



UNIVERSIDAD DE TALCA
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA EN DESARROLLO DE
VIDEOJUEGOS Y REALIDAD VIRTUAL

**Diseño y desarrollo de software educativo para el
entrenamiento fonológico de las letras S, C y Z, en
niños del primer ciclo básico.**

DIEGO ESTEBAN CONTRERAS COFRÉ

Profesor Guía: CAMILA ALIAGA

Profesor Co-guía: DR. JAIME BALLADARES H.

Memoria para optar al título de
Ingeniero en Desarrollo de Videojuegos y Realidad Virtual

Talca – Chile
Junio, 2025



UNIVERSIDAD DE TALCA
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA EN DESARROLLO DE
VIDEOJUEGOS Y REALIDAD VIRTUAL

**Diseño y desarrollo de software educativo para el
entrenamiento fonológico de las letras S, C y Z, en
niños del primer ciclo básico.**

DIEGO ESTEBAN CONTRERAS COFRÉ

Profesor Guía: CAMILA ALIAGA

Profesor Co-guía: DR. JAIME BALLADARES H.

Memoria para optar al título de
Ingeniero en Desarrollo de Videojuegos y Realidad Virtual

El presente documento fue calificado con nota: _____

Talca – Chile

Junio, 2025

*Dedicado a mi familia, amigos, compañeros de trabajos y todos/as las personas que
ayudaron a completar este escrito.*

AGRADECIMIENTOS

Agradecimientos a ...

TABLA DE CONTENIDOS

	página
Dedicatoria	I
Agradecimientos	II
Tabla de Contenidos	III
Índice de Figuras	VII
Índice de Tablas	XI
Resumen	XII
 1. Introducción	 13
1.1. Problemática	13
1.2. Contexto de la solución propuesta	14
1.3. Hipótesis	14
1.4. Objetivo general	14
1.4.1. Objetivo Específico	14
1.5. Alcance	15
1.6. Esquema del documento	15
 2. Marco Teórico	 17
2.1. Conciencia Fonológica	17
2.1.1. Enseñanza de la conciencia fonológica	18
2.1.2. Dificultades de Aprendizaje.	19
2.1.3. Tratamientos	20
2.2. Game Base Learning	20
2.3. Virtual Pet	21
2.4. Estado del Arte	22
2.4.1. Histórico	22
2.4.2. Actual	23

3. Marco Metodológico	25
3.1. Solución	25
3.2. Metodología	26
3.3. Evaluación	27
3.3.1. Objetivo de la prueba	27
3.3.2. Población y Muestra	28
3.3.3. Instrumento	28
3.3.4. Enfoque y tipo de Investigación	28
4. Diseño de Software	30
4.1. Requerimientos	30
4.2. Diagrama de Flujo	31
4.3. Menús	31
4.3.1. Selección de mascota y Menú Principal	31
4.3.2. Ejercicios	32
4.3.3. Menú de hogar	33
4.3.4. Galería de Mascotas	33
4.4. Text To Speech y Speech To Text	33
5. Implementación	36
5.1. Ejercicios	36
5.1.1. QuestionManager	36
5.1.2. ExercicePrefaController	37
5.2. SaveSystem y Load	38
5.3. Audio y voz	39
5.3.1. AudioManager	39
5.3.2. VoiceController	40
5.4. Amazon Web Service AWS	41
5.4.1. AWSManager	41
5.4.2. AWSPolly	41
5.4.3. AWSTranscribe y AWSS3	42
6. Análisis y Resultados	43
6.1. Resultados	43
6.2. Análisis	46

6.2.1. Análisis Percepción general	46
6.2.2. Análisis Usabilidad	47
6.2.3. Análisis elementos visuales y música	48
6.2.4. Análisis final	49
7. Conclusión	51
Bibliografía	54
A: Anexo estado del Arte	57
A.1. Abrapalabra	57
A.2. Mi primera encarta	59
A.3. ArbolABC.com	60
A.4. Primeras Palabras Sago Mini.	61
A.5. Abecedario, Aprende a Leer	62
A.6. LingoKids - Aprender Jugando.	63
A.7. Learn to Read Duolingo	64
B: Carta Gantt	67
B.1. Tabla carta Gantt ciclo 1(semana 1 a 3) y ciclo 2(semana 4 9)	68
B.2. Carta Gantt ciclo 3	69
B.3. Carta Gantt ciclo 4	70
B.4. Carta Gantt ciclo 5	71
B.5. Carta Gantt ciclos 1(semanas 1 a 3) y ciclos 2 (semanas 4 a 9)	72
B.6. Carta Gantt ciclos 3	73
C: Diseño de software	74
D: Anexo Código y App	81
D.1. QuestionManager	81
D.2. ExercicePrefa	85
D.3. SaveSystem	86
D.4. AudioManager	88
D.5. VoiceController	89
D.6. VoiceController	89
D.7. Amazon Web Service AWS	90

D.7.1.	AWSManager	90
D.7.2.	AWSPolly	92
D.7.3.	AWSTranscribe	93
D.7.4.	AWSS3	95
E:	Anexo Encuesta, Análisis y resultados	97
E.1.	Encuesta	97
E.2.	Gráficos	98
E.3.	Estudio de Arte	110
E.3.1.	Referencia artística de series infantiles y juegos	110
E.3.2.	Animales ocupados	113
E.3.3.	Habitad	118
E.3.4.	Diseño de personajes	121
E.3.5.	Paleta de colores	125
E.3.6.	Concepto inicial interfaz	126
E.3.7.	Interfaz primera build	127
E.3.8.	Interfaz build final	130

ÍNDICE DE FIGURAS

	página
A.1. Cuentin y el Mago Nombro.	57
A.2. Selección de mundo.	58
A.3. Ejemplo Actividad.	58
A.4. Menú Principal.	59
A.5. Ejemplo Actividad.	59
A.6. Juegos ABC.com.	60
A.7. Juego de hablar	61
A.8. Juego de Seleccionar.	61
A.9. Ejemplo de juego de selección.	62
A.10.Ejemplo de juegos	62
A.11.Ejemplo juego caligrafía.	63
A.12.Ejemplo juego caligrafía.	64
A.13.Juego armado palabra.	65
A.14.Estructura de cuento que guía el aprendizaje.	66
C.1. Diagrama de flujo de las distintas interfaces del software	74
C.2. Wireframe selección de mascota	75
C.3. Wireframe Menú Principal	75
C.4. Wireframe menu de Ejercicios	76
C.5. Wireframe de los 3 tipos de ejercicios	76
C.6. Wireframe cierre de ejercicio	77
C.7. Wireframe ventana de Cuidados	77
C.8. Wireframe ventana de galería	78
C.9. Diagrama de flujo iniciación AWS	78
C.10.Diagrama de flujo Polly Request	79
C.11.Diagrama Casos transcribe Request	80
E.1. Gráfico distribución demografía	98
E.2. Gráfico Pregunta: ¿El juego te hizo querer practicar más?	99
E.3. Gráfico Pregunta: ¿El juego te dejó hacer lo que querías?	99
E.4. Gráfico Pregunta: ¿Qué tan difícil fue el juego de seleccionar sílabas?	100

E.5. Gráfico Pregunta: ¿Qué tan difícil fue el juego de leer y hablar? . . .	100
E.6. Gráfico Pregunta: ¿Qué tan difícil fue el juego de escuchar y hablar? .	101
E.7. Gráfico Pregunta: ¿Te sirvieron las instrucciones con audio?	101
E.8. Gráfico Pregunta: ¿Cómo te sentiste al terminar la sesión de entrena- miento?	102
E.9. Gráfico Pregunta: ¿Te intereso poder jugar con otras mascotas? . . .	102
E.10. Gráfico Pregunta: ¿Te gustó la mascota con la que jugaste?	103
E.11. Gráfico Pregunta: ¿Te gustó la música de la mascota?	103
E.12. Gráfico Pregunta: ¿Te gustó la casa de la mascota?	104
E.13. Gráfico Pregunta: ¿El juego te hizo querer practicar más? por género	104
E.14. Gráfico Pregunta: ¿El juego te dejó hacer lo que querías? por género .	105
E.15. Gráfico Pregunta: ¿Qué tan difícil fue el juego de seleccionar sílabas? por género	105
E.16. Gráfico Pregunta: ¿Qué tan difícil fue el juego de leer y hablar? por género	106
E.17. Gráfico Pregunta: ¿Qué tan difícil fue el juego de escuchar y hablar? por género	106
E.18. Gráfico Pregunta: ¿Te sirvieron las instrucciones con audio? por género	107
E.19. Gráfico Pregunta: ¿Cómo te sentiste al terminar la sesión de entrena- miento? por género	107
E.20. Gráfico Pregunta: ¿Te intereso poder jugar con otras mascotas? por género	108
E.21. Gráfico Pregunta: ¿Te gustó la mascota con la que jugaste? por género	108
E.22. Gráfico Pregunta: ¿Te gustó la música de la mascota? por género . .	109
E.23. Gráfico Pregunta: ¿Te gustó la casa de la mascota? por género	109
E.24. Referencia Blue.	110
E.25. Referencia Puffin Rock.	110
E.26. Referencia Hilda.	111
E.27. Referencia Doctora juguete.	111
E.28. Referencia Magikarp Jump.	112
E.29. Referencia Docky y sus amigos.	112
E.30. Referencia sapos.	113
E.31. Referencia sapos cartoon.	113
E.32. Referencia carpinteros.	114

E.33.Referencia estudio de color del carpintero.	114
E.34.Referencia carpintero cartoon.	115
E.35.Referencia carpintero cartoon 2.	115
E.36.Referencia zorro cumpeo.	116
E.37.Referencia estudio de color del zorro cumpeo.	116
E.38.Referencia zorro cumpeo cartoon 1.	117
E.39.Referencia zorro cumpeo cartoon 2.	117
E.40.Referencia Bosque 1.	118
E.41.Referencia Bosque 2.	118
E.42.Referencia Bosque minimalista.	119
E.43.Referencia desierto 1.	119
E.44.Referencia desierto 2.	120
E.45.Referencia desierto minimalista.	120
E.46.Primer acercamiento, idea descartada de inmediato.	121
E.47.Carpintero estructura geométrica.	121
E.48.Sapo estructura geométrica.	122
E.49.Zorro cumpeo estructura geométrica.	122
E.50.Habitad estructura geométrica minimalista.	123
E.51.Carpintero final.	123
E.52.Sapo final.	124
E.53.Zorro cumpeo Final.	124
E.54.Habitad finales.	125
E.55.Paleta de colores utilizada.	125
E.56.Concepto interfaz 1 menú principal.	126
E.57.Concepto interfaz 2 casa mascota.	126
E.58.Concepto interfaz 3 minijuegos.	127
E.59.Interfaz inicio Prototipo 1.	127
E.60.Interfaz menú principal y pregunta de cierre Prototipo 1.	128
E.61.Interfaz menú ejercicios y videojuegos Prototipo 1.	128
E.62.Interfaz casa Prototipo 1.	129
E.63.Interfaz galería Prototipo 1.	129
E.64.Interfaz inicio Prototipo final.	130
E.65.Interfaz selección de mascota Prototipo final.	130
E.66.Interfaz menús y pregunta de seguridad salida prototipo final.	131

E.67. Interfaz minijuegos y termino de entrenamiento prototipo final.	131
E.68. Interfaz casa prototipo final.	132
E.69. Interfaz galería Prototipo final.	132

ÍNDICE DE TABLAS

	página
2.1. Ejercicios conciencia fonológica, según Elena Cabeza Pereiro y Jesica Araya	19
4.1. Distribución de preguntas entre la C, S y Z	32
6.1. Preguntas encuestas	44
6.2. Tabla de general resultados	44
6.3. Tabla resultados niños	45
6.4. Tabla resultados niñas	45
E.1. Preguntas encuestas	97
E.2. Valores resultados encuesta	98

RESUMEN

La conciencia fonológica corresponde a la capacidad de tomar entendimiento de cualquier unidad fonológica, siendo la base para posteriormente aprender a escribir y leer, por tanto, un mal aprendizaje a temprana edad afecta directamente a la ortografía y la comprensión lectora a futuro. Esta investigación se enfoca en las consonantes C, S y Z, las cuales pueden presentar dos tipos de problemas: el “se-seo” y la interpretación del fonema, ya que estas tres consonantes ocupan el mismo sonido, ocasionando que la distinción entre ellas sea difícil al momento de aprender. Por esto mismo, si estas dificultades persisten al término del cambio de dientes por parte del infante, estos tienen que ser tratados por diferentes especialistas. Este tratamiento pide trabajo autónomo, el cual puede llegar a ser monótono y/o aburrido, ocasionando que el tratamiento se alargue y en consecuencia el tiempo en que los niños aprenden a leer y escribir sea mucho mayor. El presente proyecto tiene como objetivo la creación de un software interactivo que acompañe el aprendizaje de estas consonantes en estudiantes de 1° básico, dividiéndose en 3 ejes: juegos fonéticos/gramaticales, gramaticales/fonéticos y fonéticos/fonéticos, donde las respuestas de los niños afectan el mundo del juego. Para la realización de dicho trabajo se utiliza la metodología de desarrollo incremental, teniendo una primera etapa de diseño centrada en el usuario, generando un prototipo nivel TRL3, el cual se encuentra enmarcado dentro del proyecto Fondecyt de iniciación 11220812 “Evaluación de impacto de juegos reglados sobre predictores de alfabetización en niños de 1° básico en contextos de alto y bajo nivel socioeconómico”, este prototipo fue probado por estudiantes del Instituto Linares, y con estos resultados se pudo apreciar la predisposición de los niños a utilizar mascotas virtuales en el aprendizaje pedagógico y la usabilidad de la aplicación, que incorpora tecnologías *text-to-speech* y *speech-to-text* para guiar las actividades. El alcance es ofrecer una herramienta autónoma y atractiva que complemente el tratamiento tradicional, pudiendo analizar los distintos puntos trabajar y cómo los usuarios fueron capaces de utilizar el prototipo.

1. Introducción

1.1. Problemática

La conciencia fonológica corresponde a la capacidad de tomar entendimiento de cualquier unidad fonológica, es decir las sílabas y los fonemas, siendo estas la base para poder aprender a escribir y leer. En consecuencia, un mal aprendizaje a temprana edad puede afectar directamente a la ortografía y la comprensión lectora a futuro. Dentro de los problemas que pueden afectar a la conciencia fonológica se encuentran, la sustitución u omisión de fonemas, segmentación de palabras o problemas al unir o fragmentar palabras, entre otras, destacando las consonantes C, S y Z las cuales presentan dos tipos de problemas: el “seseo” y la interpretación del fonema, ya que estas tres consonantes ocupan el mismo sonido, ocasionando que la distinción entre ellas sea difícil al momento de aprender. Para los problemas de pronunciación en general, existen dos factores: la fisonomía maxilofacial, que puede ser tratada por ortodoncia y/o operación, y por otra parte, por un problema de modulación, apuntando a una dificultad en el posicionamiento de la lengua y flujo del aire, cayendo por lo tanto en una incorrecta pronunciación. En este caso el tratamiento se realiza mediante distintos juegos y ejercicios de respiración, labios y lengua de manera diaria.

Estos ejercicios pueden llegar a ser monótonos a largo plazo, sobre todo en niños, ocasionando no se han capaces de mantener la atención en los ejercicios que están haciendo, haciendo que se fatiguen rápidamente y produciendo que los tratamientos se terminen alargando si es que no se sigue el régimen diario, aumentando en consecuencia el tiempo en que los niños aprenden a leer y escribir, pudiendo también llegar a afectar la ortodoncia en el futuro.[1].

Esto puede ser motivado por dos posibles factores: uno es que la materia no es llamativa y lúdica para el niño, mientras que el otro factor se encuentra la sobre satu-

ración de distracciones que existe actualmente, puesto que, es importante considerar que los niños en la actualidad son nativos digitales, Conllevando un manejo fluido e intuitivo del uso de tecnologías, y por ende, la cantidad de estímulos con los que se encuentran día a día afecta directamente a la capacidad de atención de estos[1]. Con todo lo mencionando anteriormente, es necesario considerar nuevas formas de desarrollar ejercicios para la adquisición de conciencia fonológica en niños, para así, mantener motivados a los niños sus respectivas terapias o clases, además de generar una plataforma que pueda de ser utilizada en zonas distantes o fuera del salón de clases.

1.2. Contexto de la solución propuesta

La solución propuesta consiste en un software interactivo utilizando Game Based Learning y diseño instruccional, el cual se encuentra desarrollado para dispositivos móviles de sistema con operativo Android, la cual se encuentra enfocada para niños y niñas entre los 5 a 7 años que presenten problemas de lenguaje, específicamente en la pronunciación de las consonantes S, C y Z, en el marco del trabajo autónomo del proceso de aprendizaje.

1.3. Hipótesis

Mediante la utilización de un software interactivo para dispositivos móviles que aplique técnicas de Game Based Learning, los niños en edad escolar podrán entrenar de forma autónoma los grafemas y fonemas de las consonantes S, C y Z en sus casas, mediante la utilización de dispositivos móviles.

1.4. Objetivo general

Desarrollar un software educativo para el entrenamiento fonológico de las letras S, C y Z en niños y niñas de 1° ciclo básico.

1.4.1. Objetivo Específico

- Buscar información referente a trastornos del habla, especialmente en las consonantes S, C y Z, además de sus correspondientes tratamientos fonológicos.

- Analizar técnicas fonológicas para el aprendizaje de las consonantes S, C y Z en niños entre los 5 a 7 años
- Desarrollar el documento de diseño para el desarrollo del software, teniendo en cuenta técnicas de game based learning, con enfoque en niños de 1° ciclo básico.
- Diseñar juegos mediante técnicas de gamificación y diseño instruccional, enfocadas en el rango etario.
- Desarrollar prototipo del software interactivo, en colaboración con docentes y fonoaudiólogos.
- Realizar testing del software en niños de 1° ciclo básico.
- Analizar los resultados del testing, para proponer las distintas mejoras necesarias al software.

1.5. Alcance

Se espera que este proyecto llegue a un nivel de madurez tecnológico TRL 3, es decir una prueba de concepto, siendo testada por niños y niñas de 1° ciclo básico del colegio Instituto Linares, de la comuna de Linares, con la finalidad de identificar los distintos puntos a mejorar, para continuar su posterior desarrollo y analizar el impacto en niños y niñas de esta edad.

1.6. Esquema del documento

El documento se estructura en siete capítulos: Introducción que aborda la problemática, el contexto de la solución, sus objetivos y alcance; un Marco Teórico que desarrolla la conciencia fonológica, el Game-Based Learning, el enfoque de mascotas virtuales y el estado del arte; un Marco Metodológico que detalla la solución, la metodología utilizada y el proceso de toma de datos, incluyendo objetivos, población y procedimiento de las pruebas. Posteriormente, el capítulo de Diseño de Software presenta los requerimientos, diagramas, arquitectura de menús y la integración de sistemas *text-to-speech* y *speech-to-text*, seguido de la Implementación, que explica

el desarrollo de los sistemas principales del software con sus respectivos códigos. Finalmente, se exponen el Análisis y Resultados de las pruebas realizadas en el grupo objetivo, para concluir con las Conclusiones que cierran los objetivos del trabajo y plantean el futuro desarrollo.

2. Marco Teórico

2.1. Conciencia Fonológica

La conciencia fonológica o conciencia metalingüística corresponde a la base para poder aprender a escribir y leer, puesto que se define como la toma de conocimiento consciente del hecho que las palabras se encuentra compuestas por diferentes unidades de sonidos, además de la capacidad de manipular las distintas unidades del lenguaje hablado[2]. Por consiguiente comprendiendo que las palabras se encuentran formadas por sonidos, siendo sus unidad más básica los fonemas y posteriormente las sílabas. Por otra parte, al establecer la relación entre los fonemas y los grafemas, se genera la capacidad lectora en los niños, siendo estas las bases para una correcta comprensión lectora a futuro[3]. Defior establece que las sílabas corresponden a unidades que pueden ser producidas y percibidas de maneras aisladas por las personas, pero con respecto a la relación letra y fonema establece que es una forma óptima de representación, ya que, con un número limitado de símbolos, se pueden representar todos los sonidos del lenguaje. Conllevando una gran dificultad en su aprendizaje por su alto nivel de abstracción[4].

Es importante indicar los distintos niveles de segmentación fonológica para la adquisición de la conciencia fonológica teniendo estos distintos niveles jerárquicos de complejidad, partiendo por la Conciencia Léxica, Silábica, intrasilábica y fonémica, siendo definidas de la anterior manera según Defior y Serrano[5].

Como se mencionó los niveles de conciencia fonológica tienen su propio grado de dificultad, siendo la sílaba la unidad más fácil de reconocer y aprender, pudiendo ser aprendida está antes que a leer en algunos casos, Mientras que el desarrollo de habilidades que implican la conciencia instrasilábica y fonética es a edades más

tardías[5]. Pudiéndose ver un desarrollo de la conciencia fonológica entre la edad de 3 a 6 años antes de la alfabetización formal[6].

2.1.1. Enseñanza de la conciencia fonológica

La conciencia fonológica es una capacidad que no surge de manera espontánea en los niños, sino que ocurre cuando estos se tienen que enfrentar a tareas que los fuercen a reflexionar sobre el uso del lenguaje[6]. Por esto mismo durante el proceso de aprender a leer, es importante tener en cuenta el factor cultural en que se desenvuelven los infantes puesto que diferentes estudios establecen que estos aprenden imitando a sus parientes más cercanos o maestros en el colegio, por lo tanto diferentes niveles de escolaridad en la familia y/o el acceso a material escrito afecta directamente en el proceso de aprendizaje de esta capacidad[7].

Para el proceso de adquisición de la conciencia fonológica se pueden encontrar diferentes ejercicios fonéticos y silábicos, siendo importante iniciar trabajando la toma de conciencia de las sílabas en el caso del español[3], con tareas como la identificación de cantidad de sílabas dentro de una palabras o la identificación de estas.[5]. Con respecto a conciencia intrasilábica las tareas son todas aquellas que requieran la identificación y creación de palabras con un elemento diferenciador[5]. Finalmente para la conciencia fonológica se encuentran aquellos que identifican un de fonema dentro de una palabra o armar palabras en base a distintos fonemas[5].Ocupando como base las investigaciones de Elena Cabeza Pereiro y como las de Jesica Araya Ramírez[8][9], se mencionarán algunos ejercicios que abordan estas áreas.

Finalmente hay que tomar ciertas consideraciones en la adquisición de la conciencia intrasilábica y fonética, ya que estas se consolidan mientras se aprende a leer, pero es importante que se empiece a desarrollar con anterioridad para facilitar su proceso. Ya que como muestra un estudio realizado por Jiménez y Ortiz en un grupo de niños entre los 5 años y 6 meses, se aprecia un 91 % de éxito en conciencia silábica, 56 % en tareas fonéticas que incluyen vocales y 34 % en aquellas con consonantes[10]. Por otra parte es importante considerar el vocabulario del niño, puesto que como ya se mencionó anteriormente los conocimientos del niño se ven afectados por un factor cultural y socioeconómico en que se desarrolla[7], además que a la edad de 6 años estos manejan un aproximado de 14.000 palabras[5].

Conciencia Silábica	Conciencia Intra-silábica	Conciencia Fonética
Reconocimiento de sílabas iniciales dentro de palabras.	Segmentación de sílaba inicial o final.	Juego de onomatopeyas.
Conteo de sílabas, dentro de palabras bisilábicas a trisilábicas.	Ejercicios de rimas.	Reconocimiento de sonido fonémico.
Identificación sílaba medial.	Comparación de sílabas iniciales.	Segmentación de fonemas.
Inversión de sílabas, cambiando el palabras bisilábicas.		Conteo fonémico.
Creación de palabras atreves de sílabas.		Unión de segmentos fonémicos.
		Comparación de segmentos fónicos.

Cuadro 2.1: Ejercicios conciencia fonológica, según Elena Cabeza Pereiro y Jesica Araya

2.1.2. Dificultades de Aprendizaje.

Al saber que la conciencia fonológica está directamente relacionada con las habilidades lectoras y del habla, es importante ver los diferentes trastornos que la pueden afectar de igual manera. Para lo cual Bosch define como “aquellos sistemas que difieren del que corresponde a la lengua del adulto y que en muchos casos se manifiestan con organizaciones estables, internamente coherentes, pero no idénticas a las que correspondería haber accedido si el proceso de aprendizaje hubiera culminado satisfactoriamente.” [11], es decir plantear lo que se quiere decir manera consciente y coherente, pero siendo incapaz de verbalizar satisfactoriamente por diferentes motivos. Dentro de estos factores se identifican 2 tipos de trastornos los fonológicos y los fonéticos, caracterizándose los primeros por presentar problemas en la organización de los sonidos, pudiéndose encontrar patrones en a lo largo del tiempo. Mientras que los fonéticos corresponden a un problema práxico, articulatorio, teniendo causas físicas, generando errores aislados sin presentar patrones en el tiempo [11]. Las 3 grandes categorías en los trastornos fonológicos planteados por Bosch son:

- Procesos sustitutorios, afectando a sonido que pueden manifestándose mendicante el cambio en el punto de articulación de la palabra.

- Procesos relativos a la estructura silábica, que corresponde a la reducción de palabras polisílabas mediante la pérdida de sílabas átonas o duplicación de alguna de ellas
- Procesos Asimilatorios, se refieren a las contaminaciones entre segmentos de sílabas distintas pero próximas.

Ya con esto en mente se puede entender que los procesos anteriormente mencionados terminan teniendo la finalidad de simplificar las palabras del sistema adulto para poder facilitar la producción por parte de los infantes. Teniendo en cuenta que el nivel de ininteligibilidad se encuentra directamente relacionado con la cantidad de procesos cognitivos que el niño produce[12].

