

**UNIVERSIDAD AUTÓNOMA DE MADRID**  
**ESCUELA POLITÉCNICA SUPERIOR**



**Grado en ingeniería informática**

**TRABAJO FIN DE GRADO**

**Desarrollo de una aplicación de apoyo a la lectura global para  
personas con necesidades especiales**

**Ignacio Aceña Alcázar**  
**Tutor: Javier Gómez Escribano**  
**Ponente: Germán Montoro Manrique**

**Enero 2017**



# **Desarrollo de una aplicación de apoyo a la lectura global para personas con necesidades especiales**

**AUTOR: Ignacio Aceña Alcázar**  
**TUTOR: Javier Gómez Escribano**

**Dpto. Ingeniería Informática**  
**Escuela Politécnica Superior**  
**Universidad Autónoma de Madrid**  
**Enero de 2017**



# Resumen

Leer es un aprendizaje determinante para la vida de las personas, supone una integración en todos los ámbitos. La lectura otorga una independencia, siendo esta un medio de comunicación y al mismo tiempo, una herramienta indispensable para desenvolvernó en la sociedad actual. En el caso específico de las personas con Trastornos del Espectro Autista (TEA), tienen que adquirir las habilidades sociales y de comunicación, de las que carecen, para poder relacionarse adecuadamente [1].

Desde el punto de vista educativo, se ha observado que los niños con TEA presentan numerosas dificultades a la hora de aprender siguiendo el método de imitación o modelado, por lo que es necesario utilizar estrategias específicas dirigidas al aprendizaje lector. En concreto todas estas personas presentan una característica en común, que son aprendices visuales. Por este motivo, una de las maneras de trabajar es a través de la relación imagen-objeto, debido a que en ocasiones los niños presentan dificultades a la hora de relacionar la palabra con el objeto que representan [2].

Este proyecto desarrolla una aplicación basada en el método de lectura global, dirigido a las personas con TEA con objeto de acercarlas a la lectura y favorecer su aprendizaje posterior de carácter más formal.

El objetivo de este trabajo fin de grado es, en primer lugar, proporcionar los materiales utilizados en el proyecto “Leo con Lula”<sup>1</sup> para pizarra digital, método de lectura global, de manera estructurada y ordenada para facilitar el aprendizaje. En segundo lugar, obtener información de la progresión de los usuarios por parte de los educadores.

La tecnología y especialmente las tabletas tienen todo lo necesario para cumplir estos objetivos. Se decidió realizar el trabajo para el sistema operativo Android, debido a las ventajas en cuanto a accesibilidad y facilidad de uso que tiene respecto a otros sistemas.

Este documento describe el proceso llevado a cabo para la consecución de estos objetivos. En él se recogen las diferentes etapas llevadas a cabo, se explican las fases de análisis, diseño, así como la implementación del sistema y la evaluación.

Se puede concluir que se alcanza una aplicación que cumple los objetivos. Se ha conseguido que la aplicación sea atractiva y motivadora para los usuarios teniendo de manera ordenada el material de aprendizaje, así como las capacidades de uso y gestión pedidas por los expertos.

---

<sup>1</sup> Web oficial “Leo con Lula”: <http://leoconlula.com> (último acceso:19/01/2017)

## **Palabras clave**

Aprendizaje, integración, Trastorno del Espectro Autista, lectura global, pictogramas, tecnología, tableta, accesibilidad, TIC.

# Abstract

Reading is a specific life skill required in today's world, it supports integration in all areas. Reading provides independence, it is a communication media and at the same time, an indispensable tool to increase development in today's society. In the specific case of people with Autism Spectrum Disorders (ASD), they have to acquire social and communication skills, which they may lack, in order to be able to relate properly [1].

From the educational point of view, it has been observed that children with ASD face many difficulties in learning following the method of imitation or modelling, which is why it is necessary to use specific strategies aimed at reading learning. Specifically all these people have a common characteristic, that they are visual apprentices. For this reason, one of the ways of working is through the image-object relationship, because children can sometimes present difficulties in relating the word to the object they represent [2].

This project develops an application based on the global reading method, aimed at people with ASD in order to bring them closer to reading and improve their later learning of a more formal nature.

The aim of this final grade paper is, firstly, to provide the materials used in the "Leo con Lula" project for digital slate, a global reading method, in a structured and orderly way to facilitate learning. Second, get information on the progression of users by educators.

The technology and especially the tablets have everything necessary to fulfil these objectives. It decided to do the work for the Android operating system, due to the advantages in terms of accessibility and ease of use it has with respect to other systems.

This document describes the process carried out to achieve these objectives. It covers the different stages carried out, explains the phases of analysis, design, as well as the implementation of the system and evaluation.

It can be concluded that an application is achieved that meets the objectives. It has made the application attractive and motivating for the users having in an orderly way the learning material, as well as the use and management capacities requested by the experts.

# Keywords

Learning, integration, Autism Spectrum Disorders, global reading, pictograms, technology, Tablet, accessibility, ICT.



## ***Agradecimientos***

Este apartado quería aprovecharlo para agradecer a todas esas personas que estando a mi lado han hecho este trabajo posible.

Para comenzar, quería agradecer a mi tutor Javier Gómez Escribano su entrega y apoyo a lo largo de este proyecto, así como darle las gracias por darme la oportunidad de llevarlo a cabo.

