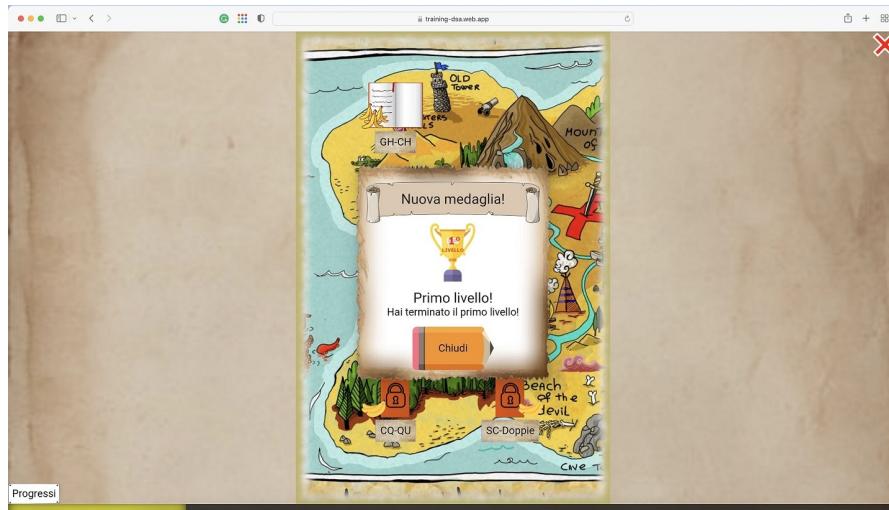
**Figura 5.5:** Logo MacOS

Per i test ho utilizzato i browser Safari e Chrome in un MacBook Pro. La grafica, così come in Windows, non presenta problemi: il comportamento è esattamente lo stesso. L'unico problema riscontrato e non risolto, per motivi di tempo, riguarda l'esecuzione dei file audio nel browser Safari. Il MacBook Pro presenta una barra touch screen, il cui nome è *touch bar*, visibile nella figura 5.6 che, posta sopra la tastiera, permette di regolare la riproduzione dei media.

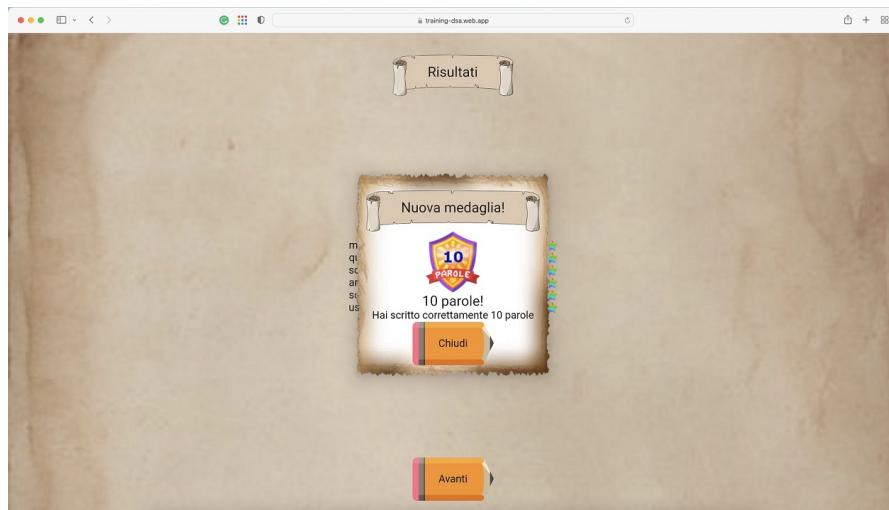


**Figura 5.6:** Touch bar di un MacBook Pro

La touch bar mostra la barra di riproduzione dei media, inclusi gli audio utilizzati nella web app. L'utente può interagire con tale barra per modificare la riproduzione dell'audio: metterlo in pausa, eseguire la riproduzione da un punto specifico, terminarla. Questo non è corretto dal punto di vista dei requisiti di “The Mind’s Ear and Eye Training”, perché ne compromette lo svolgimento. Un bambino, infatti, può saltare la riproduzione dei file audio senza che l'applicazione web se ne accorga, procedendo alla fase successiva dell'allenamento senza aver completato come dovrebbe quella corrente. La touch bar è una funzionalità esclusiva di Safari. Con altri browser, ad esempio Chrome, essa non appare e lo svolgimento degli esercizi non può essere compromesso. Riportiamo di seguito, nelle figure 5.7 e 5.8 alcuni screenshot della web app eseguita su MacOS col browser Safari.



**Figura 5.7:** Schermata scelta del livello con messaggio di una nuova medaglia in Safari



**Figura 5.8:** Schermata risultati allenamento con messaggio di una nuova medaglia in Safari

## Linux

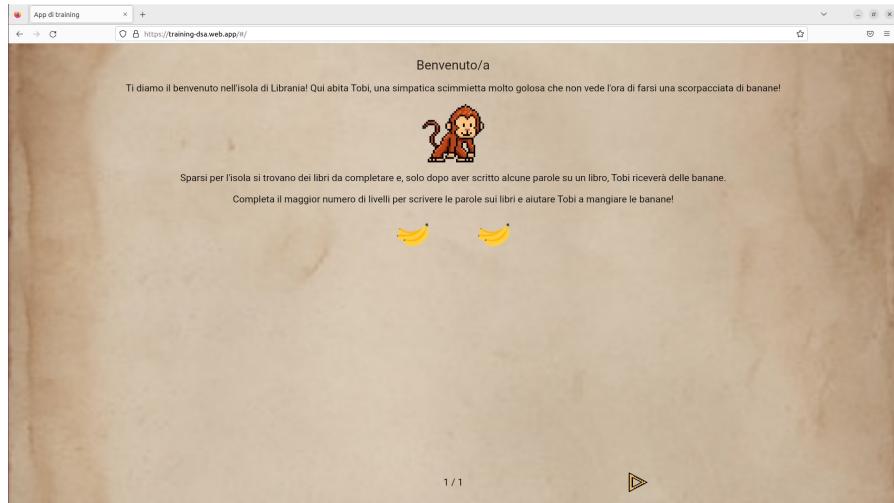
Linux è alla base di molti sistemi operativi. Ho scelto di testare Librania sulla distribuzione più comune: Ubuntu ([5.9](#)); nello specifico la versione 22.10.



**Figura 5.9:** Logo Ubuntu

Il browser utilizzato in Ubuntu è Firefox. Esattamente come per Windows, nulla in particolare deve essere riportato.

Inseriamo di seguito, per completezza, alcuni screenshot nelle figure 5.10, 5.11, e 5.12, che verificano l'adattamento dell'applicazione web a dimensioni diverse della finestra.