2.1.3. Tratamientos

Para poder enfrentar las disfunciones del lenguaje y trastornos del habla, es necesario realizar un trabajo multidisciplinario, que evalúe todos los posibles factores que imposibiliten el lenguaje. Una vez realizado, es necesario empezar con el tratamiento cuyo objetivo es poder reeducar de manera individualizada y específica. Siendo importante la integración del contexto familiar y escolar en que se encuentra el paciente. Por esto, actualmente se utilizan distintas técnicas para el manejo de problemas del habla tales como repetición de palabras o ejercicios gramaticales, utilizando también grupos de trabajo con niños que no presenten trastorno del habla con aquellos que presenten trastorno, de esta manera se generan situaciones que promuevan una comunicación natural y que estimulen la socialización, así mismo, es normal involucrar a la familia, ya que esta es la referencia lingüística del niño[13].

2.2. Game Base Learning

Esta metodología se basa en un diseño guiado por el aprendizaje[14], cuya meta es facilitar dicho proceso, encontrando un balance entre los saberes que se buscan presentar y el gameplay, ya que si el foco se encuentra enfocado en el aprendizaje el juego, no se va sentir como tal y por otra parte, si nos enfocamos en el gameplay, los aprendizajes se pueden perder[15]. No obstante, la gamificación consiste en agregar elementos de juego tales a un ambiente sin juego para poder hacer que las tareas

sean más atractivos, donde el contexto de enseñar no cambia, si no que el cómo se aplica[16]. Para esto es necesario tener en cuenta distintos factores que pueden determinar el flujo de juego, siendo definido, como el nivel inmersión determinado por el desafío presentado en el juego y la habilidad del jugador[17]. Para esto es necesario tener en cuenta: los objetivos del Juego, las mecánicas, las dinámicas, la libertad, el game fantasy, la narrativa, las sensaciones, el game value, los desafíos, el factor social y el misterio.

El diseño de juegos educativos se fundamenta en varios elementos interrelacionados. Los objetivos definen la experiencia del usuario a corto y largo plazo, alineándose con los propósitos pedagógicos [17, 15]. Las mecánicas, corresponden a las acciones que el jugador realiza para alcanzar las metas, se clasifican en aprendizaje y evaluación [15, 16]. Las dinámicas describen la interacción usuario-entorno a través de estas mecánicas [15], mientras que la libertad, la variedad de acciones disponibles, que influyen en la dificultad [15].

Además, el Game Fantasy abarca el entorno y el trasfondo del juego, requiriendo una integración orgánica de los contenidos educativos [15]. La narrativa contextualiza el aprendizaje y conecta elementos del juego. mientras que el diseño visual y sonoro impacta las sensaciones del jugador, para motivarlo y dirigiendo su atención [16, 15].

Finalmente, el Game Value, es crucial para la inmersión y motivación [17], se divide en factores intrínsecos (conocimiento adquirido) y extrínsecos (sistemas de puntuación) [16]. El factor social, mediante interacción multijugador o NPCs, busca estimular la motivación por parte de los usuarios[15]. Y el factor de misterio apela a la curiosidad y el deseo de descubrimiento [15].

2.3. Virtual Pet

Virtual pets (V-pets) o mascota virtual corresponde a un subgénero dentro de los juegos de simulación, el cual tiene como objetivo simular las distintas atenciones necesarias al tener una mascota, pero dentro de un entorno virtual. Siendo necesario tener distintos cuidados para que estas puedan seguir vivas y crecer, buscando estimular las interacciones diarias con la mascota para cumplir con este objetivo[18]. Estos presentan usualmente un ciclo de juego circular donde al usuario se le entrega una mascota, procede a iniciar los cuidados, hasta que esta es perdida, iniciando el ciclo de nuevo.

Así mismo, con el largo de los años, desde su introducción al mercado en la década de los 90', estos tipos de juegos ha impactado en la sociedad, pero principalmente en los niños por el ciclo de juego y las mecánicas que presenta. Generando distintas tendencias a lo largo de los años, como son las marcas Petz, Tamagochi, Digital Monster(Digimon), Pou o My Talking Tom, entre otras, pudiendo tener distintos elementos característicos que los diferencien, ya sea en hardware o mecánicas principales, pero conservando los mismos principios fundamentales, como la comunicación de la mascota mediante sonidos, animación o cuadro de textos. La interactividad con la mascota se da mediante diferentes acciones, como entrenamiento, juegos o cuidados necesarios. Finalmente, considerando el factor social que se genera ya sea para combatir con otras mascotas o compartirlos con personas cercanas al jugador[18].

Finalmente, se necesario mencionar el impacto que pueden tener este tipo de juegos en los niños, como se ha podido observar en distintos estudios. Pudiéndose encontrar un aumento en la actividad física realizada por niños que utilizaban V-pets como un motor motivacional[19], y se ha observado como estas son capaces de alterar el comportamiento alimenticio en niños, en base al tipo de feedback que se le entrega[20], así mismo ver su impacto para reducir la obesidad infantil, como presenta el estudio de Kyle Johnsen[21]. Por lo tanto, la implementación de este tipo de juegos en entornos educacionales como factor motivacional, puede ser capaz de generar una impacto positivo en el usuario, dependiendo del tipo de refuerzo que se entrega.

2.4. Estado del Arte

En la siguiente sección se presentara el estado de arte de software educacionales tanto históricos como actuales.

2.4.1. Histórico

Abrapalabra

Videojuego lanzado el año 1988 y diseñado por Odisea Software, cuya premisa se centra en ayudar al conejo “Cuentin”, el cual quiere aprender los trucos de magia del mago “Nombrón, los cuales funcionan como elemento conector entre el jugador y la historia que se plantea contar, ya que estos constantemente se dirigen al usuario,

en las distintas actividades.

En juego se presentan distintos ejercicios a través del abecedario español, centrados principalmente en la identificación de palabras con imágenes y sonidos, así mismo como ejercicios de encontrar sílaba inicial y rima.(ver Anexo A.1, fig. 1, 2 y 3)

Mi Primera encarta

Software educativo lanzado por Microsoft estudio el año 1998 con soporte hasta el año 2009, en la cual el usuario podía buscar información de diversos ámbitos, pudiendo, por ejemplo, explorar diferentes arquitecturas históricas, como pudiendo realizar diferentes juegos educativos en letras o matemáticas. Destacando principalmente esa dualidad existente, tanto como videojuego educativo como fuente de información digital.(ver Anexo A.2, fig. 4 y 5)

2.4.2. Actual

ArbolABC.com

Página web educativa enfocada en niños entre los 3 a 10 años, que cuenta con una variedad de juegos, enfocadas principalmente en desarrollar las habilidad de lecto-escritura en los usuarios, pasando por juegos donde se centran en memorizar vocales a aquellos de mayor dificultad, donde se le pide al menor trabajar con palabras, pasando por una gran variedad de actividades enfocadas en diferentes áreas de proceso de lectura.(ver Anexo A.3, fig.6)

Primeras Palabras Sago Mini

Aplicación con suscripción para dispositivos móviles, enfocada en aprendizaje lector en el idioma inglés, la cual es distribuida por “Piknik”. En esta se encuentran distintas actividades enfocadas fuertemente en que el niño pueda ir hablando con el software, por lo tanto se encuentran juegos de articulación, vocabulario y juegos con canciones, utilizando animales y videos con niños para ir guiando al usuario en como modular y decir las palabras que se busca enseñar.(ver Anexo A.4, fig. 7 y 8)

Abecedario, Aprender a Leer

Juego para dispositivos móviles y PC, desarrollado por Bini games, el cual se enfoca en el aprendizaje del abecedario. Utiliza figuras simples para los elementos gráficos del juego, donde sus juegos se centran principalmente en actividades de reconocimiento de letras, enfocándose en su mayoría juegos de reconocimiento y caligrafía, para abarcar los saberes que se buscan enseñar.(ver Anexo A.5, fig. 9 y 10)

Lingokids - Aprender Jugando

Aplicación con suscripción para dispositivos móviles, enfocada en aprendizaje lector en el idioma inglés, distribuida por Lingokids y ganando el premio a mejor app educativa por kidscreen awards el año 2023. Esta cuenta con una interfaz y elementos gráficos simples pero fáciles de reconocer por el usuario. Sus minijuegos tienen una gran variedad de temáticas pasando por juegos de cuidados a juegos de cocina y plataformas 2D los cuales tiene como temática principal el saber que se busca enseñar, ya sea lenguaje o matemáticas.(ver Anexo A.6, fig. 11)

Learn to Read Duolingo

Esta es una aplicación móvil desarrollada por Duolingo, la cual se enfoca a enseñar a vocabulario, pronunciación y caligrafía, mediante diversos minijuegos los cuales se enfocan en el armado de palabras a partir de letras hasta el armado de oraciones, siendo guiados por una pequeña historia en cada una de las lecciones.(ver Anexo A.7, fig. 12, 13 y 14)

3. Marco Metodológico

En esta sección se revisará la propuesta de implementación, la metodología aplicada, las herramientas a usar y el plan de pruebas a implementar.

3.1. Solución

Se plantea como solución el desarrollo de un software enfocado a niños que se encuentren entre los 4 a 6 años. Para dispositivos móviles con sistema operativo Android, adaptando los ejercicios fonéticos a la plataforma e implementando Amazon Web service para distintas funcionalidades, como las funcionalidades de voz y almacenamiento de datos, además de utilizar captación de sonido para poder identificar y almacenar la ejecución de los ejercicios por parte de los estudiantes, utilizando Game Based Learning para guiar los aprendizajes y las tareas que son necesarias que los estudiantes desarrollen en sus tratamientos. Para esto se deberán utilizar mecánicas similares a las mascotas virtuales o V-pet, donde el usuario deberá criar una mascota mediante diferentes actividades y cuidados, mezclando el ciclo de este tipo de juegos con los saberes que buscamos trabajar. Para esto el usuario deberá escoger un eje de trabajo, los cuales están asociados a cierto tipo de mascota, que sea representativa al conjunto de sílabas con las cuales se va a trabajar. Esta contará con diferentes estadísticas, las cuales deberán ser cuidadas mediante distintos ejercicios, los cuales se encuentran divididos y enfocados en los distintos niveles de la conciencia fonológica, presentando un ciclo de juego, donde el usuario deberá escoger una mascota, cuidar y finalmente guardarla, para empezar su cuidado con otra mascota que escoja.

Con cada ciclo de juego, se irán registrando y almacenando los datos para posteriormente ser procesados, estos datos consistirán: en el eje trabajado, cantidad de

ejercicios realizados por cada nivel de la conciencia fonológica y los distintos audios por ejercicio.

3.2. Metodología

Para la creación del prototipo se hará uso de metodología de desarrollo incremental, dado que facilita el hacer cambios dentro del prototipo, permitiendo ir agregando, eliminando o cambiando componentes del software. Tiene un enfoque en crear un sistema por pasos, comenzando con un prototipo básico y agregando funcionalidades de manera gradual. Durante cada iteración las dichas funcionalidades son probadas para ir ajustando y mejorando continuamente, reduciendo riesgos y garantizando que el producto final satisfaga las necesidades del usuario. Para cumplir con esto se tomaron 6 ciclos de trabajos los cuales son(ver carta gantt, anexo b1 a b6):

- Establecimiento de requisitos y diagramas de flujos iniciales con wireframes iniciales.
- Implementación primera iteración mascota sin arte inicial, enfocado en la creación de estas con el menú de hogar.
- Primer prototipo de ciclo completo pasando por la selección de mascota al ejercicio de selección utilizando únicamente ejercicios grafema/grafema, para comprobar las funcionalidades base.
- Ciclo agregando las funcionalidades finales de creación de mascota y juegos correspondientes al STT¹.
- Funcionamiento flujo completo de la aplicación sin funcionalidades de TTS² con la utilización de HuggingFace para el STT.
- Ciclo Final agregando AWS³ para todas las funcionalidades de voz, ajustando ítems y agregando sistema de puntos para ajustar a tiempos de respuestas por parte AWS.

¹STT: Speech to Text o Voz a Texto

²TTS: Text to Speech o Texto a Voz

³AWS: Amazon Web Services

El prototipo consiste en una aplicación para dispositivos Android, el cual fue programado utilizando Unity 2022.3.53f1. Para el apartado artístico se optó por Leonardo.ai para la creación de arte y Suno AI para la música, siendo ambas inteligencias artificiales generativas y las modificaciones se realizaron en adobe illustrator. Finalmente con respecto a las funcionalidades de STT y TTS se uso inicialmente Huggingface, pero por falta de soporte, se analizaron servicios como Azure, Google Cloud y Amazon Web Service, prefiriendo en última instancia a AWS por su documentación.

Finalmente el Prototipo llegó a un nivel de madurez tecnológica TLR3, el cual fue probado por niños de la comunidad de Linares. Esto con la finalidad de recabar información sobre los diferentes factores a mejorar dentro del software planteado, Pudiendo generar un plan de mejora.

3.3. Evaluación

Como se mencionó anteriormente, el objetivo es poder conseguir un prototipo TRL3, con la finalidad de probar su usabilidad.

3.3.1. Objetivo de la prueba

Teniendo en cuenta el nivel tecnológico del prototipo, se busca generar un estudio en usuarios reales, para validar y comprobar la usabilidad de la aplicación. Para esto, el objetivo de la prueba es analizar: la usabilidad de la aplicación, dificultad de los ejercicios y los elementos gráficos. Así mismo, de la facilidad de uso al realizar los ejercicios pertinentes, analizando cuál es el ejercicio preferido por los usuarios y el porcentaje de acierto en los ejercicios, entre otros factores.

Por otra parte, es importante recalcar que para ver el impacto real, es necesario realizar otro estudio, ya que un estudio de esa envergadura deberá ser medido a lo largo del tiempo, observando como el uso de la aplicación afecta el aprendizaje de las sílabas a trabajar, quedando fuera del alcance de esta memoria. Por lo tanto, del presente estudio se podrá extraer los diferentes aspectos a mejorar, ya sea en el flujo del juego, aspecto estéticos o enfoque de los ejercicios, así como temas de balance general, generando un análisis de dichos datos para establecer un plan de mejoras para su lanzamiento a futuro.

3.3.2. Población y Muestra

Niños y niñas de 1° ciclo básico, con una edad promedio entre 6 a 7 años, los cuales están en el proceso de aprendizaje de la lectura y la obtención de conciencia fonológica. Tomando en cuenta la población definida, se tiene como muestra una cantidad de 22 niños, del colegio Instituto Linares de la ciudad de Linares, región del Maule, divididos en 11 niños y 11 niñas, los cuales ya tenían un conocimiento previo sobre el uso de dispositivos móviles, debido al programa pedagógico que tiene el colegio. Por otra parte, como se probará una aplicación móvil, se contará con un total de 3 dispositivos con accesos a internet, los cuales se dividen en 2 teléfonos uno de la marca Samsung y otro de la marca Redmi y 1 tablet de la marca Samsung, teniendo todos los dispositivos sistema operativo android versiones 12, 14 y 14 respectivamente. Finalmente como elemento adicional se tendrán 3 audífonos para que los niños puedan realizar las pruebas sin ninguna dificultad, los menores fueron supervisados por: el encuestador, una ayudante de sala y la jefa de UTP del establecimiento, además todos los tutores tienen firmado el asentimiento informado, en el contexto del proyecto Fondecyt de iniciación 11220812.

3.3.3. Instrumento

Las pruebas se dividen en dos partes, los datos obtenidos desde la aplicación correspondiente al eje que se decidió trabajar, siendo estos datos acompañados con una encuesta propia enfocada en la usabilidad y los gráficos de la aplicación. Tomando como elementos principales preguntas sobre la facilidad de uso y entendimiento de los ejercicios y menú, así como el atractivo del arte, como se puede apreciar en el anexo D.1. Con esto en mente la encuesta está llevada en escala de 1 al 5 preguntando sobre estos elementos, ocupando como base principal para las preguntas la escala de Wong-Baker para cada pregunta. Esta consiste en un total de 11 preguntas divididas en 3 preguntas de percepción general, 5 de usabilidad y 3 para el arte. Finalmente, todas las preguntas serán leídas por el encuestador para facilitar la comprensión del menor.

3.3.4. Enfoque y tipo de Investigación

Teniendo estos elementos definidos, se puede establecer que la presente investigación cuenta con un enfoque mixto, puesto que en este caso se busca extraer

información de datos cualitativos, para sus posterior análisis, buscando obtener estadísticas sobre las distintas respuestas obtenidas en los test a realizar.[22] Por otra parte se puede establecer que el tipo de investigación tiene un carácter experimental, ya que basándonos en los escritos de Sampieri, define a la investigación experimental como a toda aquella que requiera de la manipulación intencionalmente de una acción o una variable para analizar sus posibles resultados. [23]

4. Diseño de Software

En el siguiente capítulo se procederá a explicar cómo se encuentra diseñada la aplicación, como su implementación. Pasando por sus distintos menús, juegos vocales y de selección, además de distintas clases del sistema, que son necesarios para el funcionamiento del software.

4.1. Requerimientos

A partir del análisis del estado de arte, se derivó en los siguientes requerimientos de software, planeándose las siguientes necesidades para la aplicación, tomando como enfoque principal el rango etario de los usuarios.

- Una paleta de colores atractiva, sin la utilización de colores muy saturados, para no molestar a la vista de los usuarios.
- Una interfaz simple, en donde todos los elementos que se quieren presentar sean claros para el usuario.
- Mascotas representativas de la sílaba con la cual se busca trabajar, así mismo que sean enfocados en la fauna chilena.
- Un diseño de amigable de las distintas mascotas, también de los íconos de la aplicación.
- Sistema de voz, ya sea vocal o por algún servicio de IA, que sea fácil de entender para el usuario. Con la finalidad de dar instrucciones a los usuarios de las actividades y así mismo de las distintas líneas de diálogos que se tiene dentro de la aplicación.

4.2. Diagrama de Flujo

La aplicación cuenta con un flujo enfocado principalmente en la realización de los ejercicios vocales y de selección de sílabas, para el entrenamiento de las mascotas. Las cuales son escogidas al iniciar la aplicación como se puede apreciar en la figura N°[4]. Por otra parte, existen otros dos flujos uno secundario y otro terciario, los cuales responden a dos necesidades, la primera enfocada en los cuidados de la mascota como en la recuperación de energía y la alimentación. Mientras que el otro flujo corresponde a la galería de mascotas, dentro la cual se pueden ver todas las distintas mascotas que se entrenaron junto con sus estadísticas, con las cuales terminaron su entrenamiento (ver figura C.1 en anexos).

4.3. Menús

4.3.1. Selección de mascota y Menú Principal

Como se puede apreciar a partir del diagrama de flujo de la aplicación, se puede identificar que al iniciar sección se puede generar dos flujos distintos, uno correspondiente a la selección de mascotas y un segundo flujo que hace de bypass para ir directamente al menú principal en el caso de contar con una mascota.

La selección de mascotas consta con 5 pantallas, funcionando la primera con un elemento introductorio al juego, mientras que las siguientes 3 para definir los elementos identificativos de la mascota, como su tipo (C, S o Z), su nombre y el del niño. y una última pestaña de confirmación. En todas estas ventanas siempre se tiene presente la imagen de la mascota, acompañado por su música característica y las instrucciones en formato audio para ir guiando al usuario (ver figura C.2 en anexos).

Por otra parte la pantalla del menú Principal, funciona como eje central de toda la aplicación donde todos los menús terminan llegando a esta pantalla, manejando 3 opciones distintas: (1)la opción de entrenamiento, (2)galería de mascotas y (3) el hogar de la mascota. Derivando a la escena correspondiente. Así mismo, aquí se puede apreciar en el nivel de la mascota y si se encuentra cansada y lista para seguir entrenando (ver figura C.3 en anexos).

Finalmente en esta pantalla se llama a Text to Speech, con la finalidad de explicar las distintas opciones al usuario y darle una bienvenida a la mascota que generó el

usuario. Así mismo del estado de la mascota, en el caso de que se encuentre cansada.

4.3.2. Ejercicios

Con respecto a las pantallas de ejercicios, el flujo principal cuenta con 3 pantallas distintas, pero en total se plantean la existencia de 5 pantallas. Partiendo por la pantalla de selección de ejercicios, la cual se divide en dos partes, la presentación del minijuego, constando del ícono y el nombre del ejercicio a escoger, y el botón de selección, cumpliendo la función de guardar el ID del Ejercicio seleccionado y transicionar al escenario correspondiente al minijuego (ver figura C.4 en anexos).

Ya con el minijuego seleccionado, se cargan uno de los 3 ejercicios existentes, los cuales todos funcionan con base de un Text to Speech para dar las instrucciones de la sílaba que tienen que encontrar de la palabra requerida. Como se puede apreciar, dos de los 3 ejercicios pide al usuario hablar presionando el botón del micrófono para poder ingresar su respuesta, mientras que el caso del de selección solamente tiene que seleccionar la sílaba requerida. Y una vez contestada la pregunta, se carga una nueva pregunta hasta que el usuario decida de jugar presionando el botón de terminar o se acabe el tiempo.

Las tres actividades implementadas se fundamentaron en los ejercicios de conciencia fonológica detallados en el marco teórico (Cuadro 2.1), abarcando los niveles silábico, intrasilábico y fonético. Estos juegos se presentaron en modalidades auditiva, vocal y de grafemas, requiriendo el reconocimiento de sílabas dentro de palabras, la segmentación de sílabas iniciales o finales, y la identificación de sonidos fonéticos de las palabras (ver figura C.5 en anexos).

Por otra parte, la sección trabaja con un total de [156] preguntas divididas de la siguiente manera entre la c, s y z, enfocándose en ejercicios de sílaba inicial y sílaba terminal.

Enfoque	C	S	Z
Sílaba Inicial	30	34	40
Sílaba Terminal	24	22	28
Total	54	56	58

Cuadro 4.1: Distribución de preguntas entre la C, S y Z

Finalmente una vez terminado el minijuego, el usuario es enviado a la pantalla de resultados, donde se le presenta la cantidad de energía restante y los ítems que

consiguió, durante el entrenamiento. Así mismo, se le da posibilidad de dirigirse al menú principal o realizar de nuevo el ejercicio en el caso de tener energía por parte de la mascota (ver figura C.6 en anexos).

4.3.3. Menú de hogar

El último menú importante corresponde al hogar, teniendo incorporada la funcionalidad de STT para dar las instrucciones a los usuarios la primera vez que ingresan a esta pantalla. Esta sección se divide en dos partes clave: la interfaz de recuperar energía, que se encuentra a la derecha de la figura 4.7, permitiendo a la mascota dormir para ganar 10 de energía cada 60 segundos y pudiendo llegar de 0 a 100 en un total de 600 segundos.

Por otra parte, la interfaz de ítems, permite alimentar a la mascota con distintos potenciadores que facilitarán los entrenamientos del usuarios. Siendo estos doble de experiencia, reducción de gasto energético al entrenar y doble ítem. Como se mencionó anteriormente, estos son entregados por cada uno de los entrenamientos, de esta manera el usuario tiene que pasar por los distintos ejercicios para conseguir ítems que se encuentran dentro de la aplicación (ver figura C.7 en anexos).

4.3.4. Galería de Mascotas

Finalmente, se encuentra la galería de mascotas. Esta funciona para guardar y mostrar las distintas mascotas entrenadas por los usuarios, donde se muestran los detalles generales de las mascotas, tales como su nombre, nivel, experiencia obtenida y total ejercicios realizados. También puede mostrar elementos más detallados, como las frutas que ocupo para mejorar los entrenamientos o cuál fue su ejercicio más jugado (ver figura C.8 en anexos).

4.4. Text To Speech y Speech To Text

Las funcionalidades de voz corresponden a los pilares fundamentales de la aplicación, ya que permite facilitar la comunicación con el usuario ya sea en los menús o en las instrucciones de los ejercicios. Por otra parte, existen actividades donde el input de los usuarios es mediante la voz por lo tanto una conversión a de Speech to Text (STT) se vuelve necesaria para la resolución de los problemas propuestos.

Por lo tanto, en el marco del desarrollo de la aplicación y su engine, se llevó a cabo una investigación exhaustiva para la implementación de las funcionalidades de Text-to-Speech (TTS) y Speech-to-Text (STT). Como resultado, se seleccionaron los servicios web de Amazon (AWS), integrando AWS Polly para la generación de voz, AWS Transcribe para el reconocimiento de voz, y AWS S3 para el almacenamiento en la nube.

Con esta finalidad para garantizar una arquitectura robusta y fácil de trabajar, se encapsuló los servicios de AWS como se puede apreciar en el diagrama de clases [10]. Donde AWSManager es el componente clave que coordina la interacción entre la aplicación y los servicios de AWS (Polly, Transcribe y S3). Encargándose de la inicialización de las clases de servicio, la gestión de las llamadas a sus funciones y la transferencia de los atributos requeridos. De esta manera las clases de AWS se centran exclusivamente en la comunicación con la nube y el procesamiento de datos, lo que facilita su desarrollo y mantenimiento.

Para el proceso de inicialización, se comienza validando los permisos de micrófono e internet, esenciales para la operación de las funcionalidades de voz. Tras la confirmación de los permisos, se incorpora el componente del servicio pertinente al Manager, quien gestiona la configuración de las credenciales de acceso (claves y región) para el servicio de AWS en cuestión. Con la conexión establecida, el sistema entra en un estado de espera de solicitudes desde la aplicación, listas para ser procesadas mediante las funcionalidades de TTS o STT (ver figura C.9 en anexos).

El sistema de generación de voz (TTS) se basa en una arquitectura que involucra las clases AWSManager y AWSPollyTTS de manera interna y las clases VoiceController y AudioManager para manejar el input y output Figura N°[50]. VoiceController actúa como un intermediario entre la aplicación y los servicios de AWS, recibiendo las solicitudes de texto. AWSManager recibe el texto desde VoiceController, y lo envía a Polly mediante el método SynthesizeSpeech. El archivo de audio resultante, en formato MP3, se almacena localmente y se reproduce a través de AudioManager, quien gestiona la reproducción y ofrece opciones de control, finalizado con el proceso y quedando en espera de una nueva llamada. Este proceso garantiza una gestión eficiente de las solicitudes de TTS y una correcta interacción con los servicios de AWS(ver figura C.10 en anexos).