No quiero olvidarme del colegio de educación especial I.P.P Quintero Lumbreras, por hacer posible el desarrollo de este trabajo, sin todos los que forman el centro no habría sido posible.

Seguidamente, quiero dar las gracias a todos mis familiares y amigos que durante tantos años han estado a mi lado y me han hecho este camino mucho más fácil.

Para finalizar quiero destacar a mis padres Javier y Sagrario, a mi hermana Alba y a mi novia Sol, son las personas más importantes que tengo, gracias por estar siempre a mi lado en los momentos buenos y malos, este trabajo es en parte vuestro, sin vosotros no habría sido posible.



## ÍNDICE DE CONTENIDOS

1	Introducción.....	1
1.1	Motivación.....	1
1.2	Objetivos.....	2
1.3	Organización de la memoria.....	2
2	Estado del arte .....	3
2.1	La lectura global .....	3
2.2	La lectura global y el uso de las TICs .....	4
2.3	Aplicaciones de Lectura Global para personas con necesidades especiales .....	4
2.4	Conclusiones.....	8
3	Desarrollo del Proyecto .....	9
3.1	Análisis de requisitos.....	9
3.2	Diseño.....	16
4	Implementación .....	25
4.1	Librerías utilizadas .....	25
4.2	Esquema de la implementación .....	27
5	Evaluación del proyecto .....	39
5.1	Proceso de evaluación .....	39
6	Conclusiones y trabajo futuro.....	41
6.1	Conclusiones.....	41
6.2	Trabajo futuro .....	41
	Referencias .....	42
	Glosario .....	43
	Anexos.....	44
A.	Manual de instalación y uso .....	44
B.	Cuestionario evaluación “Leo con Lula”.....	53

## ÍNDICE DE FIGURAS

FIGURA 1: PROCESO INVERSO DE LA LECTURA GLOBAL FRENTE A LA LECTURA SILÁBICA. <i>FUENTE:</i> [4]. .....	3
FIGURA 2: CAPTURA PANTALLA APLICACIÓN PALABRAS ESPECIALES. ....	5
FIGURA 3: CAPTURA PANTALLA APLICACIÓN JUGAMOS TODOS. ....	6
FIGURA 4: CAPTURA PANTALLA APLICACIÓN ADAPRO. ....	7
FIGURA 5: CAPTURA PANTALLA “LEO CON LULA” PIZARRA DIGITAL. ....	8
FIGURA 6: DIAGRAMA DE LA ARQUITECTURA DEL SISTEMA.....	17
FIGURA 7: PANTALLA SESIÓN “LEO CON LULA” PARA PIZARRA DIGITAL. ....	19
FIGURA 8: PROTOTIPO MENÚ INICIAL.....	20
FIGURA 9: PROTOTIPO PANTALLA AJUSTES.....	21
FIGURA 10: PROTOTIPO DIÁLOGO SELECCIÓN AJUSTES. ....	21
FIGURA 11: PROTOTIPO PANTALLA GESTIÓN VOCABULARIO.....	21
FIGURA 12: DIAGRAMA DE ACTIVIDAD PANTALLAS. ....	23
FIGURA 13: DIAGRAMA DE LA BASE DE DATOS.....	24
FIGURA 14: CÓDIGO CLASE VOLLEY SINGLETON.JAVA.....	25
FIGURA 15: CÓDIGO DE LA CLASE INTERFAZ SERVIDOR ACTIVITY.JAVA. ....	26
FIGURA 16: CÓDIGO CARGA DE IMAGEN A TRAVÉS DE LA LIBRERÍA PICASSO.....	26
FIGURA 17: CÓDIGO DEL MÉTODO MAKE REQUEST DE LA CLASE MAIN FRAGMENT.JAVA. ....	29
FIGURA 18: RESPUESTA JSON OBTENIDA DE LA API DE ARASAAC. ....	30
FIGURA 19: RESPUESTA JSON OBTENIDA DE LA API DE SYLLABER. ....	30
FIGURA 20: CÓDIGO DEL MÉTODO API SILABEO DE LA CLASE MAIN FRAGMENT.JAVA.....	30
FIGURA 21: CÓDIGO DEL BOTÓN DE CAMBIAR DE USUARIO DE LA CLASE MENU INICIAL.JAVA. ....	31
FIGURA 22: PARTE 1 DE CÓDIGO PARA CREAR CAPTCHA DE LA CLASE MENU INICIAL.JAVA.....	32
FIGURA 23: PARTE 2 DE CÓDIGO PARA CREAR CAPTCHA DE LA CLASE MENU INICIAL.JAVA.....	32
FIGURA 24: CÓDIGO INICIALIZACIÓN DE LOS LISTENER DE LOS CONTENEDORES Y DEL RÓTULO. ...	33