**Figura 5.10:** Tutorial iniziale



**Figura 5.11:** Tutorial iniziale  
(finestra ridotta)



**Figura 5.12:** Medaglia di benvenuto  
(finestra ridotta)

### Android

Android (figura 5.13) è un sistema operativo sviluppato da Google per dispositivi mobili. Progettato principalmente per smartphone e tablet, è disponibile anche per televisioni e dispositivi indossabili. È il sistema operativo mobile più diffuso al mondo.



**Figura 5.13:** Logo Android

In Android, ho testato Librania sia in versione web app sia in versione nativa in smartphone e tablet.

Librania è stata testata in diverse versioni di Android, dalla 5.0 alla 12, coprendo la grande maggioranza dei dispositivi attualmente utilizzati.

L’interfaccia si adatta a diverse dimensioni dello schermo. In uno smartphone, tuttavia, in modalità landscape, questa non sempre si adatta correttamente. L’applicazione è stata pensata principalmente per uno schermo in modalità portrait, infatti la versione apk impedisce la rotazione dello schermo, ma la versione in browser non permette ciò. Tuttavia, utilizzando un dispositivo con dimensioni dello schermo maggiori, come un tablet, il problema non si pone e la grafica si adatta orizzontalmente come se si trovasse nella finestra di un PC.

Riportiamo di seguito, in figure 5.14, 5.15, 5.16, e 5.17, alcuni screenshot presi da versioni diverse di Android, considerando sia l’esecuzione nativa sia tramite web.



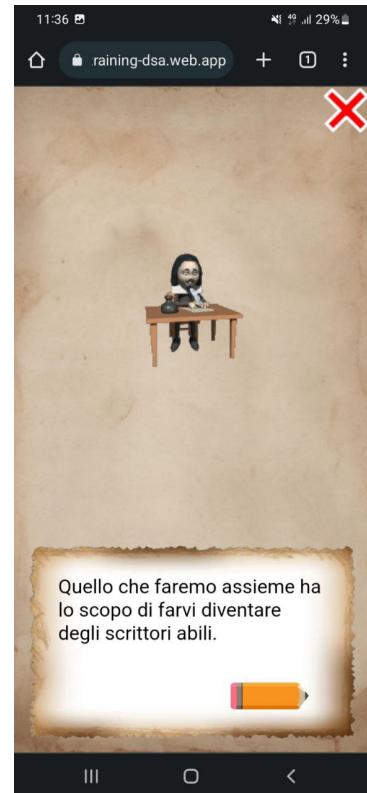
**Figura 5.14:** Android 5.0  
versione nativa



**Figura 5.15:** Android 5.0  
versione web



**Figura 5.16:** Android 12  
versione nativa



**Figura 5.17:** Android 12  
versione web

## iOS

iOS (figura 5.18), sviluppato da Apple, è il sistema operativo per iPhone, gli smartphone prodotti dalla stessa azienda. È il secondo sistema operativo mobile più diffuso al mondo.

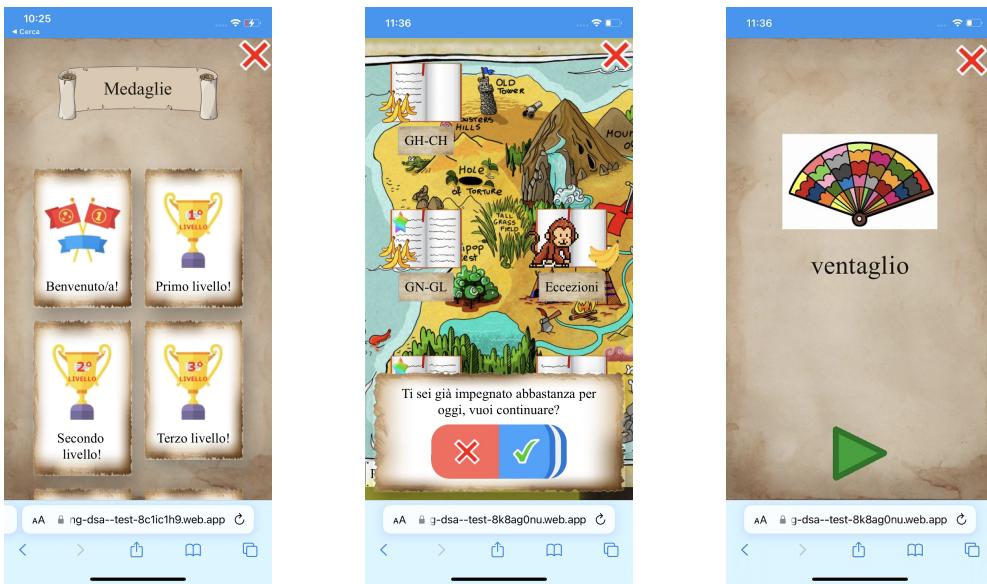


**Figura 5.18:** Logo iOS

I test sono stati effettuati con un iPhone contenente iOS 15.5. Non avendo previsto al momento una versione nativa per iOS, il test è avvenuto solo tramite browser. Come accade nei browser in Android, in modalità portrait l'interfaccia non presenta problemi, ma in landscape non riesce ad adattarsi se la dimensione dello schermo non è sufficientemente grande.

Riguardo il funzionamento della web app in sé ho riscontrato che i file audio non sempre venivano riprodotti: le istruzioni iniziali ([Istruzioni](#)) venivano lette, al contrario invece delle parole durante l'allenamento. Quando un utente iniziava l'allenamento, il file audio contenente la parola non veniva riprodotto e la pagina rimaneva in attesa della fine della riproduzione di un audio che non veniva riprodotto. Il problema consisteva

nel fatto che è necessaria un'interazione dell'utente affinché un media venga riprodotto, questo per evitare l'esecuzione automatica di media indesiderati. Per conformarsi a questa precauzione, in iOS le parole non vengono lette in automatico al caricamento della pagina, ma appare invece un pulsante che l'utente deve premere per permettere la riproduzione dell'audio (ultima immagine a destra della figura 5.19). Il pulsante dunque appare nella fase di lettura della parola, di spelling, e di lettura per sillabe. In figura 5.19 si possono vedere alcune schermate della web app in esecuzione nel sistema operativo iOS.



**Figura 5.19:** Schermate web app in browser Safari in dispositivo iPhone

## 5.2 Test di efficacia

Con test di efficacia si intende verificare il raggiungimento dell'obiettivo del prototipo: vale a dire se riesce ad aiutare veramente bambini con difficoltà nell'imparare a leggere e a scrivere.

A questo scopo sono stati effettuati dei test durante lo svolgimento della tesi da parte di alcune tirocinanti della professoressa Arfè con dei bambini e delle bambine. Il prototipo al momento sviluppato era carente di alcune funzionalità e l'interfaccia non era definitiva, ma è comunque risultata apprezzata dai bambini.