El proceso de Speech-to-Text (STT) se diferencia de TTS en que requiere la interacción del usuario para iniciar y terminar la captura de audio mediante el micrófono

(Figura C.1 de anexos). La duración de la grabación está determinada por el tiempo que el usuario mantiene el botón presionado. El archivo de audio grabado como AudioClip, es transferido a AWSManager, quien ajusta el formato del clip a WAV y lo carga al bucket designado en el servicio S3 de Amazon. AWSManager solicita la transcripción del audio a AWS Transcribe, utilizando como parámetros el lenguaje, bucket donde se encuentra almacenado el audio, entre otros. Una vez recibida la transcripción en formato JSON, AWSTranscribe procesa el archivo, extrayendo el texto transcrito en formato string. Este texto es comparado con la respuesta correcta, y asignando la puntuación correspondiente(ver figura C.11 en anexos).

5. Implementación

En el presente capítulo se presentaran la implementación de las distintas funcionalidades del software desarrollado.

5.1. Ejercicios

Para esta sección es importante mencionar los dos enfoques principales, los ejercicios que piden distintas selecciones o aquellos que donde es necesario hablar para completar las preguntas. Es importante mencionar la clase `QuestionManager`, la cual se encarga de gestionar las distintas preguntas y los distintos ejercicios, dentro de esta clase es fundamental mencionar el método `QuestionSelection`, teniendo un comportamiento común para los distintos ejercicios, seleccionando la pregunta a utilizar en cada minijuego.

5.1.1. `QuestionManager`

En primer lugar en la escena "menú de selección de ejercicio", tiene la finalidad de manejar elección del ejercicio. Para eso el identificador (ID) del ejercicio seleccionado se almacena en un `ScriptableObject` para su posterior uso en la creación dinámica en la escena "ejercicio", instanciando el prefab correspondiente. Así mismo cuando se cambia ejercicio durante la selección, se desencadena una llamada a `VoiceController` para proporcionar retroalimentación auditiva al usuario y visual.

Con respecto al método `Start()`, además de ajustar los elementos base de ambas escenas anteriormente mencionadas, también tiene la función de invoca a `LoadQuestion()`, para el cargado preguntas(ver código en Anexo D.1). Por otra parte cuando se cambia de escena a la de ejercicio `start()` también se tiene que encargar de la

generación del canvas, aparir de diferentes prefas disponibles, comunicándose con el componente controlador de cada uno de los canvas “ExercisePrefaController” para entregar los elementos necesarios para su funcionamiento, para que se puede procesar su método `start()` respectivo(ver código en Anexo D.2).

La selección de preguntas se realiza de forma aleatoria desde una lista según el enfoque y la letra en trabajo (ver Anexo D.3). Para los minijuegos de “escuchar y seleccionar”, se eligen dos opciones incorrectas de las listas disponibles, excluyendo preguntas con la misma sílaba que la respuesta correcta (ver Anexo D.4). Tras cada respuesta, se invoca al método `CheckAnswer()`, que actualiza el tiempo restante o el puntaje, incrementa los contadores de preguntas realizadas y correctas, y selecciona una nueva pregunta (ver Anexo D.5).

Al finalizar una sesión de entrenamiento, el sistema inicia una serie de llamadas que gestionan la transición a la escena de resultados, donde se presenta al usuario un resumen de su desempeño (ver código en Anexo D.6). Esta secuencia desencadena, a su vez, la llamada al método `Training` de la mascota, el cual establece la cantidad de experiencia ganada, otorga ítems, desactiva los buffs activos y calcula las subidas de nivel con la experiencia máxima requerida para el siguiente (ver códigos en Anexos D.7 a D.11). De esta manera, la escena puede mostrar los elementos claves como ítems obtenidos y la energía consumida, teniendo la opción de regresar a la pantalla principal o iniciar un nuevo entrenamiento.

5.1.2. ExercisePrefaController

Se enfoca en la gestión del comportamiento de los diferentes ejercicios. Para ello, establece una comunicación con `QuestionManager` para controlar los aspectos relacionados con las preguntas y cierre. Por otra parte, la inicialización de `ExercisePrefaController` se realiza a través de cuatro métodos distintos.

`FonenGrafSettings()` se encarga de distribuir aleatoriamente las respuestas en los botones de interacción (ver código en Anexo D.12). Una vez que el usuario selecciona una opción, se verifica la etiqueta (tag) del botón para determinar si la respuesta es correcta, enviando un valor booleano a `QuestionManager` para su procesamiento (ver código en Anexo D.13).

Se implementa un método que centraliza la gestión de estos elementos, ya que la mayoría son compartidos por los diferentes tipos de ejercicios. En el caso de Fo-

nemaFonemaSetting() y GrafemaFonemaSetting(), el ajuste se limita a los textos necesarios, principalmente el texto de puntaje. Por otra parte, GrafemaFonemaSetting() realiza ajustes en sus textos específicos.

Para la gestión del botón de micrófono se configura mediante los métodos OnPointerDown() y OnPointerUp(). El primero invoca a StartListening() y activa una bandera isRecording para evitar grabaciones repetidas (ver Anexo D.14). El segundo, al liberar el botón, desactiva isRecording y llama a StopListening() para finalizar la grabación e iniciar la transcripción (ver Anexo D.15). Una vez completada la transcripción, se invoca a SpeechAnswer(), que envía un valor booleano a QuestionManager para verificar la respuesta (ver Anexo D.16).

Finalmente, al terminar una sesión de entrenamiento, se invoca al método InitializeSliderAndTexts(), que ajusta la barra de energía del jugador y muestra información sobre el tipo y la cantidad de fruta obtenida, proporcionando retroalimentación visual al usuario mediante texto e íconos, y se complementa con retroalimentación auditiva a través del método startSpeaking() de la clase VoiceController. En caso de querer reintentar, se vuelve a iniciar todo el proceso de nuevo y en caso de terminar, el usuario es devuelto al menú principal del juego.

5.2. SaveSystem y Load

El sistema de persistencia de datos se articula en torno a tres funcionalidades clave: (1) el almacenamiento de la totalidad de las mascotas en el almacenamiento persistente de la aplicación a través de la invocación de su método correspondiente; (2) la carga de la totalidad de las mascotas existentes en el sistema, así como la carga de las preguntas cuando se requiera; y (3) la eliminación de los archivos de mascotas en puntos específicos del programa.

Para la implementación del almacenado de mascotas, el método SaveMascot(), recibe la lista de mascotas existentes desde GameManager. Para completar este procedimiento los elementos de la lista de mascotas se convierten en el objeto MascotData para su posterior serialización en formato JSON individual, tal como se detalla en el código adjunto (ver código en Anexo D.17). Almacenando datos claves como estadística, identificadores, entre otros. Por otra parte el proceso de carga de datos persiste en la aplicación se realiza de forma inversa al guardado. Se examina la cantidad de archivos JSON de mascotas almacenados. Cada archivo se deserializa en un objeto

MascotData y posteriormente se instancia un objeto Mascot con los atributos correspondientes(ver código en Anexo D.18). La lista de instancias de Mascot resultante se entrega a GameManager para que pueda extraer la mascota activa y mostrar las mascotas existentes en la galería. Finalmente un proceso similar se aplica a para cargar las preguntas dentro de la aplicación, cuya información se entrega a QuestionManager(ver código en Anexo D.19).

Finalmente, se implementa el método DeleteAllSaveFiles() para la eliminación de la totalidad de los datos persistentes de las mascotas. Este método se ejecuta en dos situaciones específicas: (1) al inicio de la aplicación, para garantizar un estado inicial consistente, sin mascotas con datos predeterminados en caso de existir; y (2) durante el proceso de creación de una nueva mascota, ya sea porque existe una mascota con atributos predeterminados al final del proceso o porque el usuario decide cancelar la creación de una mascota. La ejecución de DeleteAllSaveFiles() asegura la eliminación de los archivos de mascotas, permitiendo la actualización de la lista de mascotas con los cambios realizados y la eliminación de la mascota en curso del sistema.

5.3. Audio y voz

5.3.1. AudioManager

Este objeto se centraliza en la gestión de todos los aspectos relacionados con el audio de la aplicación, incluyendo la manipulación de AudioClips, el control del micrófono, la configuración del AudioMixer y la gestión de los AudioSources. Esta clase provee una capa de abstracción que facilita la interacción con el sistema de audio, simplificando el desarrollo y el mantenimiento

Text To Speech

Para manejar la salida del TTS por parte del sistema, AudioManager implementa 4 métodos: dos para manejar el flujo de la reproducción y dos para controlar el estado de la reproducción. Para la reproducción del clip, se dispone del método PlayAudioClip(), el cual es proporcionado a AWS Polly para la reproducción del audio una vez finalizado el proceso de conversión. Así mismo este método incorpora una co-rutina para monitorizar el estado de la reproducción del audio, permitiendo un control preciso del proceso(ver código en Anexo D.20). El cual se complementa

con `isPlayingAudio()`, permitiendo un control preciso del estado de reproducción del audio (ver código en Anexo D.21). Esta combinación de métodos evita la interrupción abrupta de clips de audio por nuevas solicitudes, asegurando una reproducción fluida. Por otra parte, la reproducción de un nuevo clip se inicia una vez finalizada la reproducción del clip actual, sin depender de temporizadores. Finalmente, `stopPlayingAudio()` fuerza la finalización de la reproducción, lo que resulta útil en situaciones donde el usuario tiene control del flujo, como durante el entrenamiento o la selección de opciones.

Speech To Text

En el ámbito de STT, `AudioManager` define tres métodos esenciales: `StartRecording()`, `StopRecording()` y `GetRecording()`. `StartRecording()` inicializa la grabación de audio seleccionando el micrófono predeterminado y almacenando la grabación en la variable `recordedClip`. `StopRecording()` finaliza la grabación cerrando el micrófono. `GetRecording()` proporciona el `AudioClip` grabado a `VoiceController` para iniciar el proceso de transcripción (ver código en Anexo D.22).

5.3.2. VoiceController

Esta clase actúa como un intermediario clave entre la aplicación y `AWSManager`, gestionando las solicitudes de audio (tanto de entrada como de salida) en colaboración con `AudioManager`. Para ello, `VoiceController` coordina las funcionalidades de Text-to-Speech (TTS) y Speech-to-Text (STT) entre el sistema y AWS. Previo a cualquier operación relacionada con el micrófono, `VoiceController` invoca a `PermissionManager()` para verificar y solicitar los permisos de micrófono necesarios.

Tras la obtención de los permisos requeridos, se procede a la generación de las solicitudes necesarias, según el contexto del juego. Para la funcionalidad de Text-to-Speech (TTS), se utiliza el método `startSpeaking()`. Este método se asegura que solo se realice una solicitud por llamada y que la instancia de `AWSManager` esté disponible. Una vez verificados estos elementos, se invoca a `AWSManager` con el texto a convertir a audio y el método `playAudioClip()` de `AudioManager`. De esta forma, una vez finalizado el proceso de conversión, se reproduce el clip de audio obtenido desde el servidor (ver código en Anexo D.23). Por otra para la funcionalidad de Speech-to-Text (STT), `VoiceController` implementa dos métodos principales:

StartListening() y StopListening(). StartListening() se asegura de la existencia de AWSManager y solicita a AudioManager iniciar la grabación a través del micrófono. StopListening() invoca el método StopRecording() de AudioManager para finalizar la grabación. GetRecording() del AudioManager recupera el AudioClip grabado y lo entrega a AWSManager para iniciar el proceso de transcripción, utilizando el método OnFinalSpeechResult() para almacenar la cadena de texto transcrito(ver código en Anexo D.25).

5.4. Amazon Web Service AWS

5.4.1. AWSManager

La clase AWSManager se implementa como un componente del objeto voiceController, cuya instancia persiste durante la ejecución de la aplicación. Esta implementación evita la necesidad de invocar los servicios de AWS en cada cambio de escena, optimizando el rendimiento. Al iniciar la aplicación, AWSManager se encarga de agregar los componentes AWSPollyTTS, AWSTranscribeSTT y AWSS3 a su instancia, inicializándolos con las credenciales y la región correspondiente(ver código en Anexo D.26). Así mismo esta clase también coordina las llamadas a cada uno de ellos según los requerimientos del proceso. Para llevar a cabo sus funciones, AWSManager utiliza clases de utilidad como Networkchecker y AudioSaver. En el caso Networkchecker garantiza la disponibilidad de conexión a Internet antes de realizar cualquier solicitud a los servicios de AWS (ver código en Anexo D.27). Por su parte, AudioSaver se encarga de almacenar los clips de audio capturados por el micrófono en un formato compatible con los servicios de AWS para su posterior procesamiento, siendo convertToWav() el método más importante del objeto el cual es llamado por un método superior saveAudioClip(), que recibe y prepara los elementos para la conversión(ver código en Anexo D.28).

5.4.2. AWSPolly

El proceso de generación de voz (TTS) con AWSPollyTTS se inicia tras la verificación de la conectividad a la red mediante Networkchecker. Una vez confirmada la conexión, se invoca el método específico de AWSPollyTTS para generar la solicitud a AWS Polly. Esta solicitud incluye el texto a convertir, las características de la voz

(como el idioma, la voz específica y el estilo), y el formato de audio deseado, considerando las opciones disponibles para la región y el formato(ver código en Anexo D.29). Tras generar la solicitud, el sistema espera la respuesta de AWS Polly(ver código en Anexo D.30). Si la respuesta es exitosa, se recibe el clip de audio generado escribiendo el archivo en la `persistentData` de la aplicación(ver código en Anexo D.31)y procede a reproducir mediante el método `PlayingAudioClip()` de `AudioManager`.

5.4.3. AWSTranscribe y AWSS3

Así mismo para AWS transcribe, Tras la finalización de la grabación de audio, el archivo es transferido a `AWSManager`. `AWSManager` emplea `AudioSaver` para almacenar el archivo en el almacenamiento persistente de la aplicación, asegurando el formato correcto para su procesamiento. Posteriormente, `AWSManager` coordina con `AWSS3` la carga del archivo a la nube a través del método `UploadFileAsync()`. Una vez cargado el archivo, `AWSManager` inicia el proceso de transcripción con AWS Transcribe, proporcionando la ubicación del archivo en S3 y otros parámetros relevantes(ver código en Anexo D.35).

Una vez preparado el archivo de audio, se invoca a `AWSTranscribeSTT` para iniciar el proceso de transcripción con AWS Transcribe. Se proporciona a AWS Transcribe la URL del archivo en S3, el idioma del audio y el bucket en el que se encuentra almacenado. `AWSTranscribeSTT` genera la solicitud correspondiente y la envía a AWS Transcribe(ver código en Anexo D.32). Una vez generada la solicitud de transcripción, el sistema espera la respuesta de AWS Transcribe. Si se produce un error durante el proceso, se registra un mensaje de error mediante `debug.error()`(ver código en Anexo D.33). En caso de una transcripción exitosa, se procesa el archivo JSON resultante. La información relevante se extrae del JSON y se utiliza para poblar una instancia de la clase interna `transcription`, que contiene la cadena de texto transcrito(ver código en Anexo D.34). Ya con todo el proceso terminado, se devuelve el resultado a la instancia de `VoiceController` que solicitó el proceso y se revisa la respuesta por parte `QuestionManager`.

6. Análisis y Resultados

En el presente capítulo se detallan los resultados y el análisis de las pruebas de usabilidad del prototipo. Estas evaluaciones se llevaron a cabo con 22 estudiantes de primer año básico del Instituto Linares. Los participantes interactuaron con la aplicación en grupos de tres niños durante siete minutos, tras lo cual completaron un cuestionario. Este instrumento, diseñado a partir del estudio de arte previo, que se centró en la usabilidad y los elementos gráficos, empleando la escala de Wong-Baker Faces Pain Scale (donde 1 indica 'muy triste' y 5 'muy feliz') para estandarizar las respuestas.

6.1. Resultados

A continuación, se presentan las preguntas de la encuesta aplicada, referenciando las preguntas con la abreviatura “P.X”, como se puede ver en el cuadro 6.1, las cuales se encuentran enfocadas en 3 puntos la percepción general de la aplicación enfocada en las motivación para continuar jugando, la usabilidad y el arte.

P.1	¿El juego te hizo querer practicar más?
P.2	¿El juego te dejó hacer lo que querías?
P.3	¿Qué tan difícil fue el juego de seleccionar sílabas?
P.4	¿Qué tan difícil fue el juego de leer y hablar?
P.5	¿Qué tan difícil fue el juego de escuchar y hablar?
P.6	¿Te sirvieron las instrucciones con audio?
P.7	¿Cómo te sentiste al terminar la sesión de entrenamiento?
P.8	¿Te intereso poder jugar con otras mascotas?
P.9	¿Te gustó la mascota con la que jugaste?
P.10	¿Te gustó la música de la mascota?
P.11	¿Te gustó la casa de la mascota?

Cuadro 6.1: Preguntas encuestas

Por otra parte, en los siguientes cuadros se presenta una síntesis de las principales estadísticas descriptivas obtenidas para cada pregunta previamente presentada, segregada por la muestra general, masculina y femenina. Asimismo, los valores individuales por encuesta se encuentran disponibles en el Anexo E.2 (Tabla de resultados). Para los propósitos de esta investigación, se estableció que un nivel de satisfacción del usuario, se estableció que un nivel de satisfacción del usuario se consideraría “bueno” si el promedio de las respuestas en la escala de Wong-Baker (1 = muy triste, 5 = muy feliz) se situara en un rango de 3.5 a 4.0. Este rango indica que, en promedio, los participantes expresaron sentimientos que van desde la “indiferencia” hasta la “alegría” (valor 4 en la escala). Este criterio permite estandarizar la percepción de comodidad y disfrute durante el uso del prototipo, siendo un indicador clave de una usabilidad y experiencia de usuario adecuadas para el público objetivo.

Datos	P.1	P.2	P.3	P.4	P.5	P.6	P.7	P.8	P.9	P.10	P.11
Promedio	4,68	3,86	4,50	4,09	3,82	4,09	3,86	4,77	4,68	4,68	4,32
Moda	5	5	5	5	5	5	5	5	5	5	5
Mediana	5	4	5	5	5	5	5	5	5	5	5
Desviación E.	0,57	1,28	0,74	1,27	1,50	1,41	1,55	0,75	0,64	0,71	0,99

Cuadro 6.2: Tabla de general resultados

En primer lugar, el promedio general de las respuestas indica una percepción

predominantemente positiva del prototipo. Específicamente, las Preguntas 1 y 8, centradas en la motivación para seguir jugando y el interés en probar más mascotas, respectivamente, presentando los promedios más elevados. La moda y la mediana refuerzan esta tendencia positiva en la mayoría de los ítems evaluados. En cuanto a la desviación estándar, esta revela un acuerdo relativo entre los participantes en sus respuestas, aunque con una dispersión más notable en las Preguntas 5 y 7.

Datos	P.1	P.2	P.3	P.4	P.5	P.6	P.7	P.8	P.9	P.10	P.11
Promedio	4,64	3,18	4,45	4,09	3,73	3,55	3,55	4,55	4,45	4,36	4,36
Moda	5	3	5	5	5	5	5	5	5	5	5
Mediana	5	3	5	5	4	4	4	5	5	5	5
Desviación E.	0,67	1,40	0,69	1,30	1,42	1,63	1,63	1,04	0,82	0,92	1,12

Cuadro 6.3: Tabla resultados niños

Para la muestra masculina, el promedio más bajo se registró en la Pregunta 2, la cual aborda la percepción de libertad del usuario, convirtiéndose en un punto clave para el análisis. Con respecto a las modas y medianas, aunque mayormente positivas en el resto de los ítems, también señalan la Pregunta 2 como el valor más bajo dentro de este grupo. La desviación estándar para los niños indica un acuerdo general en sus respuestas, si bien se observa cierta variabilidad en ítems específicos.

Datos	P.1	P.2	P.3	P.4	P.5	P.6	P.7	P.8	P.9	P.10	P.11
Promedio	4,73	4,55	4,55	4,09	3,91	4,64	4,18	5,00	4,91	5,00	4,27
Moda	5	5	5	5	5	5	5	5	5	5	5
Mediana	5	5	5	4,5	5	5	5	5	5	5	4,5
Desviación E.	0,47	0,69	0,82	1,30	1,64	0,92	1,47	0,00	0,30	0,00	0,90

Cuadro 6.4: Tabla resultados niñas

En contraste, para la muestra femenina, los promedios se mantuvieron consistentemente altos, indicando una satisfacción general elevada. Las medianas más bajas se observaron en las Preguntas 4 y 11, enfocadas en los juegos de voz y el arte del *background*, respectivamente, ambas con un valor de 4.5. La moda predominante de 5 en la mayoría de las preguntas sugiere una alta satisfacción por parte de este sector demográfico. Además, la desviación estándar muestra una menor dispersión en las

respuestas de las niñas en comparación con los niños, llegando incluso a casos de unanimidad, como en la Pregunta 10.

6.2. Análisis

En base a los resultados mencionados anteriormente, se han agrupando las preguntas en tres ejes temáticos: percepción general, usabilidad y arte. Cada pregunta siendo examinada de forma global y diferenciada por género, con el objetivo de identificar las posibles diferencias en las respuestas entre el público masculino, femenino, para llegar a un análisis global del prototipo.

6.2.1. Análisis Percepción general

Las preguntas que se analizaran en esta sección corresponden a aquellas que abordan la percepción general por parte de los menores siendo estas las preguntas P.1, P.7 y P.8 del cuadro 6.1 y sus respectivos gráficos a partir de de anexos D.2 gráficos.

La Pregunta 1 reveló un alto interés general en continuar utilizando la aplicación, con un 73 % de los participantes identificando su sentir con la categoría “alegría” (Escala Wong-Baker), un 23 % con “felicidad” y un 4 % con “indiferencia”. Este último valor se concentró en el grupo masculino, sugiriendo una menor motivación inicial hacia la aplicación en algunos niños.

Este patrón se corroboró en la Pregunta 7, donde el promedio general indicó una tendencia hacia la “indiferencia” al finalizar la sesión con un 14 %, junto con un 14 % que seleccionó la cara representativa de “pena” (valor 1) y un 4 la de “tristeza” (valor 2). Al analizar por género, se constató que la mayoría de las respuestas de “indiferencia”, “tristeza” y “pena” provinieron del grupo masculino (5 niños en total), en contraste con 3 niñas. Esta diferencia podría atribuirse a la predisposición inicial de los niños o a elementos específicos del juego que no resonaron con este grupo.

No obstante, la Pregunta 8, referente al interés en interactuar con otras mascotas virtuales, arrojó resultados positivos a nivel general, con la mayoría de los usuarios identificándose con la cara de “alegría”. Solo dos niños manifestaron “indiferencia” o “tristeza”, mientras que las niñas mostraron unanimidad en su respuesta de “alegría”.

En conclusión, si bien las niñas demostraron una aceptación inicial ligeramente superior hacia la aplicación en comparación con los niños, la implementación de mas-

cotas virtuales personalizables (nombres definidos por los usuarios y comunicación directa por el nombre del entrenador) evidenció un potencial para fomentar la adopción y el compromiso con el software en ambos grupos. La personalización pareció generar un sentido de pertenencia hacia las mascotas virtuales entre los participantes.

6.2.2. Análisis Usabilidad

A continuación se analizaran las preguntas correspondientes a la usabilidad de la aplicación siendo estas P.2, P.3, P.4, P.5 y P.6. De las cuales desde P.3 hasta P.5 corresponden a las distintas actividades que el usuario puede realizar, mientras que tanto P.2 como P.6 corresponden a usabilidad general de la misma.

En relación con la usabilidad de la aplicación, el promedio de las respuestas tendió hacia la “indiferencia”, aunque con una inclinación notable hacia la “alegría” en la percepción general de la facilidad de uso. Al analizar las respuestas por género, se observó que los niños se concentraron mayoritariamente en la opción “indiferente”, presentando además una mayor dispersión en sus valoraciones. En contraste, las niñas se inclinaron más hacia la “felicidad”, mostrando una mayor consistencia en sus respuestas. A pesar de esta diferencia, un 73 % del total de los participantes manifestó sentirse cómodo utilizando la aplicación.

Los datos de la Pregunta N°6, concerniente a la utilidad del sistema de text-to-speech para guiar al usuario en la aplicación y los ejercicios, reflejaron un patrón similar al de la Pregunta 1. A nivel general, la mayoría de los niños (59 %) percibieron las instrucciones por audio como útiles (“alegría”), seguido por un 18 % que se sintió “feliz”. El 23 % restante expresó “indiferencia” o consideró que las instrucciones no fueron de ayuda. Esta distribución sugiere una posible menor predisposición de los niños a seguir las indicaciones ya sea auditivas o por texto, así como la existencia de elementos en la implementación del text-to-speech que no conectaron completamente con este grupo demográfico.

A pesar de estas observaciones, la percepción general de la usabilidad fue positiva, indicando una afinidad general de los niños con el manejo de la aplicación. Sin embargo, los datos sugieren áreas de mejora en el diseño de la interfaz, como el realce visual y auditivo de los botones mediante animaciones y sonidos, con el objetivo de optimizar la experiencia de usuario, especialmente para el público masculino.

En cuanto a los ejercicios de selección de sílabas (Pregunta 3), los participantes

mostraron una facilidad de uso de la aplicación, así como una buena aceptación por parte de los niños. El promedio de respuestas indicó una tendencia hacia la “felicidad” en ambos grupos demográficos, con una baja dispersión de datos y un 63 % de respuestas “alegres” y un 23 % “felices”, recalcando lo anteriormente mencionado, por lo tanto esto sugiere una buena aceptación de este tipo de tarea por parte de este grupo etario.

En contraste, los ejercicios que requerían que los niños hablaran (Preguntas 4 y 5) generaron una mayor incomodidad. Las tablas comparativas y los gráficos revelaron una menor tasa de aceptación en estas tareas. Si bien un porcentaje mayoritario aún seleccionó la opción “alegría”, la proporción de respuestas que se identificaron con “indiferencia” o valores inferiores fue significativamente mayor en comparación con el ejercicio de selección de sílabas, identificándose como un área a mejorar.

Esta incomodidad se atribuye principalmente a la demora de aproximadamente 40 segundos en el proceso de transcripción de voz. Durante las pruebas, se observó que los niños reaccionaban con duda, llegando a pensar que la aplicación se había detenido. Esta problemática podría mitigarse mediante una optimización en la comunicación con el servidor o una mejora en la infraestructura de servidores que se utilizo. Adicionalmente, la incorporación de más retroalimentación visual y auditiva podría indicar al niño el estado de carga de las respuestas y reducir la incertidumbre.