FIGURA 25: CÓDIGO MÉTODO ONLONGCLICK COMÚN A TODAS LAS SESIONES DE JUEGO. ....	34
FIGURA 26: PARTE DE CÓDIGO DEL MÉTODO ONDRAG, CASE CORRESPONDIENTE A SOLTAR EL RÓTULO.....	35
FIGURA 27: PARTE DE CÓDIGO DEL MÉTODO CARGARPANTALLAINICIAL DE LAS SESIONES DE JUEGO.....	36
FIGURA 28: CÓDIGO PARA EXTRAER LAS PALABRAS DE LA BASE DE DATOS DE LA CLASE SELECCIONARVOCABULARIO.JAVA.....	37
FIGURA 29: OPCIONES RESPUESTA CUESTIONARIO SUS.....	39
FIGURA 30: GRAFICO RANGOS DEL PERCENTIL ASOCIADOS A LAS PUNTUACIONES SUS. FUENTE: [9].....	40
FIGURA 31: CAPTURA MANUAL USO, DIÁLOGO “LEO CON LULA” CONDICIONES DE USO. ....	44
FIGURA 32: CAPTURA MANUAL USO, MENÚ INICIAL. ....	45
FIGURA 33: CAPTURA MANUAL USO, AÑADIR USUARIO. ....	45
FIGURA 34: CAPTURA MANUAL USO, DIÁLOGO PREGUNTA SEGURIDAD.....	46
FIGURA 35: CAPTURA MANUAL USO, MENÚ AJUSTES. ....	46
FIGURA 36: CAPTURA MANUAL USO, PANTALLA GESTIÓN VOCABULARIO.....	47
FIGURA 37: CAPTURA MANUAL USO, PANTALLA SELECCIÓN PALABRAS JUEGO.....	48
FIGURA 38: CAPTURA MANUAL USO, PANTALLA SELECCIÓN SESIÓN. ....	49
FIGURA 39: CAPTURA MANUAL USO, TURNO DE LA SESIÓN I.....	49
FIGURA 40: CAPTURA MANUAL USO, REFUERZO ANTE ACIERTO. ....	50
FIGURA 41: CAPTURA MANUAL USO, ASOCIAR SESIÓN I ANTE CONTENEDOR CONTENIDO NULO....	50
FIGURA 42 CAPTURA MANUAL USO, PUNTUACIÓN AL FINALIZAR SESIÓN. ....	51
FIGURA 43: CAPTURA CUESTIONARIO EVALUACIÓN, DATOS DEL USUARIO. ....	53
FIGURA 44: CAPTURA CUESTIONARIO EVALUACIÓN, CUESTIONARIO USABILIDAD PARTE I.....	54
FIGURA 45: CAPTURA CUESTIONARIO EVALUACIÓN, CUESTIONARIO USABILIDAD PARTE II. ....	55
FIGURA 46: CAPTURA CUESTIONARIO EVALUACIÓN, PREGUNTAS SOBRE LA EXPERIENCIA DE USO DE LEO CON LULA. ....	56



# 1 Introducción

---

## 1.1 Motivación

Desde mediados de los años 80 el empleo de materiales gráficos, analógicos o más simbólicos como la escritura, se ha erigido en herramientas imprescindibles para los profesionales que trabajan con personas con Trastornos del Espectro Autista (TEA). La inconsistencia en la comprensión de la información verbal, la temporalidad limitada de los mensajes verbales, y el hecho de que los actos comunicativos impliquen necesariamente relaciones interpersonales, son algunas de las causas por las que la información ofrecida en formato verbal sea a menudo desatendida, no comprendida e incluso aversiva [3].

Muchos niños con autismo requieren de un sistema adaptado para aprender a leer. La observación de los lectores "naturales" indica la facilidad para el aprendizaje por sistemas globales, es decir aquellos en que se asocia el objeto con el rótulo escrito de la palabra correspondiente, lo que se denomina "lectura global [4]". Uno de los sistemas adaptados para aprender a leer que existen es el método "Leo con Lula" para pizarra digital.

"Leo con Lula", como proyecto de lectura global para pizarra digital, nace en el año 2011 del trabajo de dos maestras de educación especial: Guadalupe Montero y Laura Muñino. El objetivo principal es ofrecer un método de lectura gratuito, atractivo y eficaz para todas las personas que requieran de un método globalizado para aprender a leer y de este modo mejorar las competencias académicas y personales de los alumnos.

Adaptar el proyecto a nuevos formatos TIC permite mejorar la difusión y el uso de "Leo con Lula". En este aspecto la creación de la aplicación para tableta es un aspecto clave que permitirá hacer llegar este recurso de aprendizaje a muchos más niños y niñas, posibilitando a su vez el uso generalizado en diferentes contextos de aprendizaje: escuela, hogar, asociaciones, clases de apoyo, etc.

## **1.2 Objetivos**

El objetivo de este trabajo es diseñar e implementar un sistema que, mediante el uso de tabletas basadas en el sistema Android, den apoyo a la introducción de un procedimiento de lectura global en las personas TEA como manera de acercarlas a la lectura.

Además, se tendrá que realizar una mejora respecto al proyecto existente “Leo con Lula” para pizarra digital. El sistema deberá permitir personalizar diferentes aspectos para adaptarlo a las características e intereses personales de cada niño o niña que la utilice: tipografía, mayúsculas, minúsculas, vocabulario, comportamiento de la aplicación ante el acierto y comportamiento de la aplicación ante el error.

Por otra parte, se pretende dar soporte a los educadores, así como al grupo de padres facilitándoles la recogida de estadísticas sobre el trabajo realizado por los niños de manera que puedan llegar a conclusiones sobre estos resultados favoreciendo la evaluación continua del niño en cuanto al progreso de su aprendizaje.