Per definire la reale efficacia della web app, è però necessario testare la sua versione attuale con un gruppo di bambini per un periodo prolungato. Durante questo tempo è possibile verificare l'usabilità e la gradevolezza nell'uso dell'applicazione web (l'ultimo livello della piramide di Maslow applicata all'esperienza utente di uno strumento informatico), nonché la capacità di ridurre il numero di errori commessi dai bambini nella lettura e nella scrittura delle parole ivi contenute.

Durante il periodo impiegato per la stesura di questa tesi, non sono stati condotti dei veri e propri test con gruppi di bambini, quindi purtroppo non sono disponibili dati che avvalorino il funzionamento del prototipo.

# Capitolo 6

## Conclusioni e sviluppi futuri

### 6.1 Conclusioni

Lo scopo della nostra tesi è la digitalizzazione del metodo “The Mind’s Ear and Eye Training”, utilizzato con i bambini che presentano difficoltà nella lettura e nella scrittura. Tradizionalmente il metodo si svolge in presenza, con una figura professionista che segue il bambino nelle varie fasi della procedura. Il metodo prevede la lettura di una parola, il suo spelling, e la lettura per sillabe. Generalmente, in queste tre fasi, la parola viene scritta su una lavagna dal professionista oppure viene proiettata. Successivamente il bambino cerca di memorizzare visivamente la parola, per poi scriverla sul proprio quaderno.

L’approccio voluto in questa tesi è diverso: non è presente una figura professionale che supervisiona il bambino, la parola non viene scritta su una lavagna o proiettata, e il bambino non la riscrive a mano. Tutto avviene su schermo tramite applicazione web: le parole sono “proiettate” sullo schermo e il bambino le scrive tramite una tastiera. L’applicazione web non deve essere vista dai suoi utenti come l’ennesimo compito per casa o come esercizi aggiuntivi da svolgere, ma dovrebbe invece presentarsi come un momento stimolante e divertente. A questo scopo sono stati inseriti elementi tipici dei videogiochi, che tentano di catturare l’interesse e l’entusiasmo dei bambini, nella speranza essi vivano l’esperienza più come un gioco che come uno strumento di potenziamento delle proprie abilità.

La sintesi di questi propositi è Librania: un’applicazione web sviluppata in Flutter che implementa una versione digitale del metodo “The Mind’s Ear and Eye Training”, adattata dalla professoressa Arfè, immersa in un’ambientazione ludica con funzionalità e caratteristiche tipiche dei videogiochi.

Durante lo svolgimento della tesi ho riscontrato alcune difficoltà riguardanti lo sviluppo con Flutter, in particolare con la gestione dell’interfaccia.

Questo è il primo progetto che sviluppo in Dart e utilizzando Flutter. Confrontandolo con linguaggi di programmazione e framework a cui sono solito fare affidamento, ammetto di essermi trovato leggermente in difficoltà soprattutto per quanto concerne la creazione e la gestione dell’interfaccia grafica. Abituato a separare il “comportamento” di un’applicazione dalla sua interfaccia, inizialmente mi sono trovato un po’ disorientato dalla gestione delle due componenti messa in atto da Flutter.

Nella creazione di applicazioni web, la mia esperienza consisteva solo nello sviluppo tramite HTML, Javascript, e CSS, in cui il linguaggio HTML costruisce la struttura

della pagina, il CSS crea lo stile, e Javascript definisce il comportamento. Nella creazione di applicazioni mobile, oltre a framework che permettono una base di codice in HTML, Javascript, e CSS, come ad esempio Cordova e Ionic, ho avuto esperienza di Android Studio, nel quale la struttura e lo stile dell’interfaccia sono contenuti in file XML, separati dal codice Java.

Flutter, al contrario, mantiene comportamento, struttura, e stile assieme, incastonando ogni oggetto grafico ed ogni stile in una gerarchia di Widget accompagnata, dove necessario, da funzioni che ne determinano il comportamento. Lo stile di programmazione risulta essere quindi molto diverso.

A mio parere, Flutter è un buon strumento nel caso in cui si vogliano sviluppare applicazioni per diversi dispositivi, ma se l’interfaccia è complessa può risultare difficile gestirla nel migliore dei modi.

In base alla natura dell’applicazione, esistono strumenti che possono agevolarne lo sviluppo. Per i videogiochi, ad esempio, esiste Flame [9], consigliato dagli stessi sviluppatori di Flutter:

*“Flame is a modular Flutter game engine that provides a complete set of out-of-the-way solutions for games. It takes advantage of the powerful infrastructure provided by Flutter but simplifies the code you need to build your projects.”*

Flame può essere integrato in un progetto Flutter come semplice plugin, quindi la sua installazione è molto semplice. Presenta strumenti che rendono semplice la gestione di immagini, animazioni, e audio. Inoltre gli sviluppatori consigliano anche altri plugin che, orchestrati con Flame, rendono la progettazione e lo sviluppo ancora più efficienti ed efficaci.

## 6.2 Sviluppi futuri

La tesi consiste nello sviluppo di un prototipo e le funzionalità che possono essere aggiunte sono numerose. Proprio con questa consapevolezza, Librania è stata progettata in modo da essere espandibile con nuove feature. Nella sezione corrente, forniamo un elenco di quali possono essere eventuali nuove funzionalità.

### **Aumentare numero di parole, frasi, e messaggi**

L’applicazione web è stata progettata in modo che possano essere inserite facilmente nuove parole, famiglie, e livelli. Se lo si ritiene opportuno, questa dovrebbe essere un’operazione semplice da attuare: è sufficiente aggiungere le nuove parole nel database online, con le relative informazioni, immagine e audio, e Librania aggiorna i suoi contenuti automaticamente.

Altro contenuto che potrebbe essere ampliato sono le frasi con cui l’applicazione web si complimenta per i risultati ottenuti o incoraggia l’utente nel caso in cui l’esercizio non sia andato a buon fine. Attualmente nel prototipo il numero di frasi è molto limitato, ma potrebbe essere facilmente ampliato.

Come riportato nella sezione [Allenamento](#) e in figura 4.33, a volte appare un messaggio che ricorda ai bambini passaggi fondamentali del “The Mind’s Ear and Eye Training” o che li incentiva ad esercitarsi. Nel prototipo ora è presente un solo messaggio a scopo dimostrativo, ma perché tale funzionalità risulti efficace dovrebbe contenerne degli altri. La progettazione di Librania prevede già l’aggiunta di nuovi messaggi: è

sufficiente inserire i nuovi messaggi (contenenti sia testo che immagini) in una lista già presente nell'applicazione, lista che ora contiene solo il messaggio di figura 4.33.