A pesar de esta deficiencia específica en los ejercicios de voz, la usabilidad general de la aplicación se percibió positivamente, facilitada tanto por las guías de voz como por la interfaz. Así mismo, la existencia de un constante uso de tablets en el entorno escolar como elemento educacional podría haber contribuido a esta facilidad de manejo. No obstante, el tiempo de respuesta de los ejercicios de voz y la necesidad de integrar más elementos visuales (animaciones y guías adicionales) se identifican como puntos clave para futuras mejoras en el diseño de la aplicación.

6.2.3. Análisis elementos visuales y música

Para finalizar se analizaran las preguntas P.9, P.10 y P.11 que abordan los elementos estéticos de la aplicación, siendo estos la mascota, la música de fondo y el background respectivamente.

El análisis de los ítems correspondientes al eje artístico revela una alta tasa de aprobación general por parte de los usuarios, particularmente en los elementos de

diseño de personajes (mascotas) y la experiencia sonora de la aplicación. La baja desviación estándar observada en estos aspectos sugiere una consistencia en la percepción positiva, en contraste con el diseño del fondo (background), que presentó una mayor variabilidad en las respuestas.

La pregunta referente a la música demostró ser particularmente exitosa, con un 80 % de los usuarios expresando “alegría” (Escala Wong-Baker), un 4 % “felicidad” y un 14 % “indiferencia”. Es destacable la respuesta unánime de las niñas hacia la opción de “alegría” en este ítem, lo que subraya una resonancia positiva del diseño sonoro con el público femenino.

De manera similar, la pregunta sobre las mascotas virtuales evidenció una fuerte aprobación general, con un 77 % de los usuarios identificándose con “alegría”, un 14 % con “felicidad” y un 9 % con “indiferencia”. Al igual que con el sonido, una mayoría significativa de las niñas expresó “alegría” con su mascota, mientras que se observó una mayor diversidad de respuestas (felicidad o indiferencia) en un grupo reducido de niños.

En contraste, el diseño del fondo (background) generó una respuesta menos homogénea, con un 64 % de “alegría”, acompañado de una mayor proporción de respuestas “indiferentes” o “felices”. Esta variabilidad sugiere que el background podría haber pasado más desapercibido para los niños o no generó una reacción emocional tan marcada como los personajes y la música.

En conclusión, los elementos artísticos del juego fueron bien recibidos por la mayoría de los menores, destacándose una afinidad particular del público femenino hacia el diseño de las mascotas y el apartado sonoro. Si bien el background generó una respuesta menos intensa, la recepción general de los tres aspectos evaluados puede considerarse exitosa en términos de diseño de experiencia de usuario y estética del videojuego. La marcada diferencia en la respuesta femenina sugiere que los elementos de diseño implementados resonaron especialmente con este grupo demográfico.

6.2.4. Análisis final

El análisis integral de la percepción general, usabilidad y elementos artísticos de la aplicación, basado en las respuestas de los usuarios y las observaciones durante las pruebas, permite concluir que el prototipo resultó exitoso en términos de significancia, autonomía, curiosidad y atractivo audiovisual. La mayoría de los aspectos

fueron evaluados positivamente por los participantes, especialmente por las niñas, evidenciándose incluso interés en descargar la aplicación para uso personal. Si bien la usabilidad fue bien valorada en general, el área de transcripción de voz presenta oportunidades de mejora para optimizar la experiencia de todos los usuarios y mitigar la latencia de respuesta. Finalmente, el diseño artístico, incluyendo los elementos visuales, el apartado sonoro y la voz del text-to-speech, generó una alta satisfacción, con sugerencias espontáneas de los niños para la inclusión de mascotas adicionales como perros, gatos o peces. En resumen, el prototipo demostró un buen nivel de aceptación, con áreas específicas de refinamiento identificadas para enriquecer aún más la experiencia del usuario.

7. Conclusión

Al finalizar este documento, se concluye por parte del autor, que los objetivos generales y específicos planteados para este proyecto y prototipo se cumplieron satisfactoriamente. Esto abarca desde la investigación inicial necesaria para el desarrollo adecuado del prototipo, la creación de los diversos juegos y la selección de palabras utilizadas, hasta la creación de un producto con nivel de madurez tecnológica TRL3, el cual fue probado en el Instituto Linares, ubicado en la ciudad de Linares, región del Maule.

Durante el desarrollo, se implementó una estructura modular para los diferentes elementos del juego, lo que facilitó la adición o modificación de palabras y juegos. Un ejemplo de esta adaptabilidad fue la implementación de la funcionalidad speech-to-text. Inicialmente, se planeó un sistema basado en tiempo, pero debido a la latencia del sistema de transcripción, se optó por un sistema de puntaje. Asimismo, las limitaciones del servicio de transcripción de AWS llevaron a modificar las preguntas, requiriendo que el usuario pronunciara palabras completas en lugar de sílabas, ya que el sistema no reconocía estas últimas.

Por otra parte, La orientación proporcionada por la profesora guía fue fundamental en el desarrollo del prototipo, permitiendo el diseño de una interfaz simple, atractiva y de fácil uso para el público objetivo (niños de 6 a 8 años o personas con dificultades del habla). Su conocimiento y experiencia contribuyeron a la creación de un producto accesible, facilitando la interacción con los elementos clave del juego y logrando que las mascotas fueran atractivas para los usuarios. Esto, a su vez, potenció la efectividad de las herramientas para alcanzar los objetivos pedagógicos establecidos.

Con respecto a las pruebas realizadas, se evidenciaron un alto interés de los niños por aprender a través de juegos, tanto físicos como digitales. reforzando la idea

de que las mascotas virtuales pueden influir positivamente en el comportamiento de los usuarios, como se observó en las conversaciones sobre sus mascotas y los nombres después de las pruebas. Esto sugiere que los videojuegos tienen un impacto potencial como método educativo, principalmente como herramienta de apoyo para complementar los conocimientos impartidos. La utilización de guías textuales y de voz demostró ser eficaz para explicar las actividades y los elementos de la aplicación, minimizando la necesidad de explicaciones directas durante las pruebas.

Para la evolución de este prototipo y el desarrollo de productos futuros, se identifican varias áreas clave de mejora. La reducción de la latencia del servidor en los juegos de reconocimiento de voz se considera primordial, lo que permite comprender la prevalencia de modelos de suscripción en soluciones similares analizadas en el estado del arte. Asimismo, la integración de elementos de interfaz que proporcionen retroalimentación sobre el estado del sistema, como la espera de transcripción, o que ofrezcan guías visuales (animaciones) para asistir a los usuarios, se considera fundamental para optimizar la experiencia. Por otra parte, atendiendo a las sugerencias de los usuarios, la incorporación de nuevas mascotas (perro, gato, pez) podría aumentar el atractivo del juego, permitiendo la posibilidad de desarrollar proyectos derivados que expandan el entrenamiento a otras sílabas, explorando aún más la fauna chilena. Aunque se descartó para este prototipo por limitaciones de tiempo y organización, la adición de más juegos y palabras trisílabas representa una vía para enriquecer la oferta y la variedad de la aplicación en futuras iteraciones.

En retrospectiva, la elección de JSON como formato de almacenamiento se considera acertada debido a su flexibilidad y facilidad de modificación, lo que agilizó la gestión de palabras y la creación de mascotas durante el desarrollo y las pruebas. Así mismo, permitió almacenar los datos esenciales de cada mascota de manera compacta en la nube. La selección de AWS como proveedor de servicios en la nube también se evalúa positivamente por su fácil integración con Unity y la amplia documentación disponible para los servicios S3, Transcribe y Polly. Si bien la latencia de Transcribe afectó los tiempos de los ejercicios de voz, AWS ofreció una funcionalidad superior a otras alternativas gratuitas probadas. Si bien se deseaba explorar otros servicios como Google Cloud o Azure, la limitación económica dichas pruebas.

Finalmente, en base a los resultados obtenidos, su análisis y el trabajo realizado a lo largo de este documento, se vislumbra un amplio potencial de trabajo futuro para la aplicación, donde existe un público interesado en utilizar software para el trabajo

de la conciencia fonológica con el acompañamiento de una mascota virtual. Además, se considera fundamental implementar una integración con un software que permita a profesionales y/o tutores visualizar los resultados y el progreso de los menores. Esta funcionalidad, junto con las mejoras técnicas previamente mencionadas, será clave para ampliar el marco de utilización y la efectividad de la aplicación.

Bibliografía

- [1] Crespo Cruces, Marcelo Cárdenas San Martín, et al. Software de atención médica para el tratamiento del trastorno específico del lenguaje (tel) a nivel fonológico y morfosintáctico en niños de 3 a 5 años, utilizando videojuegos. 2015.
- [2] Raúl Gutiérrez Fresneda and Antonio Díez Mediavilla. Conciencia fonológica y desarrollo evolutivo de la escritura en las primeras edades. *Educacion XX1*, 21:395–416, 2018.
- [3] Centro de perfeccionamiento experimentos e investigación pedagógicas. ¿qué es la conciencia fonológica y cómo se desarrolla en niños?
- [4] Sylvia Defior. La conciencia fonológica y la adquisición de la lectoescritura phonological awareness and learning to read and write. *Infancia y Aprendizaje*, 17:91–113, 09 1994.
- [5] Sylvia Defior and Francisca Serrano. La conciencia fonémica, aliada de la adquisición del lenguaje escrito. *Revista de logopedia, foniatría y audiolología*, 31(1):2–13, 2011.
- [6] Liz Ysla Almonacid and Vicenta Ávila Clemente. La evaluación del conocimiento metalingüístico en niños del Último ciclo de la educación infantil peruana. *Revista Iberoamericana de Evaluación Educativa*, 10(1), abr. 2017.
- [7] Alejandra Medina. Enseñar a leer y a escribir: ¿en qué conceptos fundamentar las prácticas docentes? *Psykhé (Santiago)*, 15:45 – 55, 11 2006.
- [8] E Pereiro. Metodología para la adquisición de una correcta conciencia fonológica. In *Lingüística clínica y neuropsicología cognitiva. Actas del Primer Congreso Nacional de Lingüística Clínica*, 2006.

- [9] Jéssica Araya Ramírez. Los principios de la conciencia fonológica en el desarrollo de la lectoescritura inicial. *Revista de Lenguas Modernas*, (30), sep. 2019.
- [10] Ma del Rosario Ortiz and Juan E. Jiménez. Prereaders' early conceptions about written language. *Journal for the Study of Education and Development*, 24(2):215–231, 2001.
- [11] Laura Bosch. Los trastornos fonológicos en el niño*. *Revista de Logopedia, Foniatría y Audiología*, 7(4):195–200, 1987.
- [12] C. J. Coloma Tirapegui and Z. De Barbieri Ortiz. Trastorno fonológico y conciencia fonológica en preescolares con trastorno específico del lenguaje. *Revista de Logopedia, Foniatría y Audiología*, 27(2):67–73, 2007.
- [13] Noris Moreno-Flagge. Trastornos del lenguaje. diagnóstico y tratamiento. *Rev Neurol*, 57(Supl 1):S85–94, 2013.
- [14] AEJ Van Gaalen, J Schönrock-Adema, RJ Renken, ADC Jaarsma, JR Georgiadis, et al. Identifying player types to tailor game-based learning design to learners: Cross-sectional survey using q methodology. *JMIR Serious Games*, 10(2):e30464, 2022.
- [15] Yen Ru Shi and Ju Ling Shih. Game factors and game-based learning design model. *International Journal of Computer Games Technology*, 2015, 2015.
- [16] Jan L. Plass, Richard E. Mayer, and Bruce D. Homer. Handbook of game-based learning, 2019.
- [17] Kristian Kiili, Sara De Freitas, Sylvester Arnab, and Timo Lainema. The design principles for flow experience in educational games. *Procedia Computer Science*, 15:78–91, 2012.
- [18] Daria Bylieva, Nadezhda Almazova, Victoria Lobatyuk, and Anna Rubtsova. Virtual pet: trends of development. In *The 2018 International Conference on Digital Science*, pages 545–554. Springer, 2019.
- [19] Sun Joo Ahn, Kyle Johnsen, Tom Robertson, James Moore, Scott Brown, Amanda Marable, and Aryabrata Basu. Using virtual pets to promote physical activity in children: An application of the youth physical activity promotion model. *Journal of health communication*, 20(7):807–815, 2015.

- [20] Sahara Byrne, Geri Gay, JP Pollack, Amy Gonzales, Daniela Retelny, Theodore Lee, and Brian Wansink. Caring for mobile phone-based virtual pets can influence youth eating behaviors. *Journal of Children and Media*, 6(1):83–99, 2012.
- [21] Kyle Johnsen, Sun Joo Ahn, James Moore, Scott Brown, Thomas P Robertson, Amanda Marable, and Aryabrata Basu. Mixed reality virtual pets to reduce childhood obesity. *IEEE transactions on visualization and computer graphics*, 20(4):523–530, 2014.
- [22] Orlando Zafra Galvis. Tipos de investigación. *Revista Científica General José María Córdova*, 4(4):13–14, 2006.
- [23] Roberto Hernández Sampieri, Carlos Fernández Collado, Pilar Baptista Lucio, et al. *Metodología de la investigación*, volume 4. McGraw-Hill Interamericana México, 2018.

A. Anexo estado del Arte

Imágenes de referencia acerca del estado de arte

A.1. Abrapalabra



Figura A.1: Cuentin y el Mago Nombro.



Figura A.2: Selección de mundo.



Figura A.3: Ejemplo Actividad.

A.2. Mi primera encarta



Figura A.4: Menú Pincipal.



Figura A.5: Ejemplo Actividad.

A.3. ArbolABC.com

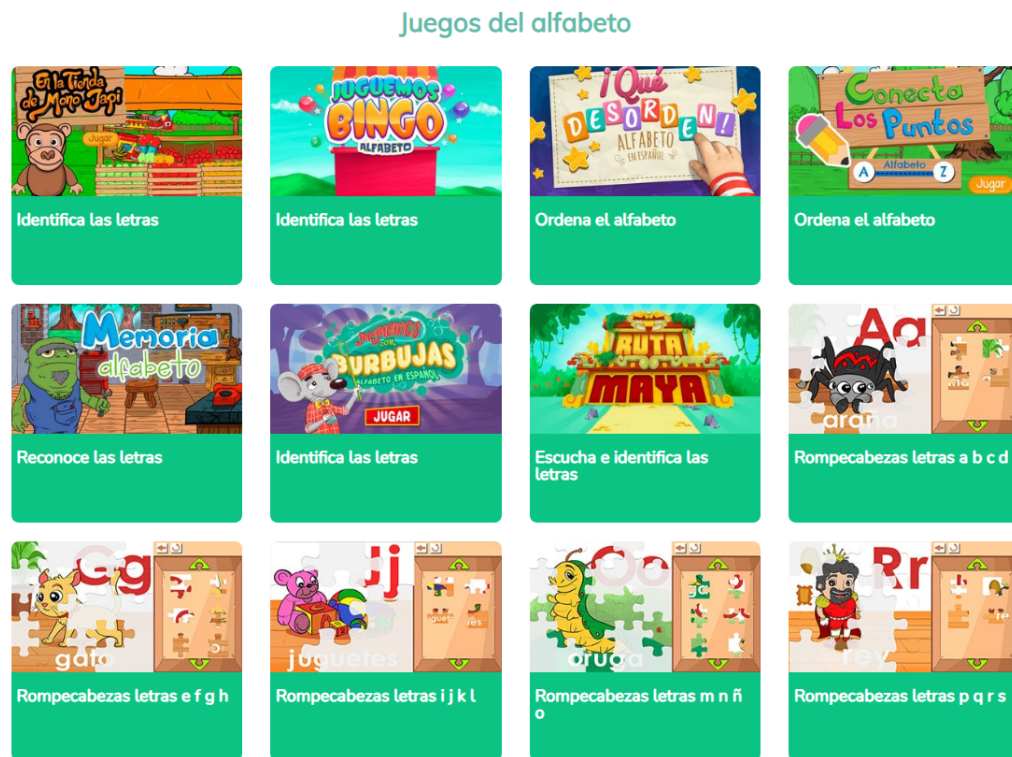


Figura A.6: Juegos ABC.com.

A.4. Primeras Palabras Sago Mini.



Figura A.7: Juego de hablar



Figura A.8: Juego de Seleccionar.

A.5. Abecedario, Aprende a Leer



Figura A.9: Ejemplo de juego de selección.



Figura A.10: Ejemplo de juegos

A.6. LingoKids - Aprender Jugando.

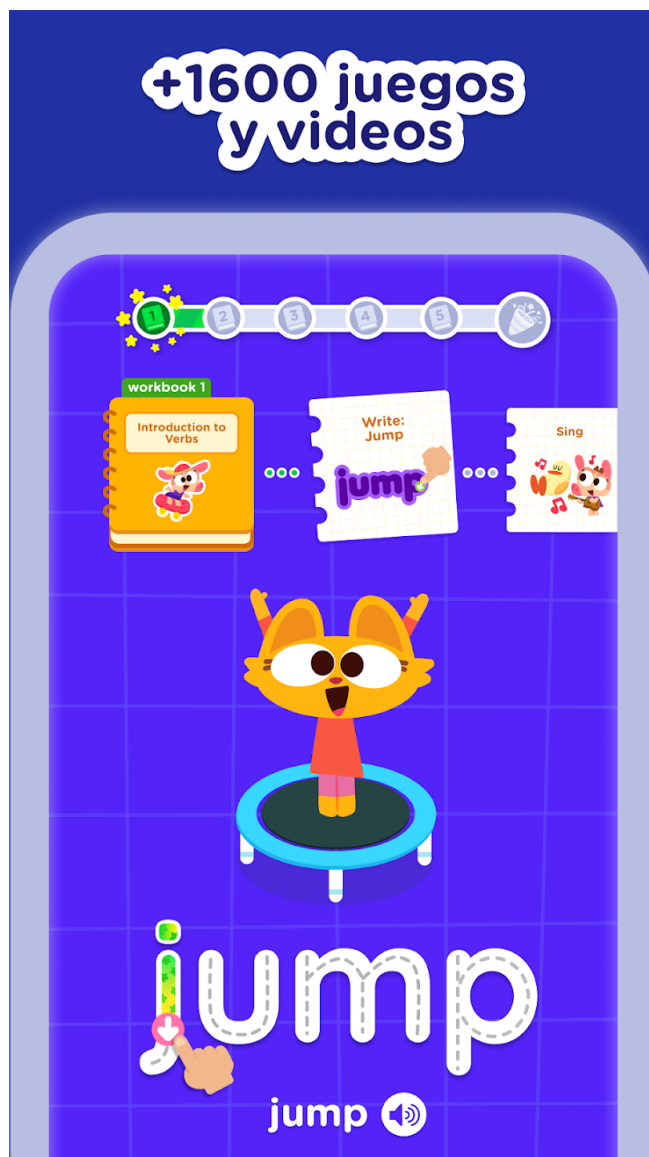


Figura A.11: Ejemplo juego caligrafía.

A.7. Learn to Read Duolingo

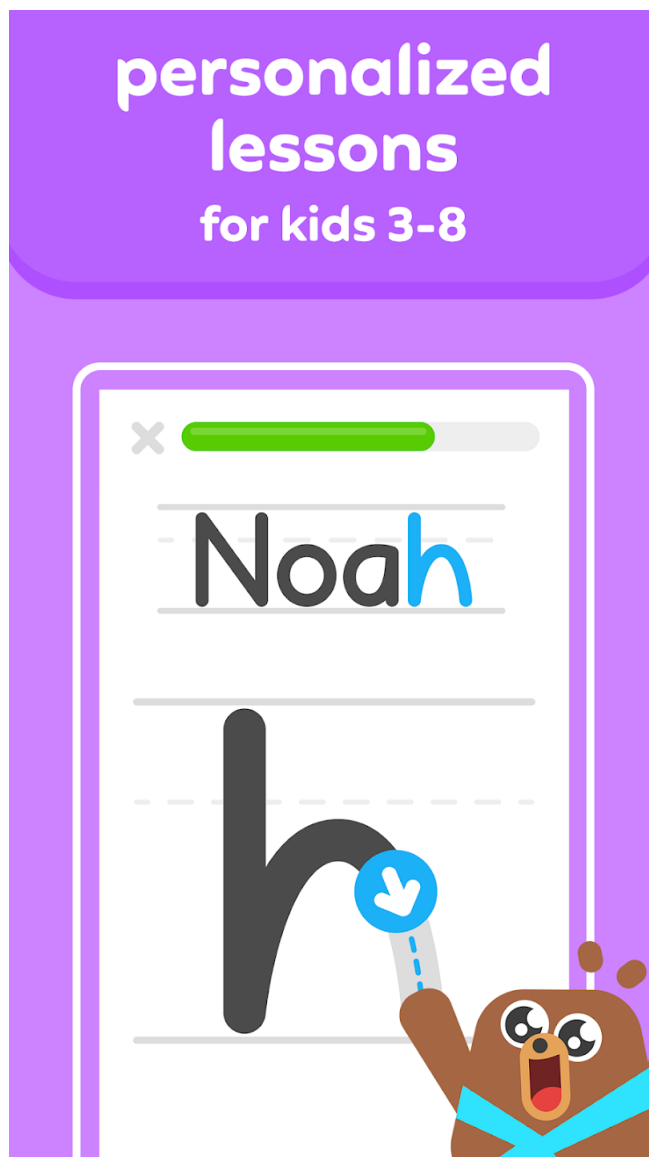


Figura A.12: Ejemplo juego caligrafía.

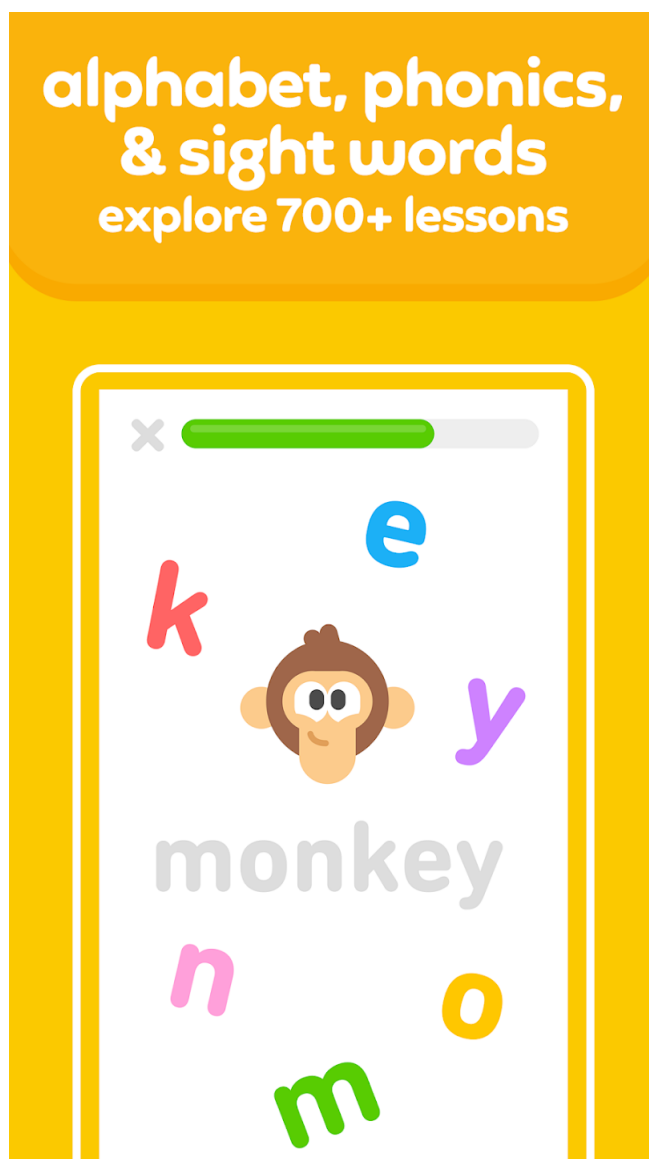


Figura A.13: Juego armado palabra.

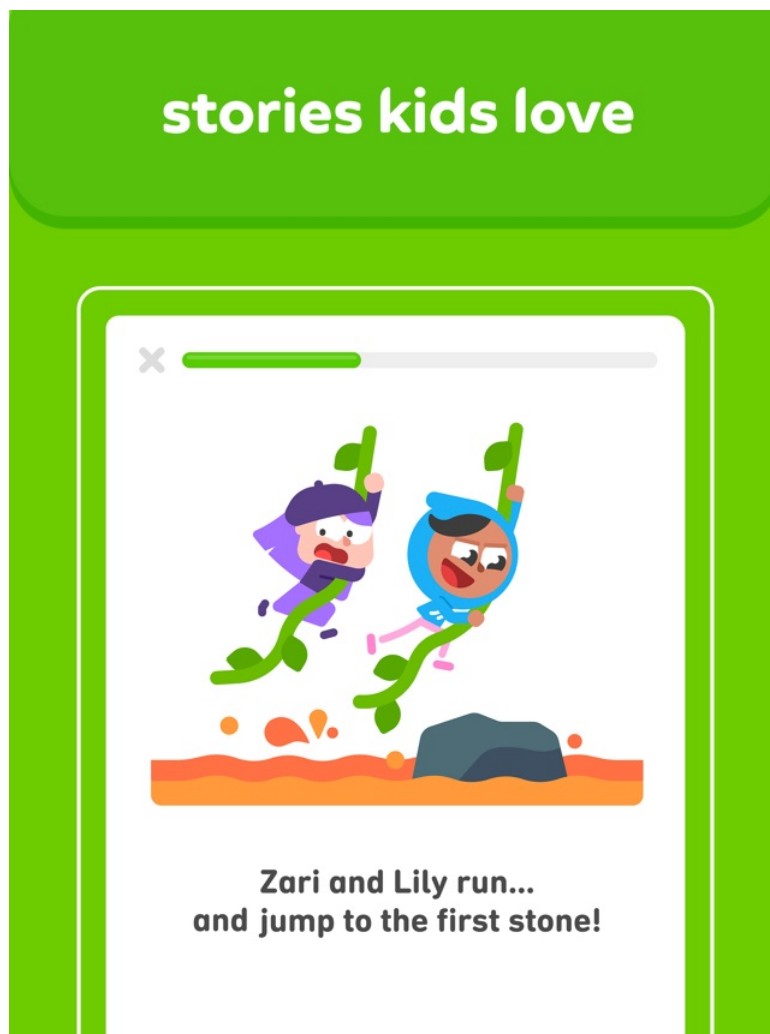


Figura A.14: Estructura de cuento que guía el aprendizaje.