Con el fin de facilitar la comprensión por parte de los usuarios del sistema, los pictogramas usados para la aplicación serán los de ARASAAC<sup>2</sup>, ya que son los más utilizados en nuestro país tanto para señalética, como para agendas y comunicadoras.

Para conseguirlo se realizará un sistema completo, desde el análisis del método “Leo con Lula” ya existente para pizarra digital, como diferentes aplicaciones para niños TEA, pasando por el análisis de usuarios, tareas y requisitos de software, el diseño, la elaboración, la implementación y la evaluación final.

## **1.3 Organización de la memoria**

La memoria del proyecto se divide en los siguientes capítulos:

- Capítulo 1. Introducción: motivación y objetivos del proyecto.
- Capítulo 2. Estado del arte.
- Capítulo 3. Desarrollo del proyecto: análisis, diseño.
- Capítulo 4. Implementación del proyecto.
- Capítulo 5. Evaluación del proyecto.
- Capítulo 6. Conclusiones y trabajo futuro.
- Referencias y apéndices.

---

<sup>2</sup> Portal de la comunicación aumentativa y alternativa: <http://arasaac.org/> (último acceso:19/01/2017)

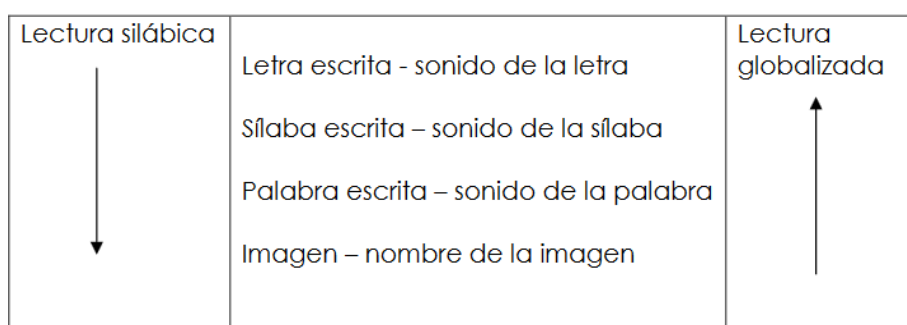


## 2 Estado del arte

### 2.1 La lectura global

La lectura global es un método que facilita aprender a leer, el cual trabaja las palabras apoyándose en imágenes, está basado en reconocer globalmente la palabra escrita percibiendo la palabra como un todo y relacionándola con su significado, es decir, es importante que se entienda que a partir de pictogramas se accede a significados, a mensajes [4].

La lectura global es un proceso inverso a la lectura que conocemos como tradicional, como podemos ver en la **Figura 1**.



**Figura 1: Proceso inverso de la lectura global frente a la lectura silábica. Fuente: [4].**

#### 2.1.1 ¿Por qué utilizar el método de lectura global en las personas con necesidades especiales?

Las personas con necesidades especiales TEA tienen generalmente dificultades a la hora de aprender a leer por el método tradicional que se usa normalmente en las escuelas, por ello necesitan un método que les facilite el aprendizaje.

Los métodos de lectura global suelen ser eficaces en este colectivo de personas debido a las siguientes características:

- Las personas TEA tienen desarrollado las competencias visuales, lo que permite que diferencien las imágenes y relacionarlas con su significado [5].
- Las personas TEA presentan dificultades a la hora de saber para qué sirve una actividad. Por ello en el momento de aprender a leer es necesario enseñarles palabras que representen actividades conocidas por los alumnos, de esta forma, es más conveniente la lectura global respecto al método silábico o tradicional en estas personas [5].

## **2.2 La lectura global y el uso de las TICs**

Las nuevas tecnologías de la información y la comunicación (TICs) presentan una herramienta útil para los educadores en cualquier ámbito del proceso de enseñanza y aprendizaje [6].

En concreto, en la lectura global las TICs nos permiten mantener la atención del alumno en la actividad, lo que facilita el aprendizaje.

Específicamente para trabajar con los niños TEA, las TICs proporcionan una estimulación de sus sentidos ya que combinan audio, vídeo e incluso en algunas tecnologías como los móviles y tabletas, vibración [6].

Otro punto a favor de las TICs en la lectura global es que permiten tener el material perfectamente organizado dentro de la tecnología, así como transportarlo de manera fácil a cualquier lugar, es decir, poder trabajar indistintamente desde casa o desde el aula. Del mismo modo, es esencial involucrar al grupo de padres en el proceso de aprendizaje, mandándoles los materiales, con lo que conseguimos generalizarlos, es decir, que la persona vaya por el mismo punto de aprendizaje tanto en el aula como en su casa, y aprenda con el mismo contenido de trabajo [7].

Configuran una herramienta que nos permite adaptar el material para cada persona, lo que es indispensable en el proceso de lectura global ya que nos posibilitan aumentar o disminuir, indistintamente el nivel de ayuda dependiendo de las necesidades específicas de cada uno, cambiando desde el tipo de letra hasta variar los estímulos visuales y sonoros.

En el proceso de la evaluación por parte del docente, las TICs también tienen una relevancia importante ya que permiten tener organizadas las estadísticas de cada alumno de manera que puede saber en cualquier momento cómo va el proceso de aprendizaje y reforzar las palabras en las que presenten dificultades.