### Verifica generalizzazione

Come accennato precedentemente nella sezione [The Mind's Ear and Eye Training](#), esistono due tipi di apprendimento e quello che stiamo adottando è l'apprendimento implicito. I bambini quindi imparano i processi di lettura e scrittura tramite degli esempi che forniamo loro e tramite lo svolgimento dei livelli e delle sfide verificano la loro capacità di leggere e scrivere quelle stesse parole. Ma per capire se hanno veramente assorbito il meccanismo che li porta a leggere e scrivere, ed escludere che il loro successo dipenda esclusivamente dalla memoria, è necessario che vengano messi alla prova con parole nuove, con le quali non si sono mai esercitati. In questo modo verifichiamo la *generalizzazione* dell'apprendimento.

L'applicazione, quindi, potrebbe contenere due sezioni: una che applica l'apprendimento implicito e una che verifica la generalizzazione. Per l'apprendimento implicito i bambini si esercitano con i gruppi di parole suddivisi per livello esattamente come accade ora. La verifica della generalizzazione, invece, potrebbe essere contenuta nelle sfide dove, al contrario dello stato attuale, le parole presentate saranno diverse da quelle dell'addestramento e i bambini proveranno a mettere in pratica autonomamente il "The Mind's Ear and Eye Training".

### Parole selezionate in base a età / classe frequentata

Nel database, per ogni utente è memorizzata anche l'età. Al momento questa informazione non è importante, ma potrebbe essere utilizzata per selezionare le parole più adatte da presentare. Forse, ancora meglio dell'età potrebbe essere memorizzata la classe frequentata, in modo da tener conto delle capacità dell'utente. In ogni caso, un campo per questo scopo è già presente.

### Chiedere conferma del significato di una parola

Come descritto in [Lettura parola](#), quando un utente afferma di conoscere il significato di una parola, viene memorizzato il timestamp di quell'istante. Salvare un booleano sarebbe più semplice, richiederebbe meno memoria, e allo stato attuale dell'applicazione web sarebbe sufficiente. Ho scelto tuttavia il timestamp perché in questo modo è possibile far sì che, dopo un tempo stabilito, si riproponga all'utente la domanda "Conosci questa parola?".

### Timestamp relativo a ciascun livello

Tra i progressi degli utenti, in [LevelProgress](#), al momento viene memorizzato anche il timestamp nel quale ciascun livello viene sbloccato. Se lo si ritiene opportuno, il timestamp potrebbe essere aggiornato ogni volta che il bambino termina il livello. Questa potrebbe essere un'informazione utile a chi monitora i progressi dei bambini.

### Audio e animazioni

In ogni videogioco che si rispetti non possono mancare gli effetti sonori. Per questioni di priorità, il prototipo ne è sprovvisto, ma molto probabilmente contribuirebbero a migliorare l'esperienza degli utenti.

Alcuni esempi di effetti sonori potrebbero essere:

- pressione dei pulsanti;
- cambio schermata;
- sottofondo musicale in tema con la grafica e l'ambientazione (fatta eccezione per l'allenamento e per le sfide, in quanto una musica di sottofondo potrebbe distrarre l'utente e compromettere l'efficacia del “The Mind's Ear and Eye Training”);
- dopo la scrittura di una parola da parte dell'utente, per la buona o cattiva riuscita dell'esercizio;
- espressioni della mascotte, magari accompagnate anche da animazioni.

Anche le animazioni possono rendere più gradevole l'esperienza utente. Seppur molto poche, Librania ne contiene, come nella pressione dei pulsanti, nell'apparizione delle finestrelle in primo piano (ad esempio come quelle in figura 4.33), la mano che segue la lettura delle sillabe, e al termine della scrittura di una parola.

Altre animazioni che potrebbero essere inserite sono:

- mascotte non visualizzata come immagine statica;
- spostamento della mascotte da un livello al successivo quando questo viene sbloccato;
- mascotte che mangia le banane al termine di un livello;
- nella pagina dei risultati, le nuove parole scritte correttamente dall'utente vengono scritte in un libro (riferendosi al libro presente nella pagina della selezione dei livelli).

Aggiungere troppe animazioni però potrebbe essere controproducente, è necessario infatti porre attenzione affinché non appesantiscono troppo l'esecuzione dell'applicazione web. In tal caso, se le performance dell'applicazione dovessero risentirne, verrebbe a mancare una componente base della piramide di Maslow: l'usabilità. È accaduto, ad esempio, che l'animazione dell'emoji al termine della scrittura di una parola impegnasse troppe risorse (era eseguita in browser su mobile) e, a causa di ciò, il pulsante per procedere con l'allenamento non fosse reattivo e non sempre recepiva la pressione. Si è reso necessario quindi diminuire la qualità dell'animazione. Visivamente la differenza con la qualità maggiore non è significativa, ma le performance sono nettamente migliorate.

L'aggiunta di suoni e animazioni, quindi, deve essere ben bilanciata e deve tener conto di tutti i dispositivi sui quali Librania può essere eseguita.

### **Adattamento istruzioni**

Le istruzioni utilizzate ora all'inizio dell'allenamento sono tratte dalla tesi *Le abilità ortografiche nella scuola primaria: Il training ortografico in italiano e in inglese* di Zancato Tamara [2], a pagina 57. Tali istruzioni però sono specifiche per sessioni di training di una classe in presenza:

*“Quello che faremo assieme ha lo scopo di farvi diventare degli scrittori abili. Quando scriviamo generalmente, anche se non ce ne rendiamo conto, “leggiamo” le parole dalla nostra mente. Il lavoro che faremo assieme sarà quello di costruire una “libreria mentale”, ricca di parole qualche volta nuove o qualche volta un po’ più*

*difficili del solito. Per diventare scrittori abili è necessario, però, allenarsi a diventare anche bravi fotografi: per questo, il compito che vi darò sarà quello di fotografare le parole che scriverò alla lavagna con l'occhio che c'è nella vostra mente, cercando di mantenere l'immagine scattata nella vostra testa e, poi, di provare a scriverle usando quelle immagini.”*

L'approccio utilizzato nel nostro prototipo è leggermente diverso e le istruzioni dovrebbero essere adattate di conseguenza.

### Maggiori interazioni con la mascotte

Tobi, la mascotte di Librania, potrebbe essere più presente ed interagire maggiormente con gli utenti. Potrebbe ad esempio comparire durante le prime esecuzioni dell'allenamento per ricordare ai bambini, in diverse schermate, in cosa consistono le varie fasi. Potrebbe apparire nelle schermate dell'applicazione, ad esempio nell'angolo in basso a sinistra, dando delle informazioni sulla pagina corrente, in modo non invasivo, e reagire con delle animazioni (ad esempio cambiando espressione) se l'utente clicca o preme su di essa.