B. Carta Gatt

B.2. Carta Gantt ciclo 3

Tarea	Semana 10	Semana 11	Semana 12	Semana 13	Semana 14	Semana 15	Semana 16
Creación Selección de ejercicio	X						
Busqueda de palabras bislabos y trislabos		X		X			
Expansión sistema de guardado y cargado a palabras				X	X		
Creación primer ejercicio selección de silabas sin TTS					X	X	
Implementación sistema de experiencia, consumo energia y entrega de items					X		
Creación canvas de los ejercicios con sistema STT, sin funcionalidad							X
Test de flujo y funcionamiento mascota, Question Manager y GameManager							X

B.3. Carta Gantt ciclo 4

Tarea	Semana 17	Semana 18	Semana 19
Creación líneas de diálogos mascota y historia de contextualización	X		
Busqueda de servicios STT y TTS para su implementación	X		
Primera iteración STT con HuggingFace		X	
Test HuggingFace		X	X

B.4. Carta Gantt ciclo 5

Tarea	Semana 20	Semana 21	Semana 22	Semana 23	Semana 24	Semana 25	Semana 26
Eliminación de sistema de comida y cambio por sistema de buffs en base al ítem de alimentación	X						
Implementación final creación de mascota con nombres de mascota y usuarios	X	X					
Creación galería de mascotas			X				
Primera implementación AWS y eliminación de HuggingFace			X	X	X		
Implementación completa de AWS				X	X		
Implementación sistema de puntaje STT para ajustarse al tiempo de respuesta de AWS					X	X	
Test Final de la aplicación							X

B.5. Carta Gantt ciclos 1(semanas 1 a 3) y ciclos 2 (semanas 4 a 9)

Tarea	Semana 1	Semana 2	Semana 3	Semana 4	Semana 5	Semana 6	Semana 7	Semana 8	Semana 9
Creación documento de referencias artísticas	X	X	X						
Primera iteración arte mascotas				X	X				
Segunda iteración arte mascota y background en Ilustrator							X	X	X
Creación arte habitad mascota								X	X
Creación iconos				X	X	X	X	X	X

B.6. Carta Gantt ciclos 3

Tarea	Semana 10	Semana 11	Semana 12	Semana 13	Semana 14	Semana 15	Semana 16
Tercera iteración arte completo en Leonardo.AI		X	X				
Ajustes arte generado a la paleta de colores establecida			X	X			
Incorporar nuevo arte a la aplicación				X			
Planteamiento estilo de música.	X						
Generación música en Sudo.ai		X	X	X	X		
Incorporación música generada en el software.					X	X	

C. Diseño de software

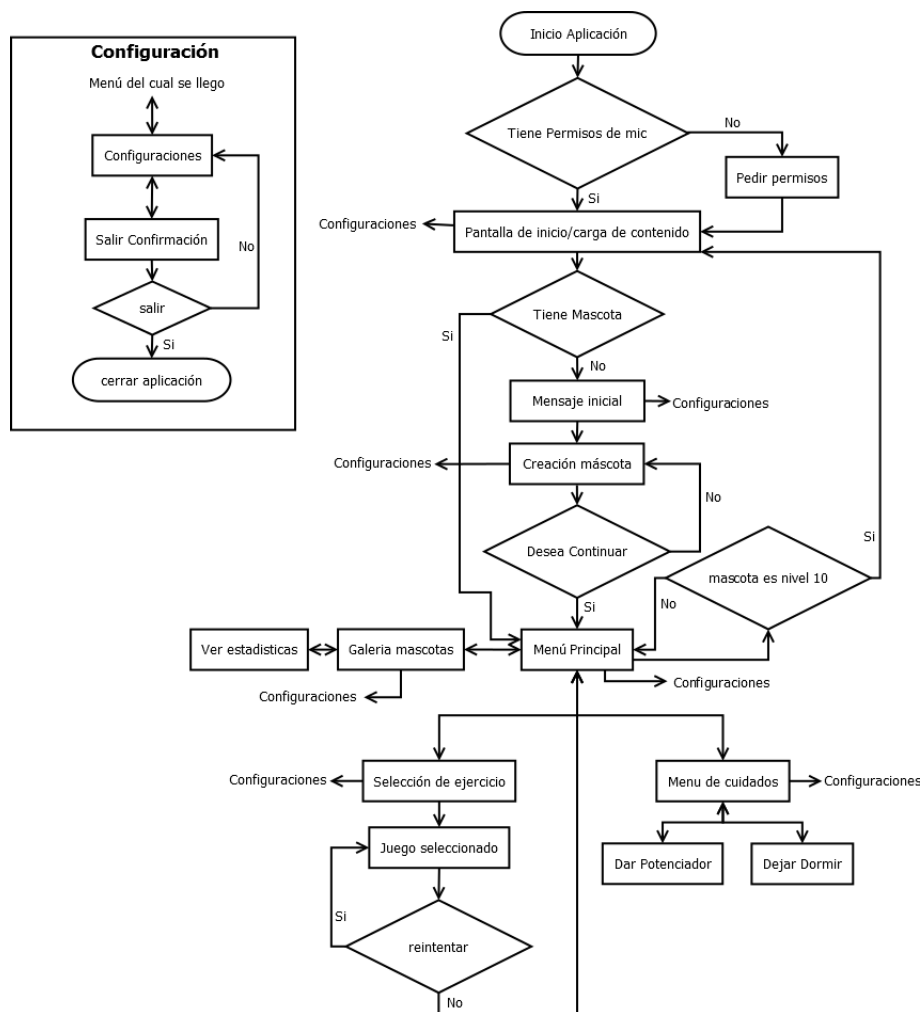


Figura C.1: Diagrama de flujo de las distintas interfaces del software



Figura C.2: Wireframe selección de mascota



Figura C.3: Wireframe Menú Principal



Figura C.4: Wireframe menu de Ejercicios



Figura C.5: Wireframe de los 3 tipos de ejercicios



Figura C.6: Wireframe cierre de ejercicio

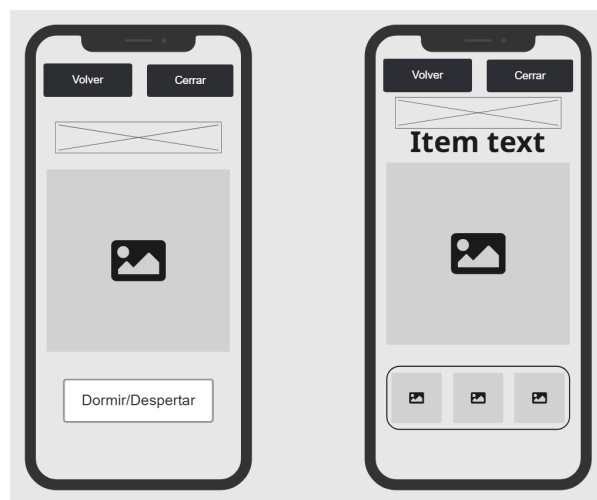


Figura C.7: Wireframe ventana de Cuidados



Figura C.8: Wireframe ventana de galería

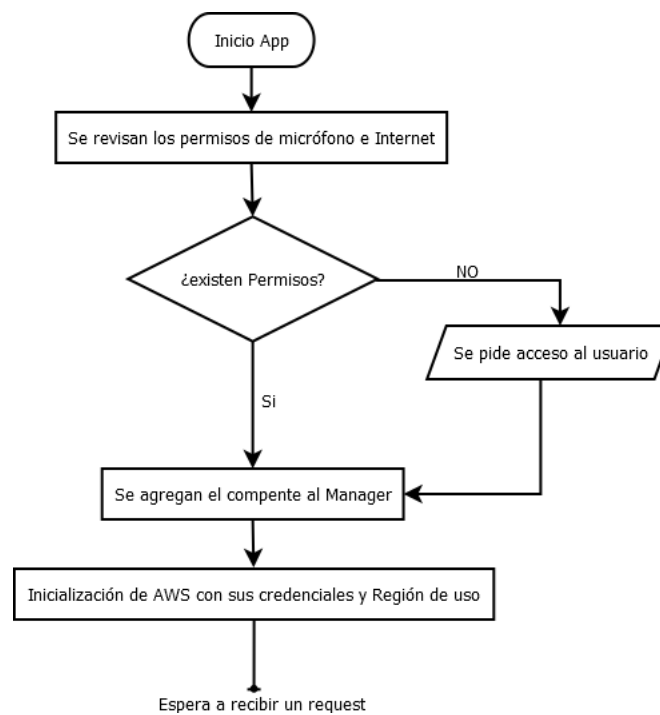


Figura C.9: Diagrama de flujo iniciación AWS

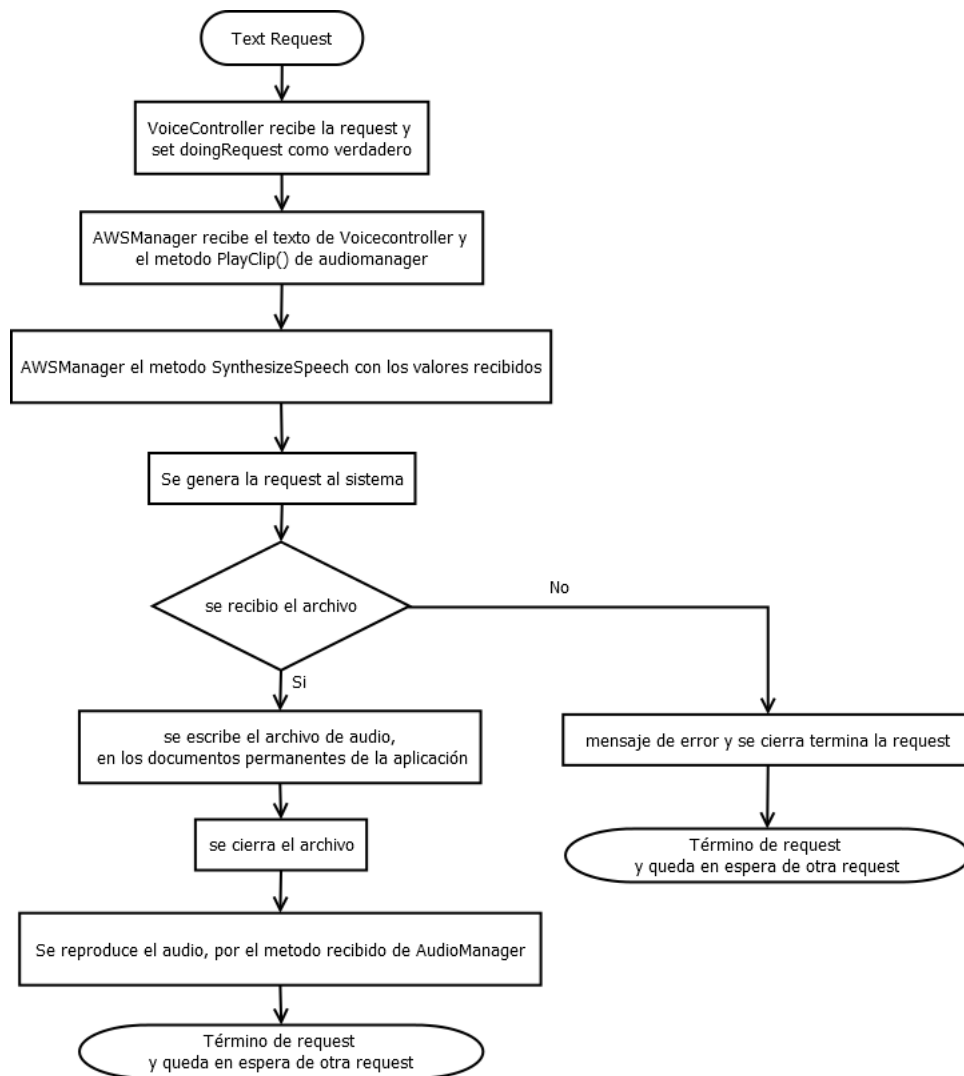


Figura C.10: Diagrama de flujo Polly Request

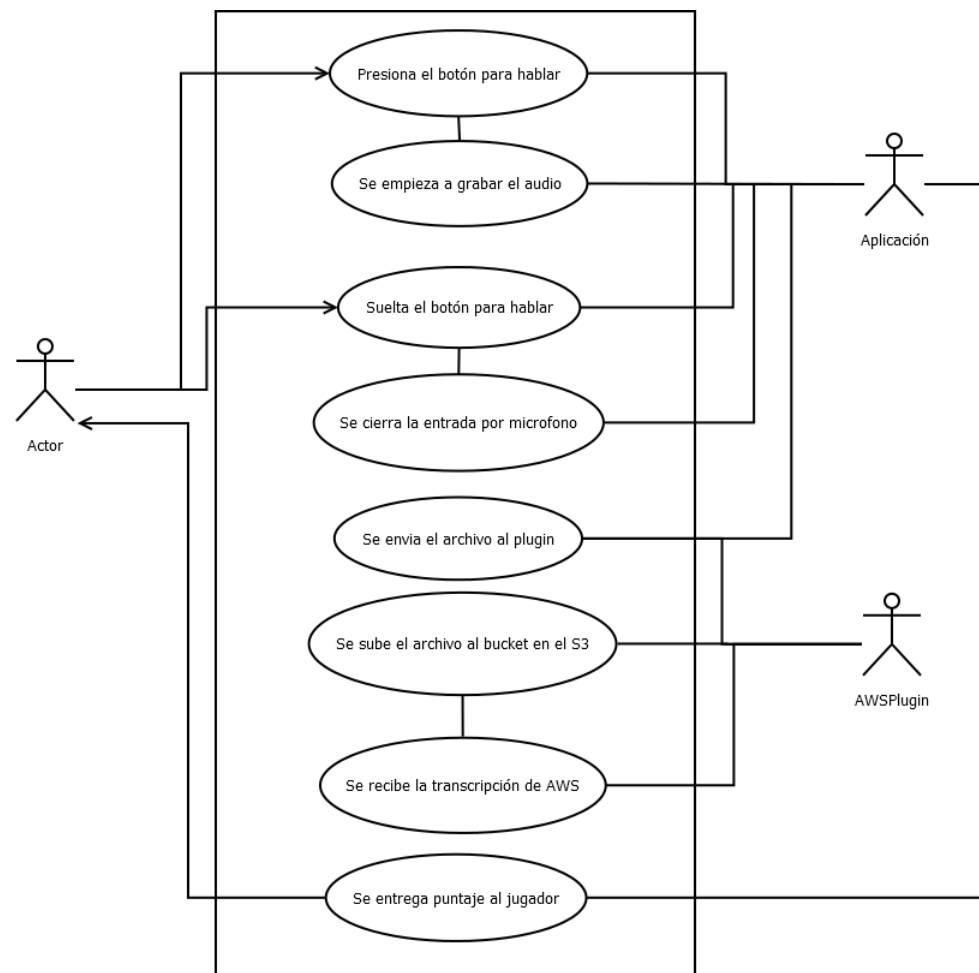


Figura C.11: Diagrama Casos transcribe Request

D. Anexo Código y App

A Continuación se presentan los distintos cuadros de códigos, referenciados en durante la sección de implementación, así mismo, con el link de drive con la apk del la aplicación la cual se encuentra funcional para android. https://drive.google.com/file/d/1cdg4Phw37Aecbxxpj_L_Ix7m3kgkmm_Z/view?usp=sharing

D.1. QuestionManager

Listing D.1: Cargado Preguntas en QuestionManager.

```
1 foreach (Questions q in list)
2 {
3     switch (q._enfoque)
4     {
5         case "Fonem_Grafem": Fonem_graf.Add(q); break;
6         case "Graf_Fonem": Graf_fonem.Add(q); break;
7         case "Fonem_Fonem": Fonem_fonem.Add(q); break;
8     }
9 }
```

Listing D.2: Instanciación del Canvas correspondiente a los ejercicios.

```
1 exerciseInstance = Instantiate(
2     questionsPref[sceneInfo._currentExercise]);
3
4 if (exerciseInstance.TryGetComponent<ExercisePrefaController>(
5     out ExercisePrefaController prefaController)){
6 }
```

```

7     prefController.id = sceneInfo._currentExercise;
8     prefController.Re = this.Re;
9     prefController.manager = this;
10  }

```

Listing D.3: Selección de pregunta a realizar.

```

1  List<Questions> list = questions.Where(
2      q => q._eje == gm.getActualMascot().ID).ToList();
3
4  UnityEngine.Random.InitState((int)DateTime.Now.Ticks);
5  return list[UnityEngine.Random.Range(0, list.Count)];

```

Listing D.4: Selección de respuestas incorrectas.

```

1  var fakeAnswers = questions.Where(
2      q => q.CorrectAnswer != selectedQuestion.CorrectAnswer)
3      .Select(q => q.CorrectAnswer)
4      .Distinct()
5      .Take(2)
6      .ToArray();
7  return fakeAnswers.Length == 2 ? fakeAnswers : new string[2];

```

Listing D.5: Proceso cierre pregunta.

```

1  ExerciseSelection();
2  Questioncounter++;
3
4  if (answer){
5      CorrectCounter++;
6      Maxtime += 5;
7      totaltime += 5;
8      AudioManager.instance.stopPlayingAudio();
9  }
10 else{
11     Maxtime -= 5;
12     totaltime -= 5;
13 }

```

```
14 | prefaController.updateExercise();
```

Listing D.6: Proceso Cierre ejercicio por QuestionManager.

```

1 Destroy(exerciseInstance);
2 intraining = false;
3
4 Mascot m = gm.getActualMascot();
5 m.training(sceneInfo._currentExercise,
6           CorrectCounter,
7           Questioncounter,
8           totaltime);
9
10 if (m.isGroundedMascot) {
11     gm.MoveSceneEndLvl();
12 }
13 else{
14     exerciseInstance = Instantiate(questionsPref[3]);
15     if (exerciseInstance.TryGetComponent<ExercisePrefaController>
16         out ExercisePrefaController prefaController))
17     {
18         prefaController.manager = this;
19         prefaController.id = 3;
20         prefaController.Re = this.Re;
21     }
22 }
```

Listing D.7: Proceso Cierre ejercicio por Mascot (Energia y Experiencia).

```

1 public void training(int id,
2                     float CorrectCount,
3                     float totalExercise,
4                     float totaltime)
5 {
6     /* Actualizar energia y experiencia */
7     int energyCost = itemsActives[0] ? 5 : 10;
8     _stats[0] -= energyCost;
9     TotalEnergyUsed += energyCost;
```

```

10     if (CorrectCount > 0)
11     {
12         _stats[2] = (itemsActives[1] ? 2 : 1)
13             * CorrectCount * 2 + totaltime * 0.6f;
14     }

```

Listing D.8: Proceso Cierre ejercicio por Mascot (Item).

```

1     int multiplier = itemsActives[2] ? 2 : 1;
2     int minRange = GetMinRange(winrate, CorrectCount);
3     int maxRange = GetMaxRange(winrate, CorrectCount);
4
5     itemsAdded = multiplier *
6         UnityEngine.Random.Range(minRange, maxRange);
7     _inventory[id] = itemsAdded;

```

Listing D.9: Proceso Cierre ejercicio por Mascot (subir nivel y apagado items).

```

1     _levelUp();
2     for (int i = 0; i < itemsActives.Length; i++){
3         if (i != 1 || (i == 1 && itemsAdded > 0)){
4             itemsActives[i] = false;
5         }
6     }
7     ObtainedItemsRecap[id] += itemsAdded;

```

Listing D.10: Cálculo cap para subir de nivel

```

1 int level = (int)_stats[3] + 1;
2 _maxExp = 4 * (int)Math.Pow(level, 3) / 5;
3 return _maxExp;

```

Listing D.11: Límites para entrega de items basado en el winrate del ejercicio

```

1 /* Rango Para la entrega de items */
2 private int GetMinRange(float winrate, float CorrectCount){
3     int baseMinRange = winrate > 0.6f ? 4 : winrate < 0.3f ? 1:2;
4     return CorrectCount > 10 ?
5         (int)(CorrectCount / 10) * baseMinRange : baseMinRange;

```

```

6 }
7 private int GetMaxRange(float winrate, float CorrectCount){
8     int baseMaxRange = winrate > 0.6f ? 6 : winrate < 0.3f ? 2:4;
9     return CorrectCount > 10 ?
10         (int)(CorrectCount / 10) * baseMaxRange : baseMaxRange;
11 }

```

D.2. ExercicePrefa

Listing D.12: Inicialización juego FonemGraf con sus botones

```

1 UnityEngine.Random.InitState((int)DateTime.Now.Ticks);
2
3 int selectedButton = UnityEngine.Random.Range(0, 3);
4 buttonList[selectedButton].tag = TAG_CORRECT;
5 buttontexts[selectedButton].text =
6     manager.selectedQuestion.CorrectAnswer;
7
8 var failAnswers = manager.failAnswer;
9 int failIndex = 0;
10
11 for (int i = 0; i < buttonList.Count; i++){
12     if (i != selectedButton){
13         buttonList[i].tag = TAG_FAIL;
14         buttontexts[i].text = failAnswers[failIndex++];
15     }
16 }

```

Listing D.13: Comprobación respuesta

```

1 manager.CheckAnswer(b.CompareTag(TAG_CORRECT));

```

Listing D.14: onpress

```

1 if (!isRecording) {
2     isRecording = true;
3     vc.StartListening();
4 }

```

Listing D.15: onrelease

```

1  if (isRecording){
2      isRecording = false;
3      vc.StopListening();
4      SpeechAnswer();
5  }

```

Listing D.16: answercheck

```

1  manager.CheckAnswer(vc.textString.Equals(
2      manager.selectedQuestion.CorrectAnswer,
3      StringComparison.OrdinalIgnoreCase));

```

D.3. SaveSystem

Listing D.17: Sistema de guardado de mascotas

```

1  if (mascots == null){
2      Debug.LogError("Mascots list is null.");
3      return;
4  }
5  MCountData.Count = mascots.Count;
6  string countDataJson = JsonUtility.ToJson(MCountData);
7  File.WriteAllText(GetFilePath("/Mascot.Count"), countDataJson);
8
9  for (int i = 0; i < mascots.Count; i++){
10     MascotData data = new(mascots[i]);
11     string json = JsonUtility.ToJson(data);
12     File.WriteAllText(GetFilePath($"MascotData{i}"), json);
13 }

```

Listing D.18: Sistema de cargado de mascotas

```

1  public List<Mascot> LoadMascot(){
2
3  List<Mascot> mascots = new List<Mascot>();
4  string countPath = GetFilePath("/Mascot.Count");

```

```

5  if (File.Exists(countPath)){
6      string countDataJson = File.ReadAllText(countPath);
7      MCountData = JsonUtility.FromJson<Counter>(countDataJson);
8  }
9  for (int i = 0; i < MCountData.Count; i++){
10     string dataPath = GetFilePath($"MascotData{i}");
11     if (File.Exists(dataPath)){
12         string jsonText = File.ReadAllText(dataPath);
13         MascotData m = JsonUtility.FromJson<MascotData>(jsonText);
14         mascots.Add(m.loadMascot());
15     }
16     else{
17         Debug.LogError("Error loading Mascots.");
18     }
19 }
20 return mascots;
21 }

```

Listing D.19: Sistema de cargado de preguntas

```

1  public List<Questions> LoadQuestions(int index){
2      List<Questions> questionList = new List<Questions>();
3      if (index < 0 || index >= QuestionTxt.Count){
4          Debug.LogError("Indice de preguntas fuera de rango.");
5          return questionList;
6      }
7      TextAsset questionData = QuestionTxt[index];
8      if (questionData != null){
9          QuestionsData q = JsonUtility.FromJson<QuestionsData>
10                                     (questionData.text);
11          for (int i = 0; i < q._ejes.Length; i++){
12              questionList.Add(q.getvalue(i));
13          }
14      }
15      else{
16          Debug.LogError(
17              "No se pudo cargar el recurso de preguntas: " + index);

```

```
18     }
19     return questionList;
20 }
```

D.4. AudioManager

Listing D.20: TTSPay

```
1  voiceSource.clip = clip;
2  voiceSource.Play();
3  StartCoroutine(checkIfAudioFinished());
4
5  while (voiceSource.isPlaying)
6  {
7      yield return null;
8  }
9  OnAudioFinished?.Invoke();
```

Listing D.21: AudioChecker

```
1  while (voiceSource.isPlaying)
2  {
3      yield return null;
4  }
5  OnAudioFinished?.Invoke();
```

Listing D.22: Start y Stop Rercording microfono

```
1  if (Microphone.devices.Length > 0){
2      microphoneDevice = Microphone.devices[0];
3      recordedClip = Microphone.Start(microphoneDevice,
4      false,
5      60,
6      44100);
7  }
8  else{
9      Debug.LogError("No microphone devices found.");
```

```
10 }  
11  
12 if (Microphone.IsRecording(microphoneDevice)){  
13     Microphone.End(microphoneDevice);  
14 }
```

D.5. VoiceController

Listing D.23: Start y Stop Rercording microfono

```
1 if (Microphone.devices.Length > 0){  
2     microphoneDevice = Microphone.devices[0];  
3     recordedClip = Microphone.Start(microphoneDevice ,  
4     false ,  
5     60 ,  
6     44100);  
7 }  
8 else{  
9     Debug.LogError("No microphone devices found.");  
10 }  
11  
12 if (Microphone.IsRecording(microphoneDevice)){  
13     Microphone.End(microphoneDevice);  
14 }
```

D.6. VoiceController

Listing D.24: Inicio request a AWSManager para usar Polly.

```
1 public void startSpeaking(string message){  
2     if (!doingRequest)  
3     {  
4         doingRequest = true;  
5         awsManager.StartSpeaking(  
6             message ,  
7             AudioManager.instance.PlayAudioClip);
```

```

8      doingRequest = false;
9      OnRequestFinished?.Invoke();
10     }
11 }