## **2.3 Aplicaciones de Lectura Global para personas con necesidades especiales**

En la actualidad existen pocas aplicaciones de lectura global debido a que normalmente se trabaja en las aulas a través de fichas y material en papel. A continuación, se van a citar algunas de estas aplicaciones que se han considerado más relevantes y que tienen ciertas similitudes con la funcionalidad de “Leo con Lula” para Android.

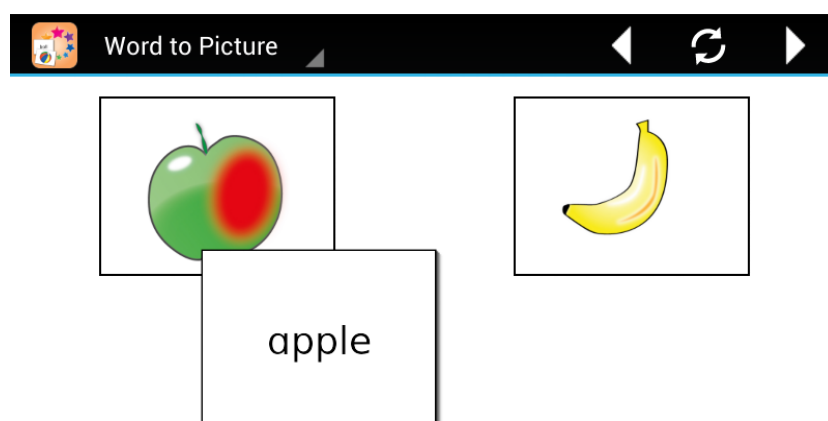
### 2.3.1 Palabras especiales



Esta aplicación de pago disponible para Android e IOs, permite a través de una interfaz simple e intuitiva captar la atención de las personas con necesidades especiales. Es una aplicación para niños con problemas de aprendizaje que tiene las siguientes características:

- Estimula el habla y ayuda a la claridad.
- Amplía el vocabulario de palabras frecuentes.
- Desarrolla la coordinación viso-manual.
- Mejora la motricidad fina.
- Personalizable con propias palabras fotos<sup>3</sup>.

En cuanto a la lectura global esta aplicación puede prepararse para el método, ya que tiene un modo de emparejar palabra con imágenes como se puede ver en la **Figura 2**, se asocia el rótulo apple con la imagen que muestra la manzana.



**Figura 2:** Captura pantalla aplicación palabras especiales.

<sup>3</sup> <https://play.google.com/store/apps/details?id=com.specialiapps.specialwords&hl=es> (último acceso:19/01/2017)

## 2.3.2 Jugamos Todos



Jugamos Todos es una aplicación de pago disponible solamente para IOS, está preparada para la integración de personas con Síndrome de Down y autismo en las tecnologías modernas. Su funcionalidad se basa en el aprendizaje de objetos a través de experiencias tecnológicas<sup>4</sup>.

Para la lectura global podemos utilizar el primer nivel de esta aplicación ya que tiene un modo similar a la descrita en el apartado 2.3.1, de emparejar la palabra con la imagen como se puede ver en la **Figura 3**, el problema es que solo se pueden manejar las mismas palabras, en concreto animales.



**Figura 3:** Captura pantalla aplicación Jugamos Todos.

---

<sup>4</sup> <https://itunes.apple.com/es/app/jugamos-todos/id588386762?mt=8> (último acceso:19/01/2017)

### 2.3.3 Adapro



Adapro es un procesador de textos gratuito para Windows, Linux y MacOS, orientado a personas con dificultades de aprendizaje como dislexia o autismo. Es sencillo e intuitivo a su vez se puede configurar para adaptarlo a las características de la persona con el objetivo de mantener su atención.

En cuanto a la lectura global se puede usar para trabajar asociaciones de imagen palabra como se puede ver en la **Figura 4**.

Una similitud a destacar con Leo con lula es que usa los pictogramas de ARASAAC.



**Figura 4:** Captura pantalla aplicación Adapro.

### 2.3.4 Leo con Lula para pizarra digital



“Leo con Lula” para pizarra digital es un proyecto que ofrece una serie de material orientados a aprender a leer por el método de la lectura global.

“Leo con Lula” cuenta con diferentes fases de aprendizaje en las que el niño va a ir adquiriendo diferente léxico. Las palabras se irán trabajando a través de las sesiones de entrenamiento, aprendizaje, discriminación y discriminación inversa. Para utilizarlo es necesario tener instalado el software Smart Notebook.

En este material nos hemos basado para la realización del presente proyecto. Se siguen tanto las fases de aprendizaje como el método de aprendizaje adaptándolas a las opciones de personalización que nos permiten las tabletas. En la **Figura 5**, se puede ver una ronda de la sesión de discriminación donde se tiene que asociar el rótulo palomitas con el pictograma del cuadro derecho.

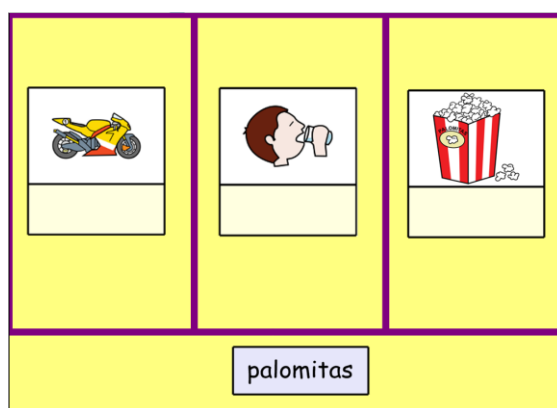


Figura 5: Captura pantalla “Leo con Lula” pizarra digital.