### Aumentare il numero e la complessità delle medaglie

Le medaglie presenti ora nel prototipo hanno obiettivi semplici da verificare a livello di codice, in quanto lo scopo ora è verificarne la fattibilità. Sarebbe più accattivante l'inserimento di obiettivi più complessi da raggiungere, che considerino ad esempio anche una continuità dell'uso dell'applicazione web o una costanza dei risultati ottenuti.

### Impostazioni

Inizialmente erano previste delle impostazioni che, per motivi di priorità, non sono più state aggiunte:

- **font:** dare la possibilità di cambiare il font dei caratteri utilizzati per le parole durante l'allenamento e le sfide. Ad esempio utilizzare un font più adatto a bambini con dislessia, oppure utilizzare il maiuscolo o il corsivo per familiarizzare anche con altri tipi di scrittura utilizzati quotidianamente. Un esempio a puro scopo dimostrativo si trova in figura 6.1.

Anche questa modifica non risulta complessa. Il codice dell'applicazione web contiene una classe che prepara il widget contenente la parola visualizzata durante le fasi di lettura, modificando lo spazio tra le lettere per lo spelling e lo sfondo dei caratteri per la lettura in sillabe. È sufficiente aggiungere i font ritenuti più adeguati tra gli asset del progetto e modificare nella classe appena citata il valore del parametro riguardante il font, nello specifico il parametro `fontFamily`, previsto e già presente nel codice;

- **lingua inglese:** prendendo spunto da *Language-Specific Effects in Response to Spelling Intervention in Italian and in English as an Additional Language* di Barbara Arfè e Tamara Zancato [3], sarebbe interessante testare la versione digitale di “The Mind’s Ear and Eye Training” anche su parole di lingua inglese. A questo scopo le modifiche da apportare sono più complesse, in quanto si dovrebbe distinguere nel database le parole in lingue diverse ed eventualmente diversificare anche le classifiche. Molto probabilmente le famiglie e i livelli saranno distinti e, con queste premesse, l'interfaccia potrebbe subire pesanti modifiche, soprattutto nella schermata della scelta del livello.



**Figura 6.1:** Esempio dimostrativo dell'utilizzo di un font diverso

### 6.3 Potenzialità e possibile evoluzione dell'applicazione

I dati che possono essere raccolti da Librania sono davvero molti e potrebbero essere utili ad esempio per valutare l'efficacia dell'applicazione web, del metodo utilizzato, e dell'impegno dei bambini.

Una persona che supervisiona il percorso dei bambini e con accesso a tutti i dati, potrebbe farsi un'idea del quadro generale delle capacità di ciascun utente o di un gruppo di utenti (parliamo ad esempio della classe di una scuola). Uno strumento in grado di accedere a tali dati e di visualizzarli in modo funzionale potrebbe, a mio parere, dare un grande contributo perché permetterebbe di capire quali sono i “punti deboli” e le difficoltà negli allenamenti per poter agire di conseguenza.

Librania, quindi, potrebbe essere affiancata da un altro strumento, disponibile a chi segue l'insegnamento dei bambini, per monitorare i loro percorsi. Tutti i dati raccolti in questo stadio del prototipo e quelli che l'applicazione web è già pronta per raccogliere, costituiscono una base dati utilizzabile.

Vediamo di seguito alcuni esempi di informazioni che possono essere derivate dai dati raccolti.

### **Monitoraggio durata allenamento**

Tra le informazioni memorizzate riguardo i progressi con le parole, oltre al numero di volte in cui ciascuna parola viene scritta correttamente o in modo errato, si potrebbe memorizzare anche il tempo che l'utente impiega per svolgere l'allenamento di ciascuna parola. Questa informazione potrebbe essere indice di quanto ciascuna parola risulti difficile per il bambino, anche se la scrive correttamente. Se invece le parole vengono scritte sbagliate e il tempo impiegato è al di sotto di una certa soglia, potrebbe significare che il bambino non si stia impegnando adeguatamente. Inoltre, avere uno storico di tali durate, potrebbe aiutare a capire la confidenza che l'utente ha con determinate parole, se scritte correttamente, e i miglioramenti nel tempo.

### **Monitoraggio livelli giocati**

Il componente **TrainingManager**, descritto nella sezione [TrainingManager](#), memorizza dati che nel prototipo non vengono utilizzati, ma che potrebbero risultare utili a fini di analisi. Nello specifico, i dati memorizzati e non ancora utilizzati sono:

- lista di parole scritte correttamente;
- lista di parole scritte in modo sbagliato;
- livelli giocati.

Le informazioni sopra riportate sono memorizzate giorno per giorno. Questo vuol dire che in Firebase, nella sezione dei progressi di ogni utente, si trova un documento per ogni giorno di allenamento che contiene le tre liste.

Da questo si può dedurre, ad esempio, quanto un bambino si esercita ogni giorno, se durante la stessa giornata affronta livelli diversi o si concentra su alcuni in particolare, e qual è la sua performance media. Con quest'ultima conoscenza, se un giorno dovesse commettere molti più errori rispetto al suo andamento medio, i motivi potrebbero essere esterni e non dipendenti dagli esercizi.



## Appendice A

# Parole utilizzate e informazioni correlate

Riportiamo nell'appendice A le informazioni riguardanti le parole, le famiglie, e i livelli presenti in Librania.

Nella tabella A.1 le colonne contengono rispettivamente:

- le parole in ordine alfabetico;
- la divisione in sillabe delle parole (ricordiamo che la sillabazione non è quella classica, in quanto le lettere doppie non vengono divise);
- la famiglia, o famiglie, di appartenenza della parola.

**Tabella A.1:** Lista parole

Parola	Sillabazione	Famiglia di appartenenza
acciughe	a - cciu - ghe	GH
accogliere	a - cco - glie - re	GL
acquazzone	a - cqua - zzo - ne	CQ-QU
acquitrino	a - cqui - tri - no	CQ-QU
acquolina	a - cquo - li - na	CQ-QU
aiuola	a - iuo - la	Eccezioni
allievo	a - llie - vo	Eccezioni
annacquare	a - nna - cqua - re	CQ-QU

aquilone	a - qui - lo - ne	Eccezioni
artiglieria	ar - ti - glie - ri - a	GL
ascensore	a - scen - so - re	SC
assaggiare	a - ssa - ggia - re	Doppie
cassettiera	ca - sse - ttie - ra	Doppie
chiacchierare	chia - cchie - ra - re	CH
chilogrammo	chi - lo - gra - mmo	CH
ciliegia	ci - lie - gia	Eccezioni
cinghiale	cin - ghia - le	GH
cinquanta	cin - quan - ta	CQ-QU
compagna	com - pa - gna	GN
coniglio	co - ni - glio	GL
cuocere	cuo - ce - re	Eccezioni
disegno	di - se - gno	GN
eschimese	e - schi - me - se	CH
falegname	fa - le - gna - me	GN
familiare	fa - mi - lia - re	Eccezioni
germogliare	ger - mo - glia - re	GL
ghiacciolo	ghia - ccio - lo	GH
ghirigoro	ghi - ri - go - ro	GH
ghirlanda	ghir - lan - da	GH
giubbotto	giu - bbo - tto	Doppie
groviglio	gro - vi - glio	GL
igiene	i - gie - ne	Eccezioni
inchino	in - chi - no	CH