```

Listing D.25: Inicio y término de proceso de request Transcribe

```

1 public void StartListening(){
2     AudioManager.instance.StartRecording();
3     talking = true;
4 }
5
6 public void StopListening(){
7     AudioManager.instance.StopRecording();
8     awsManager.StopRecordingAndUpload(
9         AudioManager.instance.GetRecordedClip(),
10        OnFinalSpeechResult);
11     talking = false;
12 }

```

D.7. Amazon Web Service AWS

D.7.1. AWSManager

Listing D.26: Inicialización de AWSManager e inicio web service.

```

1  AWSPolly = gameObject.AddComponent<AWSPollyTTS>();
2  AWSTranscribe = gameObject.AddComponent<AWSTranscribeSTT>();
3  AWSS3 = gameObject.AddComponent<AWSS3>();
4  networkChecker = gameObject.AddComponent<NetworkChecker>();
5
6  var credentials = new BasicAWSCredentials(AccesKey, SecretKey);
7  AWSPolly.Initialize(credentials, RegionTTS);
8  AWSTranscribe.Initialize(credentials, Region);
9  AWSS3.Initialize(credentials, Region);

```

Listing D.27: Comprabación conexión internet por Newtwokchecker.

```
1 private IEnumerator CheckInternetConnectionCoroutine(Action<bool>
2                                     callback){
3     using (UnityWebRequest www = UnityWebRequest.Get(
4         "https://www.google.com")){
5
6         yield return www.SendWebRequest();
7         if (www.result == UnityWebRequest.Result.ConnectionError
8             || www.result == UnityWebRequest.Result.ProtocolError){
9
10            Debug.LogError("No internet connection: " + www.error);
11            callback(false);
12        }
13        else{
14            callback(true);
15        }
16    }
17 }
```

Listing D.28: Conversión Audioclip a Formato WAV.

```
1 private static byte[] ConvertToWav(
2     float[] samples,
3     int channels,
4     int frequency){
5
6     MemoryStream stream = new MemoryStream();
7     BinaryWriter writer = new BinaryWriter(stream);
8
9     writer.Write("RIFF".ToCharArray());
10    writer.Write(36 + samples.Length * 2);
11    writer.Write("WAVE".ToCharArray());
12    writer.Write("fmt ".ToCharArray());
13    writer.Write(16);
14    writer.Write((short)1);
15    writer.Write((short)channels);
16    writer.Write(frequency);
17    writer.Write(frequency * channels * 2);
```

```

18     writer.Write((short)(channels * 2));
19     writer.Write((short)16);
20     writer.Write("data".ToCharArray());
21     writer.Write(samples.Length * 2);
22
23     foreach (var sample in samples){
24         writer.Write((short)(sample * short.MaxValue));
25     }
26     writer.Flush();
27     return stream.ToArray();
28 }

```

D.7.2. AWSPolly

Listing D.29: Preparación de la request por parte de Polly

```

1 var request = new SynthesizeSpeechRequest {
2     Text = text,
3     VoiceId = VoiceId.Mia,
4     Engine = Engine.Neural,
5     OutputFormat = OutputFormat.Mp3,
6 };

```

Listing D.30: Envío Request y procesamiento audio.

```

1 var responseTask = await
2     pollyClient.SynthesizeSpeechAsync(request);
3
4 WriteIntoFile(responseTask.AudioStream);
5 using (var www =
6     UnityWebRequestMultimedia.GetAudioClip($
7         "{Application.persistentDataPath}/audio.mp3",
8         AudioType.MPEG)){
9
10     var op = www.SendWebRequest();
11     while (!op.isDone) await Task.Yield();
12

```

```

13     if (www.result == UnityWebRequest.Result.ConnectionError ||
14         www.result == UnityWebRequest.Result.ProtocolError){
15         Debug.LogError(www.error);
16     }
17     else{
18         var clip = DownloadHandlerAudioClip.GetContent(www);
19         callback?.Invoke(clip);
20     }
21 }

```

Listing D.31: Escritura archivo MP3

```

1  using (FileStream fileStream = new FileStream(
2      path: $"{Application.persistentDataPath}/audio.mp3",
3      FileMode.Create)){
4
5      byte[] buffer = new byte[8 * 1024];
6      int bytesRead;
7      while((bytesRead = stream.Read(buffer, 0, buffer.Length))>0)
8      {
9          fileStream.Write(buffer, 0, bytesRead);
10     }
11 }
12 };

```

D.7.3. AWSTranscribe

Listing D.32: Preparación request de transcripción.

```

1  var request = new StartTranscriptionJobRequest
2  {
3      TranscriptionJobName = jobName,
4      LanguageCode = LanguageCode.EsUS,
5      Media = new Media { MediaFileUri = s3Uri },
6
7      OutputBucketName = bucketName
8  };

```

Listing D.33: Envío Request y trabajo de recepción.

```

1 var response = await
2     transcribeClient.StartTranscriptionJobAsync(request);
3
4 await PollTranscriptionJobAsync(jobName, callback);
5
6 var request = new GetTranscriptionJobRequest{
7     TranscriptionJobName = jobName
8 };
9
10 while (true){
11     var getResponse = await
12         transcribeClient.GetTranscriptionJobAsync(request);
13
14     var job = getResponse.TranscriptionJob;
15
16     if (job.TranscriptionJobStatus == TranscriptionJobStatus.COMPLETED){
17         string transcript = await
18             GetTranscriptionJobResultAsync(
19                 job.
20                 Transcript.
21                 TranscriptFileUri);
22         callback?.Invoke(transcript);
23         break;
24     }
25     else if (job.TranscriptionJobStatus == TranscriptionJobStatus.FAILED){
26         Debug.LogError("Transcription job failed.");
27         break;
28     }
29     await Task.Delay(5000);
30 }

```

Listing D.34: Extracción string del JSON recibido por parte de la nube.

```

1 private async Task<string> GetTranscriptionJobResultAsync(
2     string transcriptionFileUri){
3

```

```

4      using (var webClient = new System.Net.WebClient())
5      {
6          string transcriptionJson = await
7              webClient.DownloadStringTaskAsync(transcriptionFileUri);
8
9          var transcriptionObject =
10             JsonUtility.FromJson<TranscriptionResult>(transcriptionJson);
11
12             if (transcriptionObject != null &&
13                 transcriptionObject.results != null &&
14                 transcriptionObject.results.transcripts != null &&
15                 transcriptionObject.results.transcripts.Count > 0){
16
17                 string transcript = transcriptionObject.
18                                     results.
19                                     transcripts[0].
20                                     transcript;
21
22                 return transcript;
23             }
24             else{
25                 Debug.LogError("Error parsing transcription result.");
26                 return null;
27             }
28     }

```

D.7.4. AWSS3

Listing D.35: proceso cargado de archivos a la nube por el AWSS3.

```

1 public async Task UploadFileAsync(
2     string filePath,
3     string key,
4     string bucketName){
5     try{
6         var putRequest = new PutObjectRequest{
7             BucketName = bucketName,

```

```
8         Key = key ,
9         FilePath = filePath ,
10        ContentType = "audio/wav"
11    };
12    var response = await s3Client.PutObjectAsync(putRequest);
13
14    if (response.HttpStatusCode == System.Net.HttpStatusCode.OK){
15        Debug.Log("File uploaded successfully to S3.");
16    }
17    else{
18        Debug.LogError("Failed to upload file to S3.");
19    }
20 }
21 catch (AmazonS3Exception e){
22     Debug.LogError("Error uploading file: " + e.Message);
23 }
24 };
```

E. Anexo Encuesta, Análisis y resultados

E.1. Encuesta

P.1	¿El juego te hizo querer practicar más?
P.2	¿El juego te dejó hacer lo que querías?
P.3	¿Qué tan difícil fue el juego de seleccionar sílabas?
P.4	¿Qué tan difícil fue el juego de leer y hablar?
P.5	¿Qué tan difícil fue el juego de escuchar y hablar?
P.6	¿Te sirvieron las instrucciones con audio?
P.7	¿Cómo te sentiste al terminar la sesión de entrenamiento?
P.8	¿Te interesa poder jugar con otras mascotas?
P.9	¿Te gustó la mascota con la que jugaste?
P.10	¿Te gustó la música de la mascota?
P.11	¿Te gustó la casa de la mascota?

Cuadro E.1: Preguntas encuestas

Género	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11
Niño	5	2	5	5	3	2	5	5	5	5	5
Niño	4	1	5	4	4	5	1	5	5	3	5
Niño	3	1	5	1	1	1	4	5	5	5	5
Niña	5	5	4	4	5	5	1	5	5	5	4
Niño	5	5	4	5	5	5	5	5	4	5	5
Niño	5	3	5	4	5	5	5	5	5	3	5
Niño	5	4	3	5	3	4	3	5	5	5	5
Niño	5	4	5	5	5	3	1	2	5	5	5
Niño	5	3	4	3	3	5	5	5	3	3	3
Niño	5	3	4	5	5	1	2	5	3	5	2
Niña	5	5	3	1	1	5	5	5	5	5	3
Niña	4	4	3	4	1	4	5	5	5	5	3
Niña	5	4	5	3	3	2	5	5	4	5	3
Niño	5	5	5	5	5	5	5	5	5	5	5
Niño	4	4	4	3	2	3	3	3	4	4	3
Niña	5	5	5	5	5	5	5	5	5	5	5
Niña	4	5	5	5	5	5	2	5	5	5	5
Niña	5	4	5	5	5	5	3	5	5	5	4
Niña	5	5	5	5	5	5	5	5	5	5	5
Niña	5	5	5	5	5	5	5	5	5	5	5
Niña	5	5	5	5	5	5	5	5	5	5	5
Niña	5	5	5	5	5	5	5	5	5	5	5
Niña	4	3	5	3	3	5	5	5	5	5	5

Cuadro E.2: Valores resultados encuesta

E.2. Gráficos

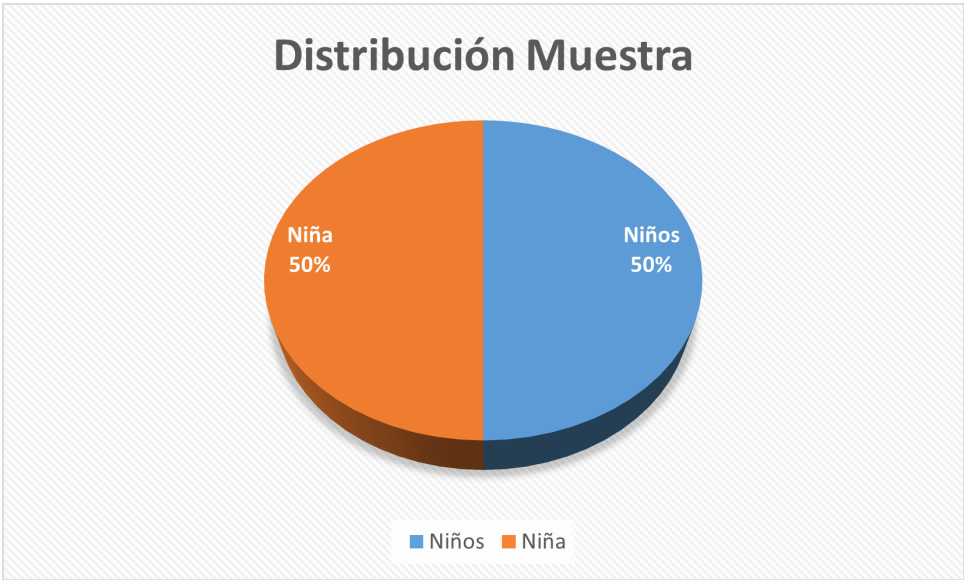


Figura E.1: Gráfico distribución demografía



Figura E.2: Gráfico Pregunta: ¿El juego te hizo querer practicar más?



Figura E.3: Gráfico Pregunta: ¿El juego te dejó hacer lo que querías?



Figura E.4: Gráfico Pregunta: ¿Qué tan difícil fue el juego de seleccionar sílabas?

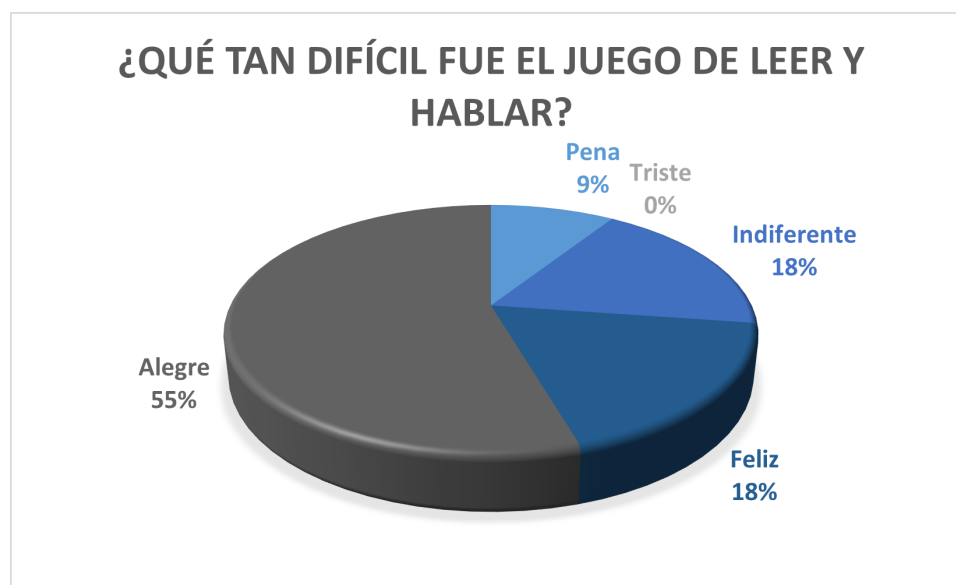


Figura E.5: Gráfico Pregunta: ¿Qué tan difícil fue el juego de leer y hablar?



Figura E.6: Gráfico Pregunta: ¿Qué tan difícil fue el juego de escuchar y hablar?

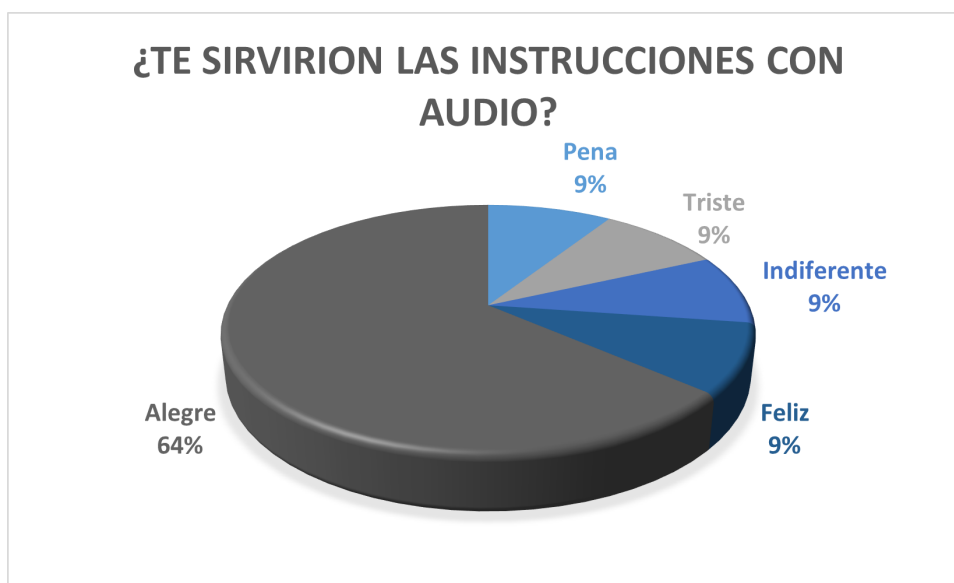


Figura E.7: Gráfico Pregunta: ¿Te sirvieron las instrucciones con audio?

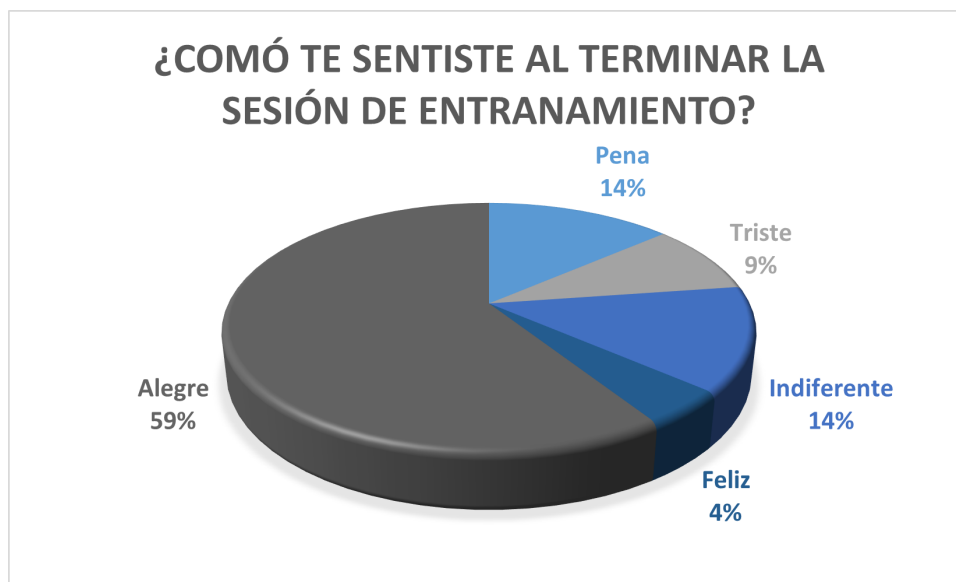


Figura E.8: Gráfico Pregunta: ¿Cómo te sentiste al terminar la sesión de entrenamiento?



Figura E.9: Gráfico Pregunta: ¿Te intereso poder jugar con otras mascotas?

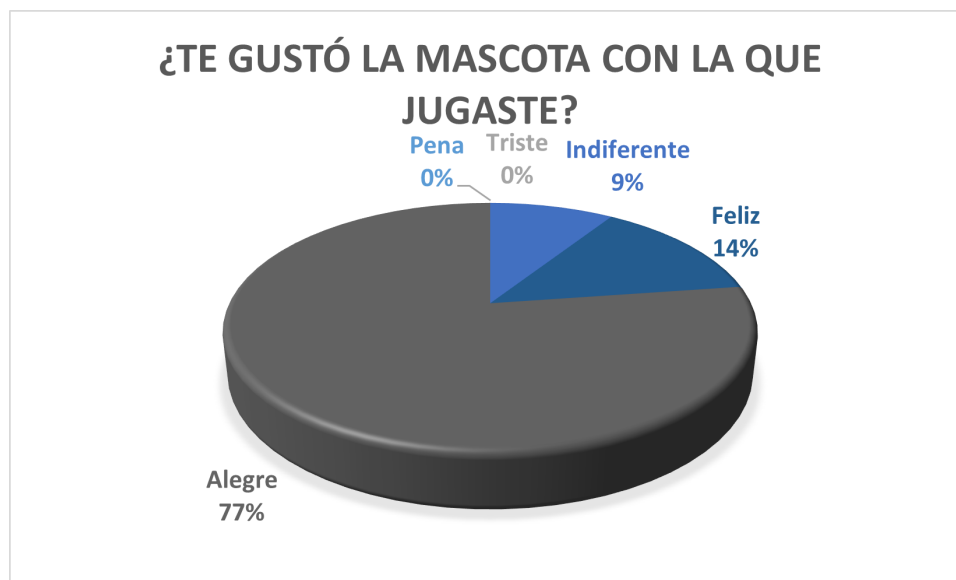


Figura E.10: Gráfico Pregunta: ¿Te gustó la mascota con la que jugaste?

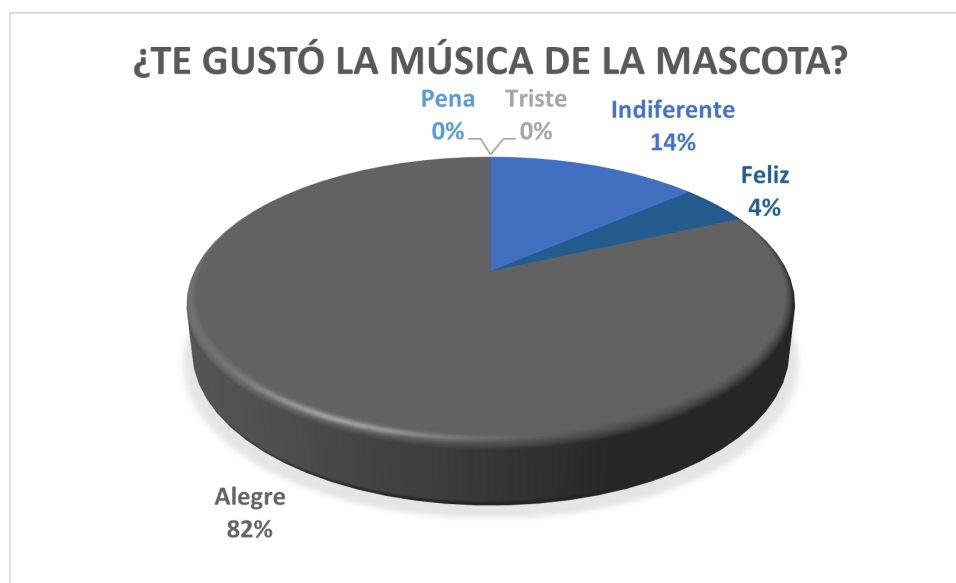


Figura E.11: Gráfico Pregunta: ¿Te gustó la música de la mascota?

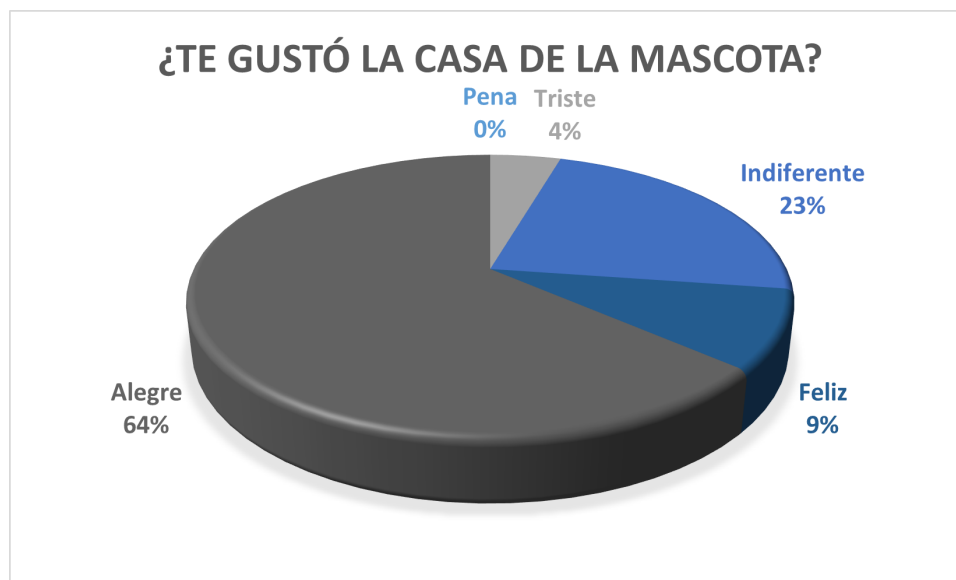


Figura E.12: Gráfico Pregunta: ¿Te gustó la casa de la mascota?