## 2.4 Conclusiones

En conclusión, podemos decir que existen muy pocas aplicaciones que se centren exclusivamente en la lectura global y menos que lleguen a la gran mayoría de usuarios. La lectura global sigue utilizando material tradicional para su desarrollo, es decir, no está muy integrado en las TICs. Podemos encontrar en internet numeroso material en formato PDF para imprimir y trabajar en papel, pero eso dificulta la adaptación del material a cada alumno, la posibilidad de tener el material organizado y las facilidades para realizar la evaluación de los alumnos.

Cabe destacar que “Leo con Lula” se ha realizado para Android para llegar a un número más amplio de personas ya que se pueden encontrar dispositivos mucho más baratos con este sistema operativo respecto al resto de sistemas y que la aplicación se podrá descargar gratuitamente desde Play Store.

Por lo tanto, la aplicación “Leo con Lula” para Android será muy útil tanto para el grupo de maestros como el grupo de padres y las personas a las que va dirigido el sistema para facilitar el desarrollo del proceso lector.

## 3 Desarrollo del Proyecto

---

### 3.1 Análisis de requisitos

Para que el desarrollo del proyecto concluya con éxito cumpliendo los objetivos descritos en el apartado 1.2, en esta sección se realiza un análisis de requisitos que nos permite tener claro el funcionamiento del sistema.

El proyecto se va a centrar en el usuario experto, siguiendo un diseño incremental ya que este se basa en conocer y comprender las necesidades, limitaciones, comportamiento y características del usuario final en base a la experiencia del experto. Trabajando de esta manera se evitará alargar el proyecto, consiguiendo una herramienta útil con mayor rapidez, involucrando al experto en la realización del proyecto ya que se le irá enviando de manera periódica los avances del sistema para evitar caer en el error.

Se ha estimado que esta es una metodología que cumple los objetivos propuesto del proyecto, tanto para permitir a los usuarios ajustar las características del sistema a su propio aprendizaje como para que los educadores obtengan los datos necesarios para llevar a cabo el proceso de evaluación de sus alumnos.

El análisis está dividido en dos apartados que clasifican los requisitos en requisitos funcionales y requisitos no funcionales. La primera categoría define las funcionalidades que se implementarán en la aplicación, la segunda categoría define los criterios que debe satisfacer las operaciones de la aplicación. Estos requisitos fueron extraídos de las reuniones celebradas con el experto, así como del material “Leo con Lula” disponible para pizarra digital que nos proporcionaron.

Una vez terminado el análisis, tendremos la información necesaria para desarrollar el sistema de la manera correcta. Esta fase es una de las más importantes y críticas dentro del desarrollo de software.

#### 3.1.1 Requisitos funcionales

En este apartado se muestran los requisitos acerca de la lógica de la aplicación, los que detallan tanto las funcionalidades disponibles para los usuarios como las internas y necesarias para gestionar la lógica de la aplicación.

**Requisitos de funcionalidades disponibles para los usuarios durante el uso de la aplicación.** Las funcionalidades de la aplicación que están disponibles para el usuario, es decir, los requisitos relacionados con la interacción que puede realizar el usuario o visualizar en la aplicación, los cuales son los siguientes:

**RF(01)** Crear un usuario nuevo.

- Se debe crear un usuario para poder cambiar los ajustes y jugar una partida.
- Al iniciar por primera vez la aplicación e intentar realizar cualquier interacción se solicitará crear un usuario.
- Se podrá crear un usuario nuevo en cualquier momento desde la pantalla de ajustes, pulsando el botón añadir usuario, escribiendo el nombre escogido en el cuadro de texto y pulsando aceptar.
- Se producirá error y se mostrará un diálogo de alerta indicándolo si el nombre de usuario escogido ya existe en la aplicación.

**RF(02)** Cambiar tipo de letra Mayúsculas/Minúsculas

- Se podrá cambiar el tipo de letra a visualizar en la partida entre mayúsculas o minúsculas.
- Se cambiará el tipo de letra a visualizar desde la pantalla de ajustes, pulsando el botón minúsculas/mayúsculas, cuyo icono es A+, y pulsando en el diálogo que aparece la opción elegida.
- Únicamente se podrá realizar esta acción desde la pantalla de ajustes una vez se haya creado un usuario.

**RF(03)** Cambiar tipo de fuente Helvética/Chalkboard

- Se podrá cambiar el tipo de fuente a visualizar en la partida entre helvética o chalkboard.
- Se cambiará el tipo de fuente a visualizar desde la pantalla de ajustes, pulsando el botón helvética/chalkboard, cuyo icono es F, y pulsando en el diálogo que aparece la opción elegida.
- Únicamente se podrá realizar esta acción desde la pantalla de ajustes una vez se haya creado un usuario.

**RF(04)** Escoger si se quieren ayudas visuales o no durante la partida

- Se podrá escoger entre si se quieren ayudas visuales o no se quieren ayudas visuales durante las partidas.
- Se elegirá si se quieren ayudas visuales o no desde la pantalla de ajustes, pulsando el botón con ayuda/sin ayuda, cuyo icono es un libro con una interrogación, y pulsando en el diálogo que aparece la opción elegida.
- Únicamente se podrá realizar esta acción desde la pantalla de ajustes una vez se haya creado un usuario.