ingegnoso	in - ge - gno - so	GN
liquirizia	li - qui - ri - zia	CQ-QU
lucchetto	lu - cche - tto	CH
luccicante	lu - cci - can - te	Doppie
magnete	ma - gne - te	GN
margherita	mar - ghe - ri - ta	GH
migliorare	mi - glio - ra - re	GL
moscerino	mo - sce - ri - no	SC
obliquo	o - bli - quo	CQ-QU
orecchino	o - re - cchi - no	CH
padiglione	pa - di - glio - ne	GL
parrucchiere	pa - rru - cchie - re	CH
passerotto	pa - sse - ro - tto	Doppie
pastasciutta	pa - sta - sciu - tta	SC
percuotere	per - cuo - te - re	Eccezioni
pesciolino	pe - scio - li - no	SC
quadrifoglio	qua - dri - fo - glio	GL, CQ-QU
ragnatela	ra - gna - te - la	GN
righello	ri - ghe - llo	GH
ringhiera	rin - ghie - ra	GH
risciacquare	ri - scia - cqua - re	CQ-QU
riunione	riu - nio - ne	CQ-QU
scherzare	scher - za - re	CH
sciabola	scia - bo - la	SC
sciatore	sci - a - to - re	SC

scienziato	scien - zia - to	SC
sciopero	scio - pe - ro	SC
sciropo	sci - ro - ppo	SC
scogliera	sco - glie - ra	GL
scricchiolare	scri - cchio - la - re	CH
segnalibro	se - gna - li - bro	GN
serratura	se - rra - tu - ra	Doppie
sgusciare	sgu - scia - re	SC
singhiozzo	sin - ghio - zzo	GH
sognare	so - gna - re	GN
soquadro	so - qqua - dro	CQ-QU
sostegno	so - ste - gno	GN
sotterrare	so - tte - rra - re	Doppie
spadaccino	spa - da - cci - no	Doppie
spaghetti	spa - ghe - tti	GH
sussurrare	su - ssu - rra - re	Doppie
tacchino	ta - cchi - no	CH
usignolo	u - si - gno - lo	GN
ventaglio	ven - ta - glio	GL
zattera	za - tte - ra	Doppie

Riportiamo ora i livelli dell'applicazione e quali sono le famiglie che li compongono:

- **Livello 1:**

GH, CH;

- **Livello 2:**

GN, GL;

- **Livello 3:**

CQ-QU;

- **Livello 4:**

SC, Doppie;

- **Livello 5:**

Eccezioni.

Come ausilio all'elenco appena riportato, includiamo una lista in cui sono riportate le parole suddivise in base alla famiglia di appartenenza, in modo da fornire un elenco consultabile più facilmente per capire quali parole sono presenti in un determinato livello:

- **GH:**

acciughe, ghiacciolo, margherita, spaghetti, ghirigoro, cinghiale, righello, ringhiera, singhiozzo, ghirlanda;

- **CH:**

scherzare, parrucchiere, lucchetto, orecchino, inchino, tacchino, eschimese, chiacchierare, scricchiolare, chilogrammo;

- **GN:**

ragnatela, disegno, ingegnoso, sostegno, magnete, sognare, compagna, segnalibro, usignolo, falegname;

- **GL:**

scogliera, quadrifoglio, artiglieria, coniglio, padiglione, accogliere, germogliare, groviglio, ventaglio, migliorare;

- **CQ-QU:**

annacquare, acquitrino, acquazzone, risciacquare, obliquo, quadrifoglio, liquirizia, cinquanta, acquolina, soquadro;

- **SC:**

pastasciutta, scienziato, sciopero, moscerino, sciabola, sciropo, sgusciare, pesciolino, sciatore, ascensore;

- **Doppie:**

serratura, sussurrare, assaggiare, cassetiera, passerotto, sotterrare, giubbotto, zattera, luccicante, spadaccino;

- **Eccezioni:**

taccuino, igiene, aquilone, ciliegia, cuocere, aiuola, familiare, percuotere, riunione, allievo;



## Appendice B

# Informazioni di input per il database

*In questa appendice riportiamo i passaggi da svolgere per inserire nuovi utenti e il contenuto dei file usati come input per immettere in Firebase le informazioni relative a parole, famiglie, e livelli.*

### B.1 Utenti

Per inserire un nuovo utente è necessario compiere due passaggi:

1. inserire le credenziali nella sezione *Authentication* di Firebase;
2. aggiungere a Firestore Database le informazioni necessarie a Librania.

Il primo passaggio si esegue tramite l’interfaccia di Firebase. L’amministratore di Librania, o chi si occupa dei dati, deve inserire nella sezione *Authentication* di Firebase l’email del nuovo utente e la sua password. L'email non è necessaria per altri scopi ed è composta dal nome utente seguito dalla stringa “@trainingapp.it”.

Il secondo passaggio può essere svolto a mano, inizializzando i dati come descritto nella sezione [Child](#). Per comodità, ho scritto uno script che avendo come input un file contenente le generalità dell’utente, prepara il documento nel database. Le informazioni di ogni utente sono suddivise in quattro righe contenenti, in ordine:

1. username;
2. nome;
3. cognome;
4. età.

Un esempio è:

```
vera.rossi  
Vera  
Rossi  
8
```

## B.2 Parole

Il file contenente le informazioni delle parole contiene quattro righe per ogni parola ed esse corrispondono, in ordine, a:

1. parola;
2. sillabe separate da una virgola;
3. intervalli di tempo per la lettura delle sillabe separate da una virgola;
4. significato della parola.

Tra le quattro righe di una parola e la successiva si trova una riga vuota.