Figura E.13: Gráfico Pregunta: ¿El juego te hizo querer practicar más? por género



Figura E.14: Gráfico Pregunta: ¿El juego te dejó hacer lo que querías? por género

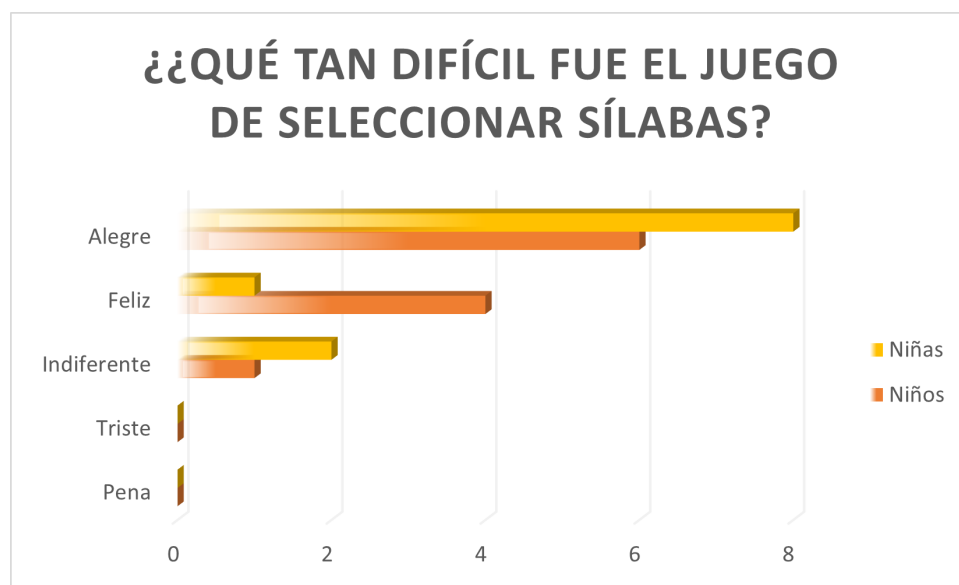


Figura E.15: Gráfico Pregunta: ¿Qué tan difícil fue el juego de seleccionar sílabas? por género

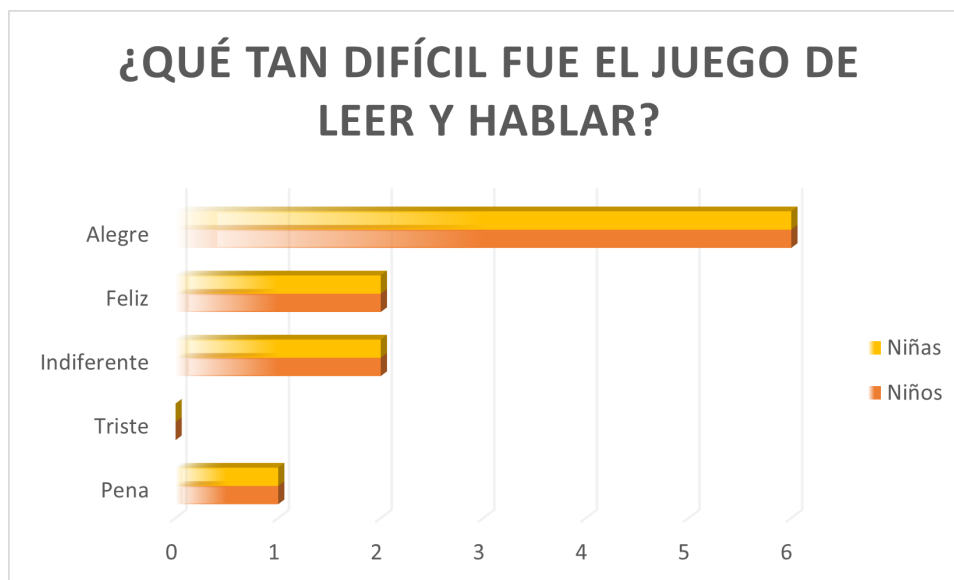


Figura E.16: Gráfico Pregunta: ¿Qué tan difícil fue el juego de leer y hablar? por género

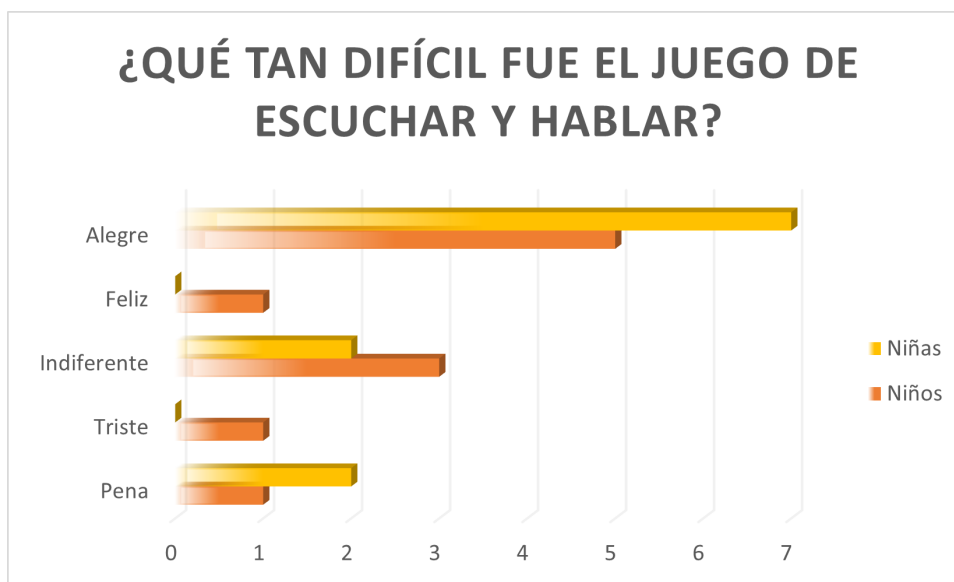


Figura E.17: Gráfico Pregunta: ¿Qué tan difícil fue el juego de escuchar y hablar? por género



Figura E.18: Gráfico Pregunta:¿Te sirvirion las instrucciones con audio? por género

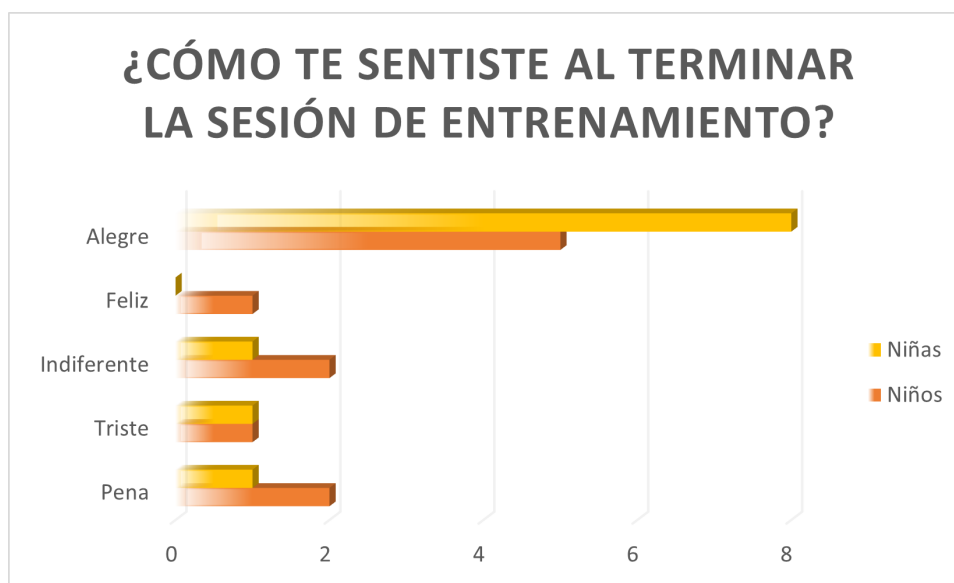


Figura E.19: Gráfico Pregunta: ¿Cómo te sentiste al terminar la sesión de entrenamiento? por género



Figura E.20: Gráfico Pregunta: ¿Te intereso poder jugar con otras mascotas? por género

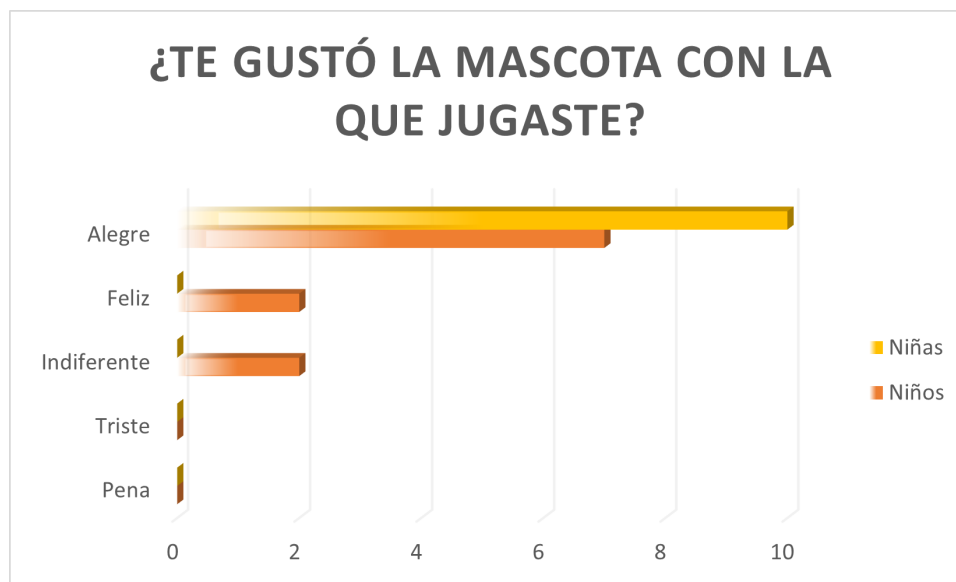


Figura E.21: Gráfico Pregunta: ¿Te gustó la mascota con la que jugaste? por género

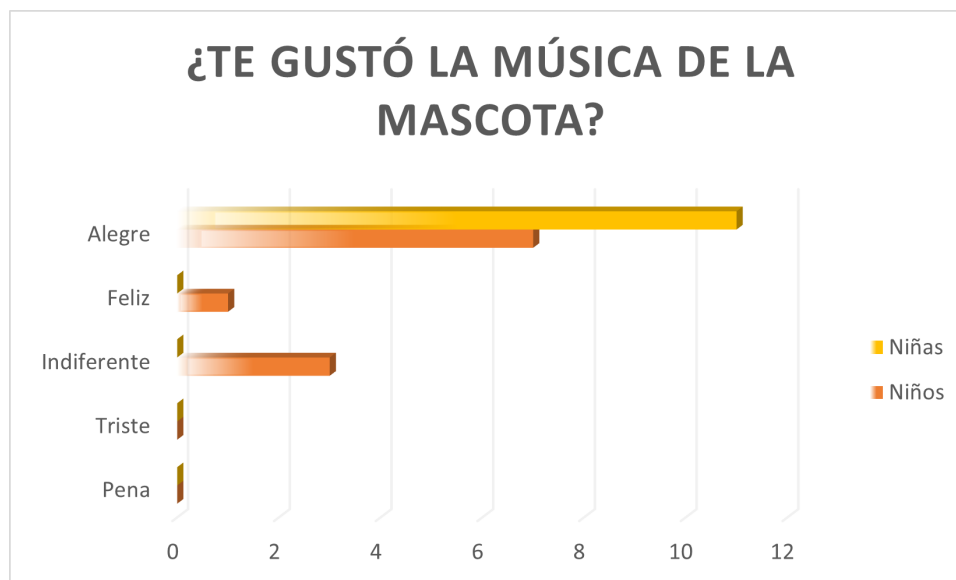


Figura E.22: Gráfico Pregunta: ¿Te gustó la música de la mascota? por género



Figura E.23: Gráfico Pregunta: ¿Te gustó la casa de la mascota? por género

E.3. Estudio de Arte

E.3.1. Referencia artística de series infantiles y juegos



Figura E.24: Referencia Blue.

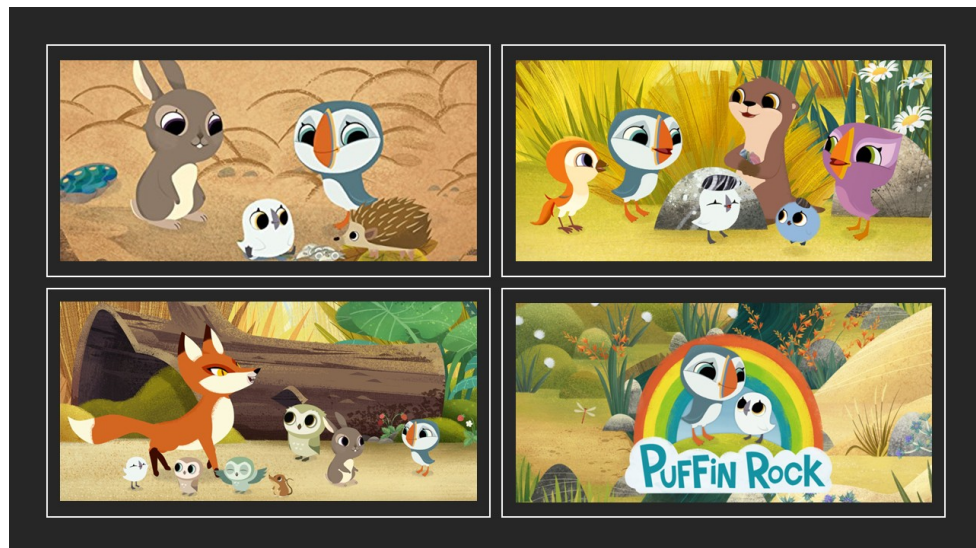


Figura E.25: Referencia Puffin Rock.

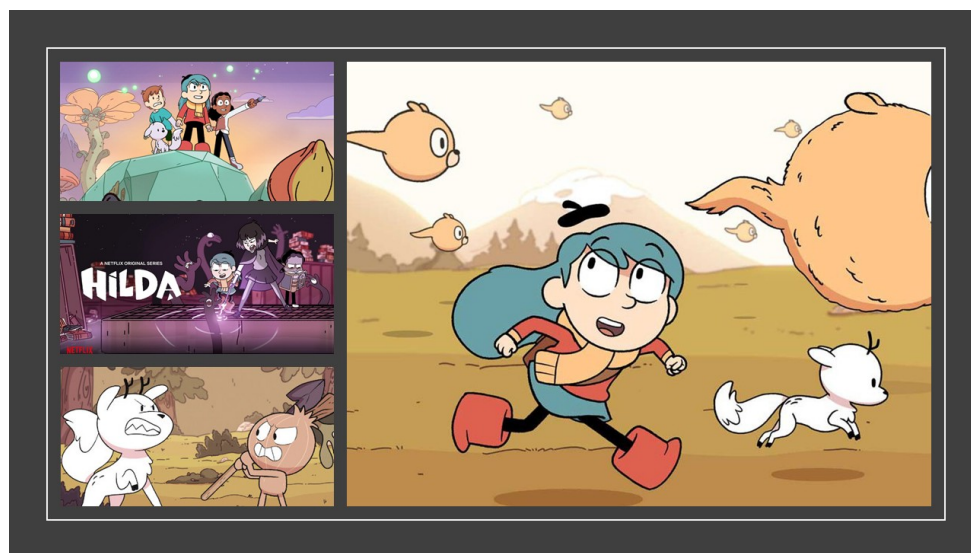


Figura E.26: Referencia Hilda.

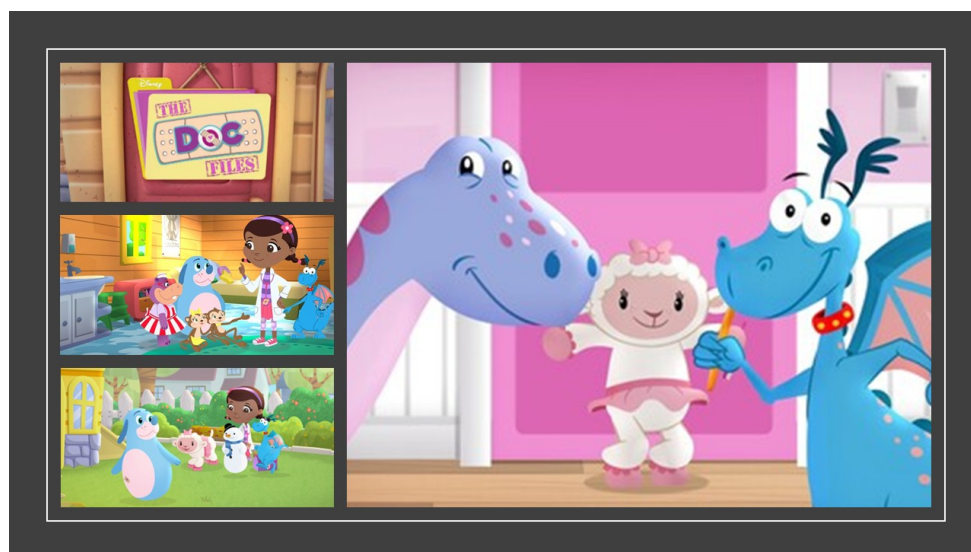


Figura E.27: Referencia Doctora juguete.

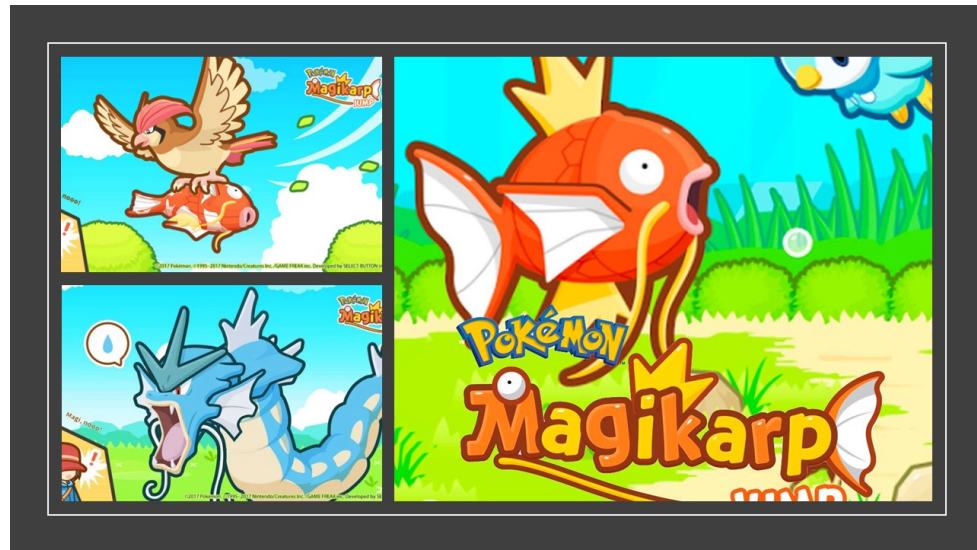


Figura E.28: Referencia Magikarp Jump.



Figura E.29: Referencia Docky y sus amigos.

E.3.2. Animales ocupados

Sapo



Figura E.30: Referencia sapos.



Figura E.31: Referencia sapos cartoon.

Carpintero



Figura E.32: Referencia carpinteros.

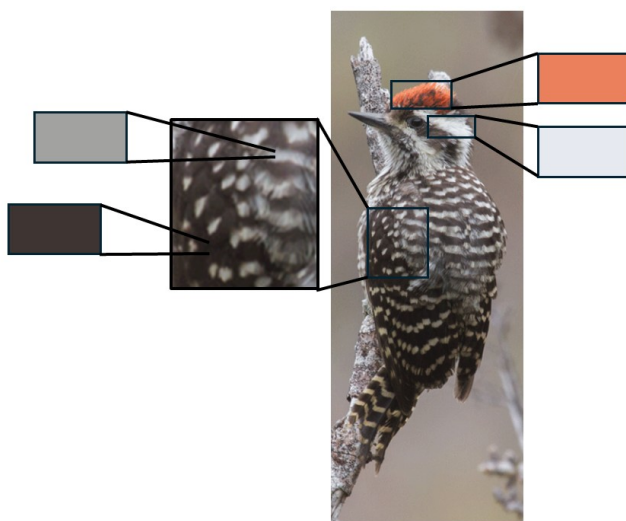


Figura E.33: Referencia estudio de color del carpintero.

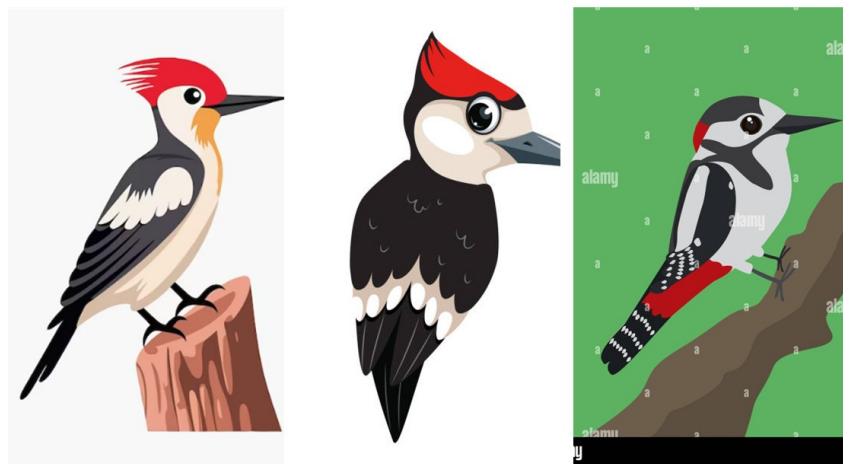


Figura E.34: Referencia carpintero cartoon.



Figura E.35: Referencia carpintero cartoon 2.

Zorro cumpeo.



Figura E.36: Referencia zorro cumpeo.

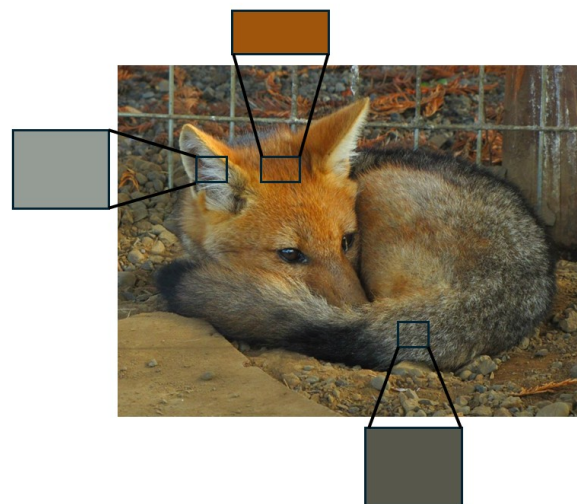


Figura E.37: Referencia estudio de color del zorro cumpeo.

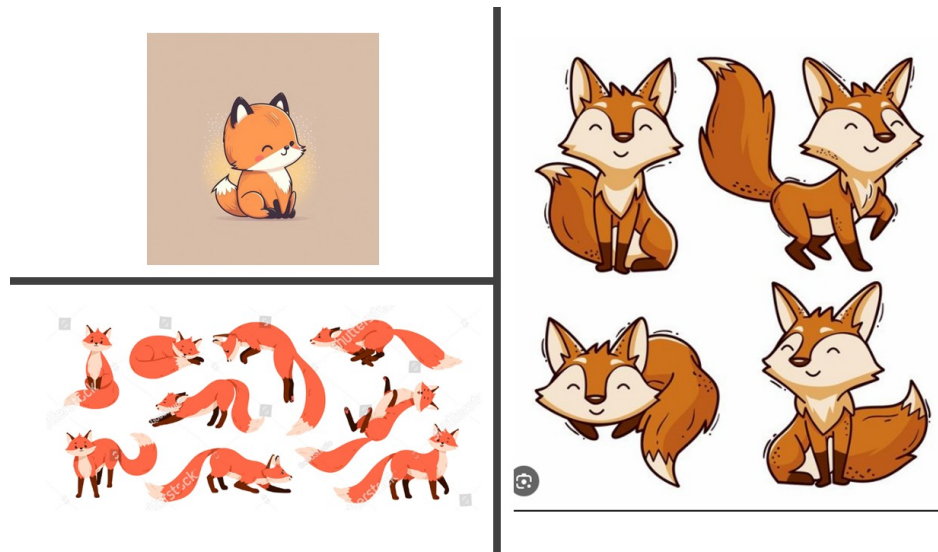


Figura E.38: Referencia zorro cumpeo cartoon 1.

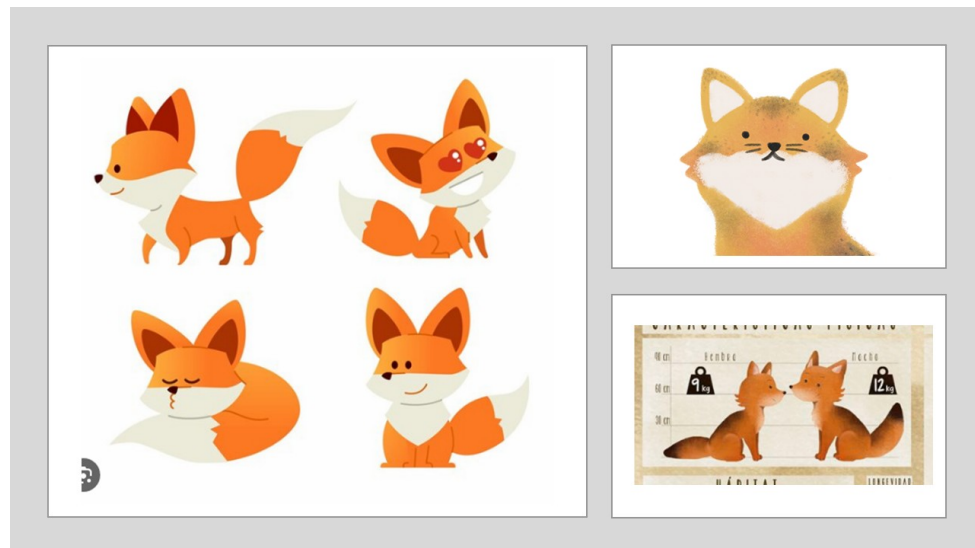


Figura E.39: Referencia zorro cumpeo cartoon 2.

E.3.3. Habitación



Figura E.40: Referencia Bosque 1.



Figura E.41: Referencia Bosque 2.

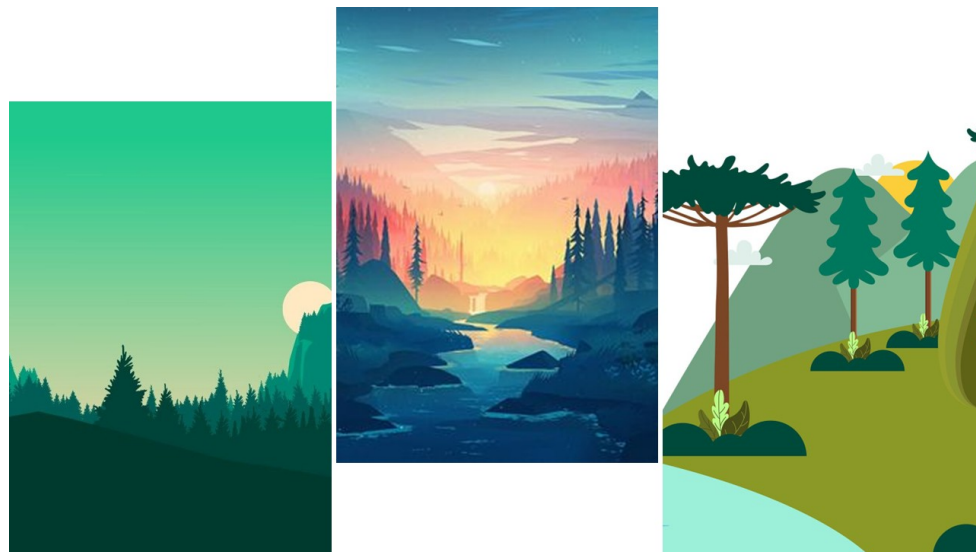


Figura E.42: Referencia Bosque minimalista.



Figura E.43: Referencia desierto 1.



Figura E.44: Referencia desierto 2.



Figura E.45: Referencia desierto minimalista.