**RF(05)** Escoger sonido ante acierto o no durante la partida.

- Se podrá escoger entre si se quiere sonido ante acierto o no durante las partidas.
- Se elegirá si se quiere sonido ante acierto o no desde la pantalla de ajustes, pulsando el botón con sonido/sin sonido, cuyo icono es una cara feliz, y pulsando en el diálogo que aparece la opción elegida.
- Únicamente se podrá realizar esta acción desde la pantalla de ajustes una vez se haya creado un usuario.



**RF(06)** Escoger sonido ante error o no durante la partida.

- Se podrá escoger entre si se quiere sonido ante error o no durante las partidas.
- Se elegirá si se quiere sonido ante error o no desde la pantalla de ajustes, pulsando el botón con sonido/sin sonido, cuyo icono es una cara triste, y pulsando en el diálogo que aparece la opción elegida.
- Únicamente se podrá realizar esta acción desde la pantalla de ajustes una vez se haya creado un usuario.

**RF(07)** Escoger lectura de palabras o no durante la partida.

- Se podrá escoger entre si se quiere lectura de palabras o no durante las partidas.
- Se elegirá si se quiere lectura de palabras o no desde la pantalla de ajustes, pulsando el botón sí leer/no leer, cuyo icono es un altavoz, y pulsando en el diálogo que aparece la opción elegida.
- Únicamente se podrá realizar esta acción desde la pantalla de ajustes una vez se haya creado un usuario.

**RF(08)** Eliminar un usuario de la aplicación.

- Se podrá eliminar un usuario en cualquier momento desde la pantalla de ajustes, pulsando el botón eliminar usuario, la cual mostrará una lista con los usuarios registrados, se seleccionará de la lista el usuario a eliminar.
- Se producirá error no eliminándose el usuario y se mostrará un cuadro de alerta indicándolo si el usuario escogido es el usuario activo o el último usuario de la aplicación, siempre tiene que haber al menos un usuario registrado.

**RF(09)** Visualizar estadísticas de usuario.

- Se podrá visualizar estadísticas de usuario, en cualquier momento desde la pantalla de ajustes, pulsando el botón de estadísticas de usuario.
- Se mostrará la pantalla de estadísticas donde se visualizarán como mínimo las sesiones realizadas por el usuario, fecha de la sesión, hora de la sesión, las palabras utilizadas así como los números de aciertos y fallos por sesión.
- Únicamente se podrá realizar esta acción desde la pantalla de ajustes una vez se haya creado un usuario.

**RF(10)** Gestionar vocabulario.

- Se podrá añadir palabras al vocabulario para jugar en cualquier momento desde la pantalla de ajustes, pulsando el botón gestionar vocabulario.
- Se mostrará la pantalla de gestión de vocabulario donde se podrá buscar la palabra a añadir mediante un buscador en la parte superior de la pantalla que enlaza con los pictogramas de ARASAAC. En la parte inferior de la pantalla se mostrarán los pictogramas buscados mediante una lista. Para añadir la palabra al vocabulario tendremos que pulsar sobre el pictograma elegido y esta se añadirá al sistema.
- Se guardará en la base de datos, el pictograma, el nombre del pictograma, y las sílabas correspondiente al nombre de la palabra.
- Únicamente se podrá realizar esta acción desde la pantalla de ajustes una vez se haya creado un usuario.

**RF(11)** Gestionar número de turnos por sesión.

- Se podrá gestionar el número de turnos por sesión desde la pantalla de ajustes, pulsando el botón gestionar número de juegos.
- Se mostrará una lista con el número de turnos disponibles. El número mínimo de turnos es 10 ya que menos de eso no conseguiría ningún aprendizaje. Habrá que pulsar sobre el número elegido y se abrirá un diálogo preguntando si se desea realizar el cambio de número de turnos, con las opciones de sí y de no.
- Únicamente se podrá realizar esta acción desde la pantalla de ajustes una vez se haya creado un usuario.

**RF(12)** Cambiar de usuario

- Se podrá cambiar de usuario, entre los usuarios creados dentro de la aplicación.
- Se cambiará de usuario desde la pantalla de ajustes o desde la pantalla inicio, pulsando sobre el botón situado en la esquina superior izquierda con el nombre del usuario activo actualmente, se mostrará una lista de los usuarios creados y se deberá pulsar sobre el usuario elegido. Finalmente se mostrará un diálogo preguntando si se desea realizar el cambio de usuario, con las opciones de sí y de no.
- Únicamente se podrá realizar esta acción desde la pantalla de ajustes o desde la pantalla de inicio si hay más de un usuario creado.