Di seguito è riportato il contenuto del file.

```

acciughe
a,cciu,ghe
700,1700,2700
Piccolo pesce commestibile di acqua salata.

ghiacciolo
ghia,ccio,lo
500,1600,2600
Gelato a base di ghiaccio colorato, arricchito con aromi vari.

margherita
mar,ghe,ri,ta
700,1750,2600,3700
Pianta dai fiori gialli o bianchi.

spaghetti
spa,ghe,tti
1000,2450,3466
Pasta lunga e sottile che si lessa in acqua e poi si condisce in
vari modi.

ghirigoro
ghi,ri,go,ro
600,1720,2600,3600
Disegno formato da un intreccio casuale di linee curve.

cinghiale
cin,ghia,le
700,1800,2800
Mammifero selvatico simile al maiale, con pelo ruvido, coda corta e canini in fuori a forma di zanne.

righello
ri,ghe,llo
700,1780,2680

```

Piccolo strumento di materiale vario con i bordi graduati, usato per tracciare linee rette o per misurare.

ringhiera

rin, ghie, ra

700, 1820, 2860

Struttura di ferro, marmo o legno che fa da riparo sul vuoto in scale e terrazzi

singhiozzo

sin, gho, zzo

700, 2000, 2950

Contrazione involontaria del diaframma accompagnata da un caratteristico rumore

ghirlanda

ghir, lan, da

600, 1950, 3000

Corona di fiori e foglie intrecciati insieme per addobbo festivo

scherzare

scher, za, re

500, 2130, 2950

Comportarsi e parlare in modo spiritoso e vivace per divertirsi

parrucchiere

pa, rru, cchie, re

700, 2000, 3000, 4100

Chi per mestiere acconcia e taglia i capelli

lucchetto

lu, cche, tto

500, 1777, 2930

Serratura metallica mobile per il bloccaggio di porte, bauli etc.

orecchino

o, re, cchi, no

500, 1457, 2413, 3287

Piccolo elemento decorativo che si porta all'orecchio

inchino

in, chi, no

700, 1860, 2826

Atto di riverenza in cui si china il capo e il busto

tacchino

ta, cchi, no

600, 1780, 2688

Grosso uccello commestibile con testa e collo senza penne dal piu-

maggio bruno.

eschimese  
e,schi,me,se  
700,1692,2770,3750  
Abitante nativo delle regioni artiche dell'America

chiacchierare  
chia,cchie,ra,re  
900,2000,3000,4000  
Conversare per passatempo su cose piacevoli o poco importanti

scricchiolare  
scri,cchio,la,re  
1000,2632,3940,5000  
Produrre rumori secchi e leggeri nel rompersi

chilogrammo  
chi,lo,gra,mmo  
1000,2200,3200,4200  
Unità di misura del peso uguale a mille grammi

annacquare  
a,nna,cqua,re  
1000,2000,3000,4250  
Allungare un liquido con acqua

acquitrino  
a,cqui,tri,no  
531,1483,2378,3127  
Terreno paludososo con ristagno

acquazzone  
a,cqua,zzo,ne  
800,2100,3230,4350  
Pioggia improvvisa battente ma di breve durata

risciacquare  
ri,scia,cqua,re  
500,1483,2500,3347  
Lavare con acqua pulita

obliquo  
o,bli,quo  
1000,1800,2870  
Inclinato rispetto al piano d'appoggio

quadrifoglio  
qua,dri,fo,glio  
700,1900,2900,3800

Pianta con quattro foglioline che viene considerata portafortuna per la sua rarità

liquirizia  
li, qui, ri, zia  
500, 1900, 2900, 3900

Pianta con radici legnose, usate per produrre farmaci e dolci

cinquanta  
cin, quan, ta  
1000, 2300, 3400

Numero formato da 5 decine e 10 unità

acquolina  
a, cquo, li, na  
1000, 2000, 3000, 4000

Salivazione abbondante dovuta al desiderio di un cibo appetitoso

soquadro  
so, qqua, dro  
700, 2000, 3200

Grande scompiglio confusione

ragnatela  
ra, gna, te, la  
531, 1753, 2852, 3598, 4508

Struttura di fili molto sottili costruita dal ragno per catturare la preda

disegno  
di, se, gno  
226, 1350, 2364, 3335

Rappresentazione per mezzo di linee e segni di immagini varie

ingegnoso  
in, ge, gno, so  
531, 1698, 2755, 3824, 4654

Fornito di ingegno, creativo e abile nel superare le difficoltà

sostegno  
so, ste, gno  
531, 1948, 3000, 4100

Aiuto materiale o morale che dà sicurezza.

Elemento che regge il peso di qualcosa, appoggio

magnete  
ma, gne, te  
531, 1820, 3000, 3900

Corpo che ha la capacità di attirare a sé oggetti di ferro

sognare

so, gna, re

531, 1832, 2779, 3634

Immagini che si presentano alla mente mentre dormiamo

compagna

com, pa, gna

531, 2187, 3176, 4312

Significato

segnalibro

se, gna, li, bro

531, 1643, 2779, 3732, 4648

Striscia di cartone leggero che si mette tra le pagine di un libro per ricordare dove si è arrivati nella lettura

usignolo

u, si, gno, lo

531, 1503, 2553, 3720, 4477

Uccello rossiccio che emette un canto melodioso

falegname

fa, le, gna, me

500, 1777, 2791, 3830, 4549

Persona che lavora il legno

pastasciutta

pa, sta, sciu, tta

595, 1598, 2781, 3781, 4655

Pasta di varia forma, cotta in acqua bollente, scolata e condita.

scienziato

scien, zia, to

500, 2200, 3514

Persona che si dedica a studi e ricerche in campo scientifico

sciopero

scio, pe, ro

600, 2068, 3036

Astensione dal lavoro organizzata come forma di protesta per ottenere migliori condizioni

moscerino

mo, sce, ri, no

10, 950, 2151, 3180

Insetto piccolissimo che vola

sciabola

scia, bo, la  
595, 1778, 2781, 3673  
Arma da taglio con una lunga lama più o meno curva

sciropo  
sci, ro, ppo  
500, 1700, 3000, 4000  
Soluzione zuccherina utilizzata come medicinale

sgusciare  
sgu, scia, re  
770, 2034, 3010  
Levare dal guscio

pesciolino  
pe, scio, li, no  
700, 1730, 2828, 3900  
Animale acquatico con pinne per nuotare e branchie per respirare di piccole dimensioni

sciatore  
sci, a, to, re  
1000, 2000, 3000, 4000  
Uomo che usa gli sci per andare sulla neve o sull'acqua

ascensore  
a, scen, so, re  
600, 1667, 3000, 4100  
Cabina che trasporta persone e cose da un piano all'altro di un edificio

serratura  
se, rra, tu, ra  
500, 1600, 2600, 3500  
Meccanismo di metallo per chiudere porte, finestre e simili

sussurrare  
su, ssu, rra, re  
1000, 2100, 3300, 4434  
Parlare a bassa voce

assaggiare  
a, ssa, ggia, re  
500, 1276, 2345, 3379  
Gustare cibo o bevanda in piccole quantità per sentire il sapore