E.3.4. Diseño de personajes

Primera iteración

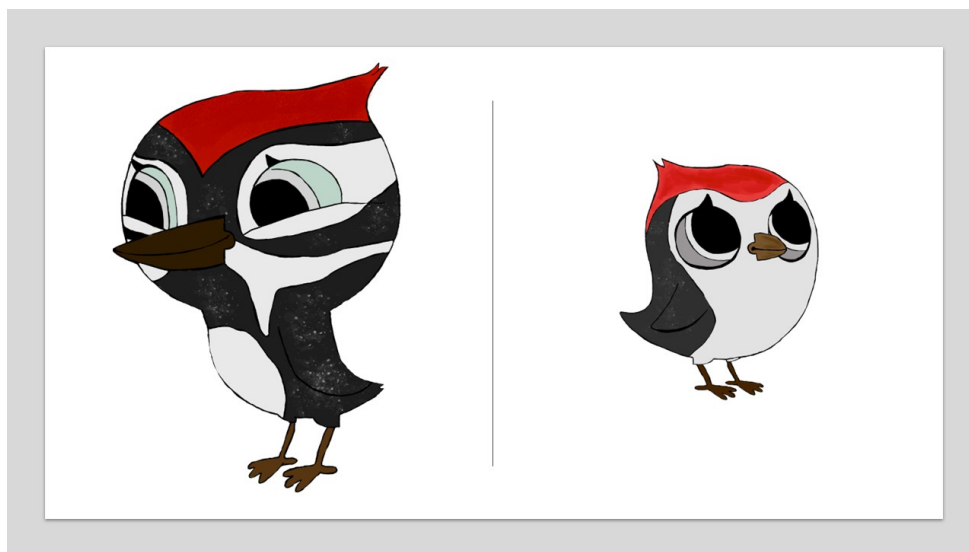


Figura E.46: Primer acercamiento, idea descartada de inmediato.

Segunda iteración

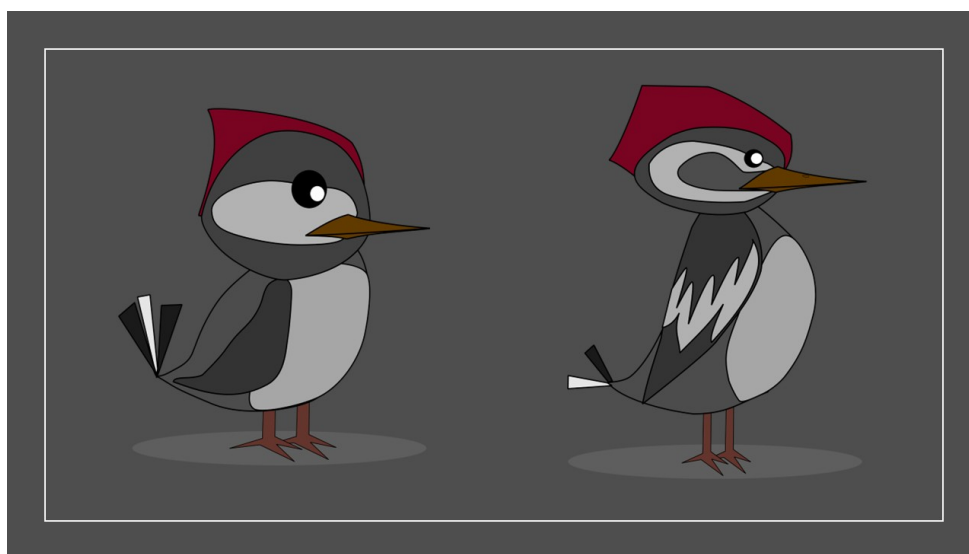


Figura E.47: Carpintero estructura geométrica.

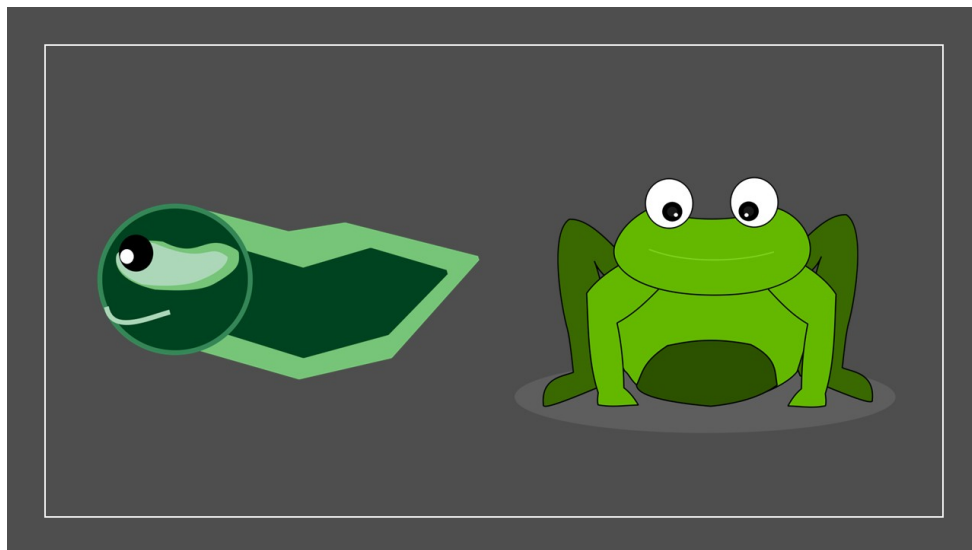


Figura E.48: Sapo estructura geométrica.

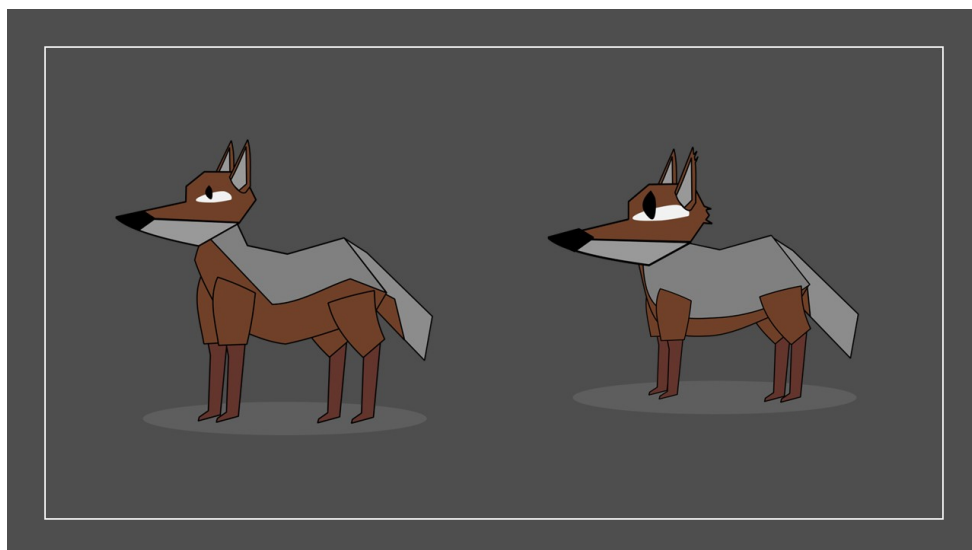


Figura E.49: Zorro cumpeo estructura geométrica.

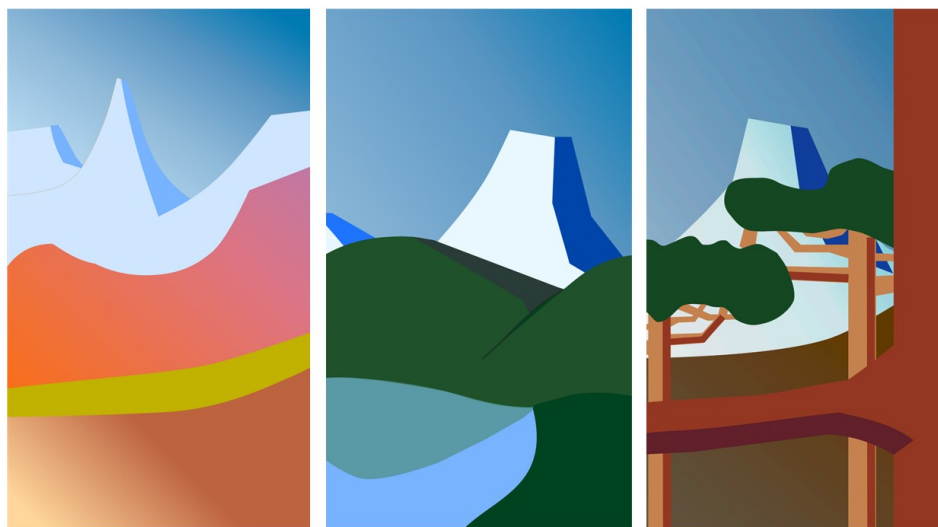


Figura E.50: Habitación estructura geométrica minimalista.

Mascotas finales



Figura E.51: Carpintero final.

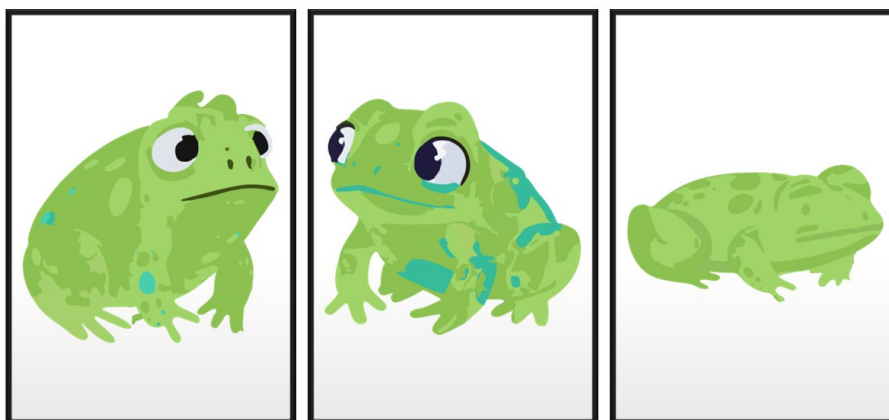


Figura E.52: Sapo final.

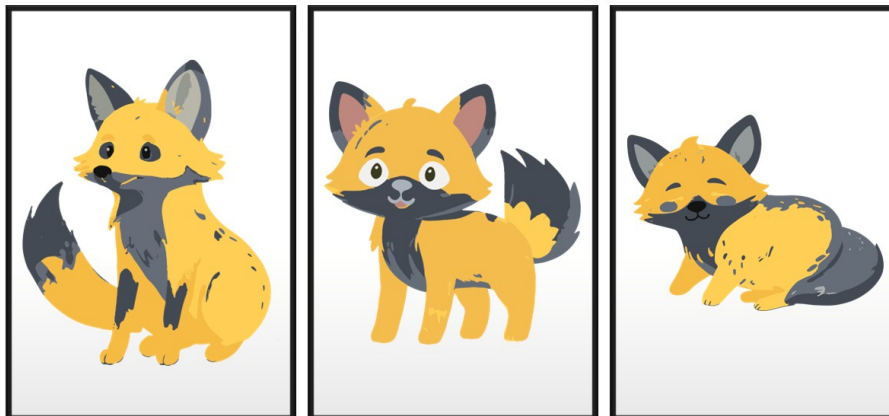


Figura E.53: Zorro cumpleo Final.

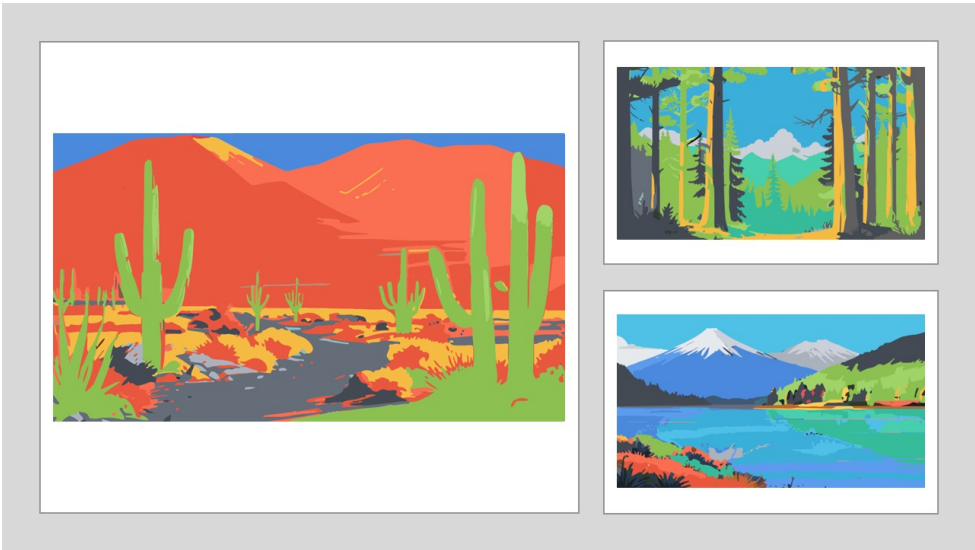


Figura E.54: Habitat finales.

E.3.5. Paleta de colores

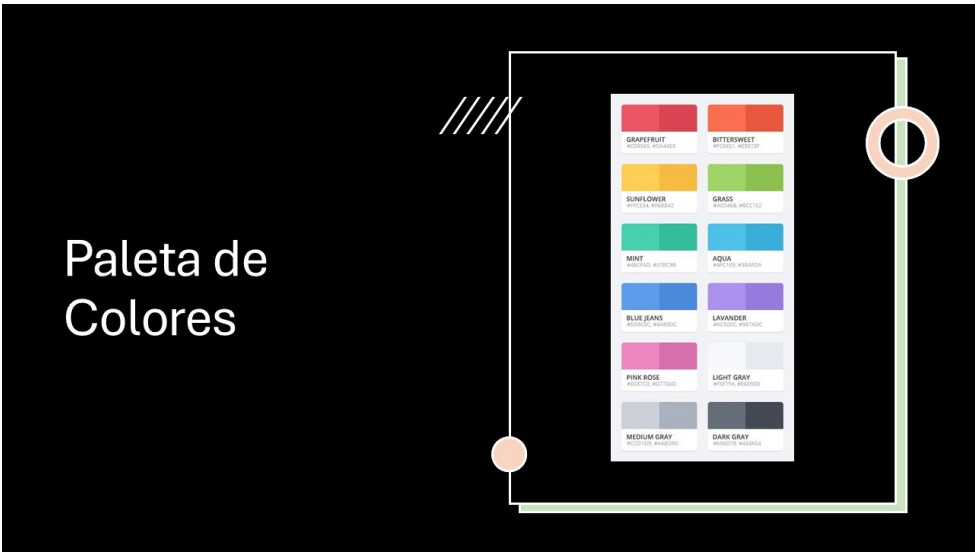


Figura E.55: Paleta de colores utilizada.

E.3.6. Concepto inicial interfaz

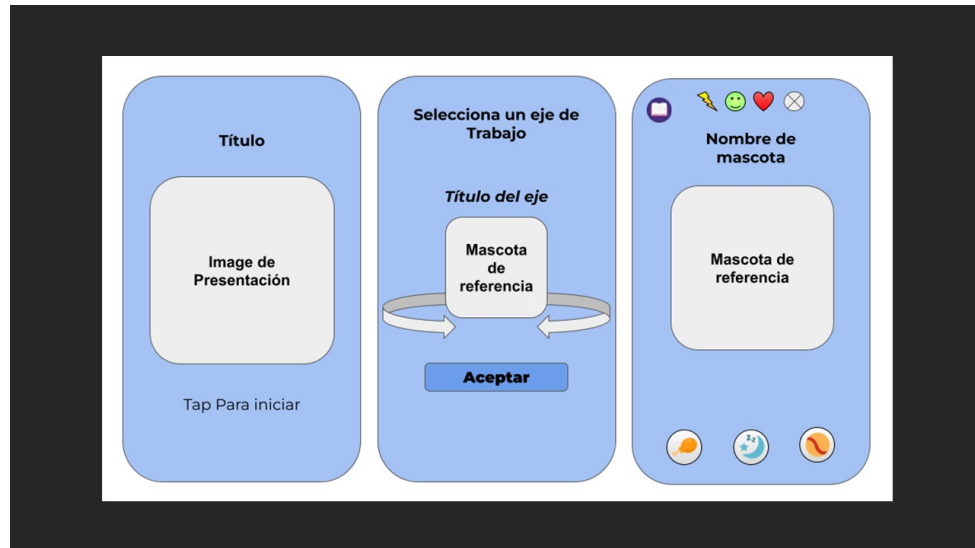


Figura E.56: Concepto interfaz 1 menú principal.

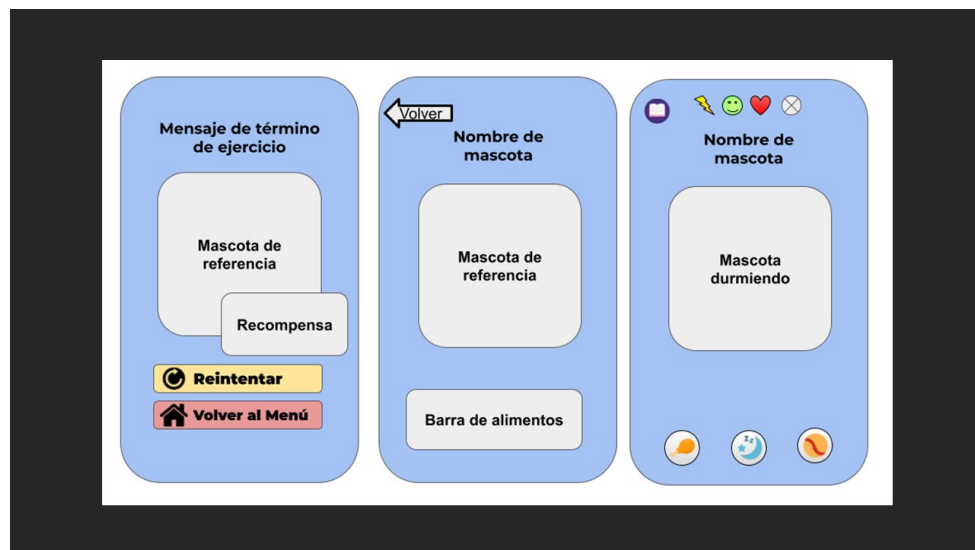


Figura E.57: Concepto interfaz 2 casa mascota.

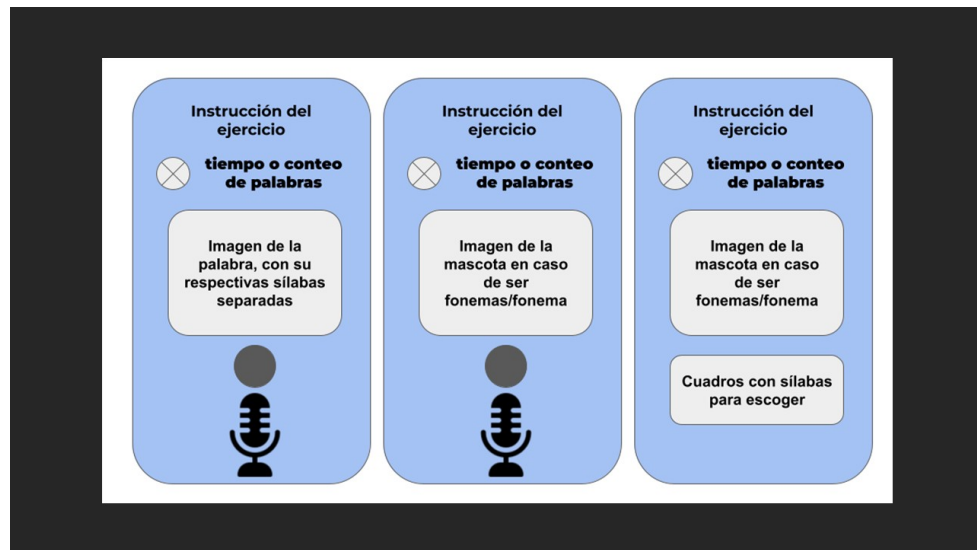


Figura E.58: Concepto interfaz 3 minijuegos.

E.3.7. Interfaz primera build



Figura E.59: Interfaz inicio Prototipo 1.

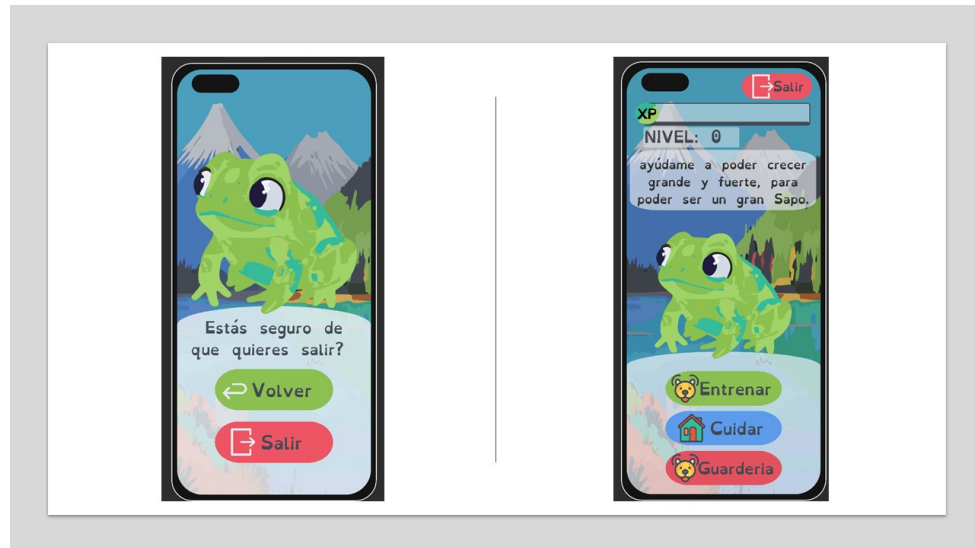


Figura E.60: Interfaz menú principal y pregunta de cierre Prototipo 1.



Figura E.61: Interfaz menú ejercicios y videojuegos Prototipo 1.



Figura E.62: Interfaz casa Prototipo 1.

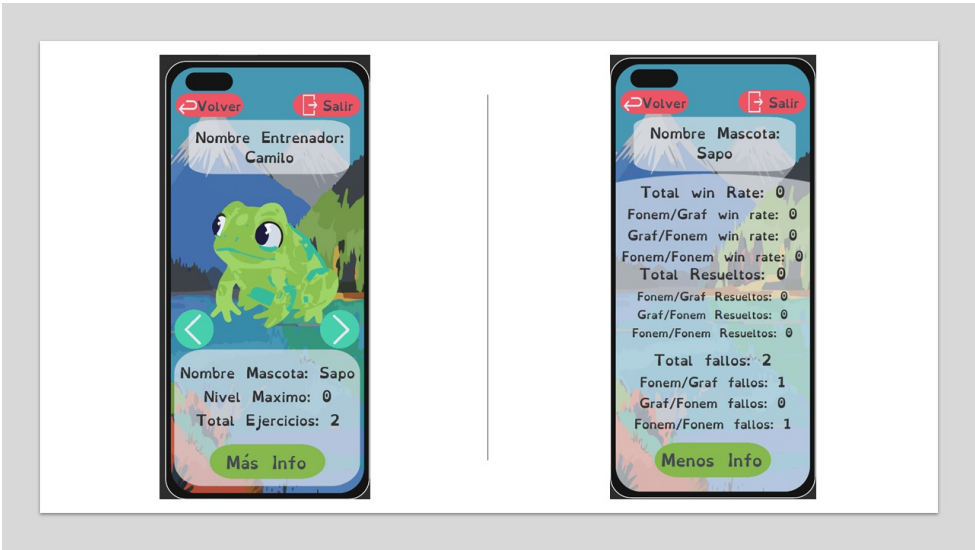


Figura E.63: Interfaz galería Prototipo 1.

E.3.8. Interfaz build final

Figura E.64: Interfaz inicio Prototipo final.



Figura E.65: Interfaz selección de mascota Prototipo final.

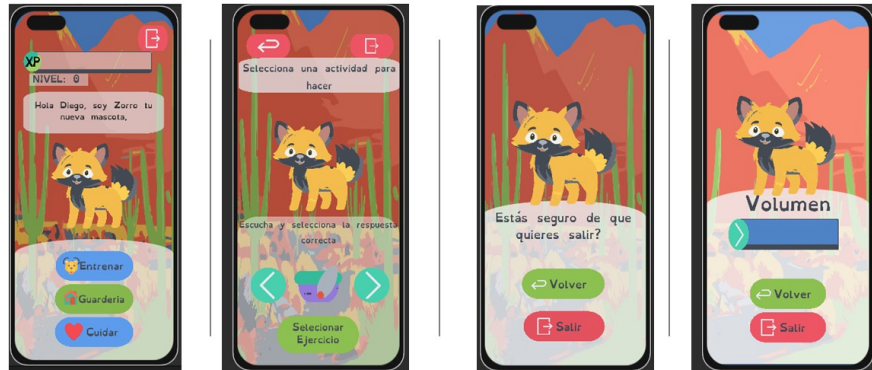


Figura E.66: Interfaz menús y pregunta de seguridad salida prototipo final.



Figura E.67: Interfaz minijuegos y termino de entrenamiento prototipo final.



Figura E.68: Interfaz casa prototipo final.

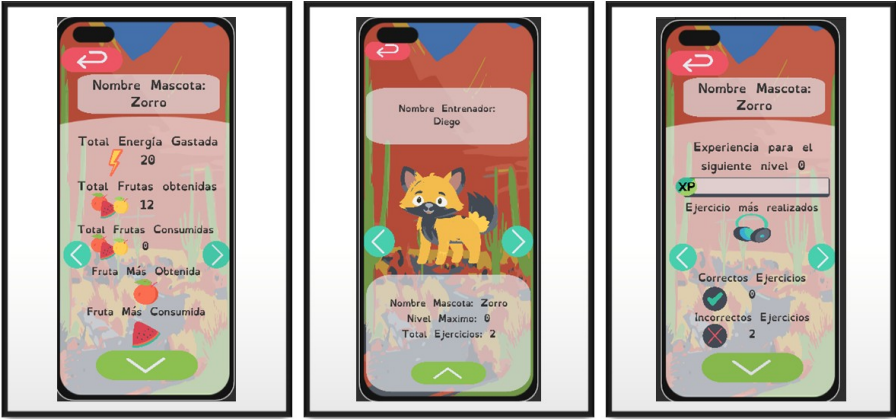


Figura E.69: Interfaz galería Prototipo final.