cassettiera  
ca, sse, ttie, ra  
1000, 2178, 3667, 5000

Mobile dotato di cassetti sovrapposti

passerotto

pa,sse,ro,tto

560,1600,2899,4347

Piccolo uccello che si nutre di insetti e cereali

sotterrare

so,tte,rra,re

833,2474,3619,4892

Mettere sottoterra

giubbotto

giu,bbo,tto

663,1800,2900

Giacca corta e pesante

zattera

za,tte,ra

633,2146,3146

Piattaforma rettangolare galleggiante, usata per trasportare merci o persone

luccicante

lu,cci,can,te

1000,2222,3544,4940

Che riflette luce, scintillante

spadaccino

spa,da,cci,no

865,2346,3411,4654

Persona che sa usare la spada con abilità

taccuino

ta,ccui,no

1100,2480,3778

Quadernetto per prendere appunti

igiene

i,gie,ne

600,1600,2600

Insieme delle norme da seguire per salute personale e ambientale

aquilone

a,qui,lo,ne

1000,2000,3000,4000

Giocattolo costruito con carta o stoffa che si fa volare tirandolo controvento con un filo

ciliegia  
ci, lie, gia  
500, 1600, 2600  
Frutto tondeggiante e rosso

cuocere  
cuo, ce, re  
600, 1700, 2600  
Sottoporre qualcosa a cottura

aiuola  
a, iuo, la  
660, 1747, 3060  
Striscia di terreno in giardini o piazze coltivata a fiori e piantine ornamentali

familiare  
fa, mi, lia, re  
600, 1800, 2900, 4000  
Che riguarda la famiglia - persona che appartiene alla propria famiglia.

percuotere  
per, cuo, te, re  
600, 1900, 2900, 3900  
Dare colpi picchiare

riunione  
riu, nio, ne  
625, 1862, 2977  
Incontro, ritrovo di più persone

allievo  
a, llie, vo  
1000, 2229, 3700  
Chi frequenta un corso di studi o di addestramento

scogliera  
sco, glie, ra  
600, 2060, 3255  
Successione di scogli collocati a breve distanza l'uno dall'altro

artiglieria  
ar, ti, glie, ri, a  
500, 1600, 2500, 3500, 4500  
Insieme di armi pesanti

coniglio  
co, ni, glio

600, 1500, 2450

Mammifero o roditore domestico o selvaggio con folto pelo, orecchie lunghe e coda corta

padiglione

pa, di, glio, ne

500, 1670, 2520, 3473

Costruzione leggera situata in parchi o giardini, utilizzata come ritrovo in occasioni particolari

accogliere

a, cco, glie, re

520, 1650, 2500, 3500

Ricevere qualcuno, ospitare presso di sé

germogliare

ger, mo, glia, re

600, 2000, 2915, 3940

Crescita di una pianta da un seme

groviglio

gro, vi, glio

600, 2000, 2940

Ammasso di fili intrecciati

ventaglio

ven, ta, glio

700, 2188, 3000

Strumento formato da stecche unite nella parte inferiore, sulle quali è incollato un foglio di diverso materiale

migliorare

mi, glio, ra, re

500, 1500, 2500, 3500

Diventare migliore, cambiare in meglio

### B.3 Famiglie

La struttura del file contenente le informazioni riguardanti le famiglie è molto semplice. Ogni famiglia è composta da un insieme di righe e, tra le righe di una famiglia e la successiva, se ne trova una vuota.

In un insieme di righe corrispondenti ad una famiglia, la prima riga contiene il nome della famiglia, mentre le successive contengono le parole che compongono tale famiglia.

Riportiamo di seguito il contenuto del file.

GH

acciughe

ghiacciolo

margherita  
spaghetti  
ghirigoro  
cinghiale  
righello  
ringhiera  
singhiozzo  
ghirlanda

CH  
scherzare  
parrucchiere  
lucchetto  
orecchino  
inchino  
tacchino  
eschimese  
chiacchierare  
scricchiolare  
chilogrammo

GN  
ragnatela  
disegno  
ingegnoso  
sostegno  
magnete  
sognare  
compagna  
segnalibro  
usignolo  
falegname

GL  
scogliera  
quadrifoglio  
artiglieria  
coniglio  
padiglione  
accogliere  
germogliare  
groviglio  
ventaglio  
migliorare

CQ-QU  
annacquare  
acquitrino  
acquazzone  
risciacquare

obliquo  
quadrifoglio  
liquirizia  
cinquanta  
acquolina  
soqquadro

SC  
pastasciutta  
scienziato  
sciopero  
moscerino  
sciabola  
sciroppo  
sgusciare  
pesciolino  
sciatore  
ascensore

Doppie  
serratura  
sussurrare  
assaggiare  
cassettiera  
passerotto  
sotterrare  
giubbotto  
zattera  
luccicante  
spadaccino

Eccezioni  
taccuino  
igiene  
aquilone  
ciliegia  
cuocere  
aiuola  
familiare  
percuotere  
riunione  
allievo

## B.4 Livelli

Il file contenente le informazioni circa i livelli è il più breve.

Come nei file precedenti, le informazioni sono suddivise in insiemi di righe. Ogni insieme corrisponde ad un livello ed è separato dal successivo da una riga vuota.

Le righe di ciascun insieme contengono, in ordine:

1. numero del livello, che specifica l'ordinamento all'interno dell'applicazione web;
2. nome del livello;
3. famiglia compresa nel livello. Nel caso siano comprese più famiglie, esse si trovano in righe distinte.

Di seguito è riportato il contenuto del file. Come si può notare, solo i livelli “CQ-QU” ed “Eccezioni” sono composti da un'unica famiglia.

1  
GH-CH  
GH  
CH

2  
GN-GL  
GN  
GL

3  
CQ-QU  
CQ-QU

4  
SC-Doppie  
SC  
Doppie

5  
Eccezioni  
Eccezioni







# Bibliografia

- [1] OMBRETTA GAGGI, MATTEO CIMAN, *An empirical analysis of energy consumption of cross-platform frameworks for mobile development*, Department of Mathematics, University of Padua, Padua, Italy, 2016
- [2] TAMARA ZANCATO, *Le abilità ortografiche nella scuola primaria: Il training ortografico in italiano e in inglese*, Dipartimento di Psicologia dello Sviluppo e della Socializzazione, Università di Padova, Padova, Italia, 2018/2019
- [3] BARBARA ARFÉ, TAMARA ZANCATO, *Language-Specific Effects in Response to Spelling Intervention in Italian and in English as an Additional Language*, Dipartimento di Psicologia dello Sviluppo e della Socializzazione, Università di Padova, Padova, Italia
- [4] KAI HUOTARI, JUHO HAMARI, *Defining Gamification - A Service Marketing Perspective*
- [5] SEBASTIAN DETERDING ET AL., *Gamification: Toward a Definition*
- [6] ABT C., *Serious Games*, University Press of America, 1987
- [7] Growth Engineering, <https://www.growthengineering.co.uk/definition-of-gamification/>
- [8] Flutter - Documentazione, <https://docs.flutter.dev/>
- [9] Flame - Documentazione, <https://docs.flame-engine.org/1.4.0/>