**RF(13)** Seleccionar vocabulario para la sesión de juego.

- Se podrá seleccionar vocabulario desde la pantalla de selección de vocabulario. Para acceder a ella se pulsará el botón jugar, este estará disponible tanto en el menú inicial como el menú de ajustes.
- Se mostrarán dos listas en la pantalla de selección de vocabulario así como un botón siguiente que enlazará con la pantalla de selección de sesión. La primera lista mostrará todas las palabras añadidas en la gestión de vocabulario a ese usuario. La segunda lista será una lista vacía donde se añadirán las tres palabras seleccionadas de la primera lista para comenzar la sesión de aprendizaje.
- Para seleccionar las palabras simplemente se pulsará sobre la palabra a añadir de la primera lista y se añadirá a la segunda lista, habrá que añadir tres palabras.
- Para continuar hacia la pantalla de selección de sesión habrá que pulsar el botón siguiente. Se producirá error si no se han añadido tres palabras a la lista mostrando un mensaje de alerta.
- Únicamente se podrá realizar esta acción desde la pantalla de selección de vocabulario.

**RF(14)** Seleccionar sesión a iniciar.

- Se podrá seleccionar la sesión a iniciar desde la pantalla de selección de sesión. Para acceder a ella se tendrá que haber seleccionado previamente el vocabulario y pulsar siguiente en la pantalla de selección de vocabulario.
- Se mostrarán los diferentes niveles de juego, los cuales serán: asociación imagen-palabra/palabra, asociación imagen/palabra y descomposición silábica. Cada nivel mostrará las sesiones del método “Leo con Lula” que son cuatro: entrenamiento, aprendizaje y profundización, discriminación y discriminación inversa<sup>5</sup>.
- Para seleccionar la sesión se pulsará el botón de la sesión escogida. Es recomendable realizar las sesiones en orden. Una vez pulsado el botón, comenzará el juego de aprendizaje con los ajustes escogidos por el usuario en la pantalla de ajustes.
- Únicamente se podrá realizar esta acción desde la pantalla de selección de sesión.

---

<sup>5</sup> La descomposición silábica solo cuenta con una sesión.

**RF(15)** Deslizar cuadros que contienen el rótulo de la palabra a contenedor correspondiente

- Dentro de la pantalla de juego de cualquiera de las sesiones, se podrán deslizar los cuadros con el rótulo de palabra escrito por la pantalla de juego.
- Al terminar de deslizar el cuadro por la pantalla, al soltar, el usuario deberá ver reflejado su acción, es decir, si se suelta en el contenedor correcto se añadirá el cuadro de la palabra al contenedor y se mostrará una señal de acierto, en caso de añadirse al contenedor incorrecto el cuadro de la palabra volverá a su posición inicial y se señalará el error.
- Únicamente se podrá realizar esta acción desde la pantalla de juego de cualquiera de las sesiones.

**RF(16)** Reiniciar sesión

- Al finalizar la sesión se mostrará un diálogo preguntando al usuario si quiere repetir la sesión. Para reiniciar la sesión deberá pulsar el botón reintentar nivel.
- Únicamente se podrá realizar esta acción al finalizar la pantalla de juego de cualquiera de las sesiones.

**RF(17)** Pasar a la siguiente sesión

- Al finalizar la sesión se mostrará un diálogo preguntando al usuario si quiere pasar a la siguiente sesión. Para pasar a la siguiente sesión deberá pulsar el botón siguiente nivel.
- Únicamente se podrá realizar esta acción al finalizar la pantalla de juego de cualquiera de las sesiones.

**Requisitos de lógica interna del sistema.** Las funcionalidades que el sistema realiza de manera interna, sin orden del usuario:

**RF(18)** Calcular puntuación del usuario al finalizar la sesión

- Al finalizar la sesión en función de los datos guardados durante el desarrollo de la sesión **RF(19)** se mostrará una puntuación de forma gráfica, por ejemplo, unas estrellas.

**RF(19)** Almacenar datos correspondientes de la sesión

- La información guardada será la siguiente: identificador de la sesión, fecha de la sesión, palabras de la sesión, número de aciertos y número de errores.

### 3.1.2 Requisitos no funcionales

La calidad del sistema aumentará si se cumplen los siguientes requisitos no funcionales:

#### Requisitos de interfaz:

**RNF(01)** La interfaz debe ser “minimalista”, ayudando a los usuarios a centrar la atención en el aprendizaje.

**RNF(02)** Los colores de las pantallas elegidos permitirán al usuario distinguir los distintos elementos que se muestran, así como diferenciar las acciones que realiza sin dar lugar a equivocación.

**RNF(03)** En el transcurso del juego se deberá diferenciar claramente los diferentes contenedores así como los distintos cuadros que contienen el rótulo de la palabra. De esta manera el usuario deberá ser capaz de reconocer los cuadros que se arrastran y los contenedores donde se sueltan las palabras.

**RNF(04)** Permanencia de forma al mostrar los mismos elementos en diferentes pantallas.

#### Requisitos de fiabilidad:

**RNF(05)** El sistema mostrará mensajes de alerta en caso de producirse cualquier error, ya sea por acciones incorrectas realizadas por el usuario como por fallos del sistema.

**RNF(06)** El sistema debe recuperarse de fallos, sin producirse pérdida de datos.

#### Requisitos de rendimiento:

**RNF(07)** Las transacciones deben realizarse en menos de 5 segundos.

#### Requisitos de funcionalidad

**RNF(08)** La aplicación será evaluada por expertos de manera que se detecten posibles fallos, así como introducir mejoras sugeridas por ellos.

#### Requisitos de interacción:

**RNF(09)** La aplicación debe poder usarse por todas las personas capaces de tocar y deslizar sobre una pantalla táctil.

**RNF(10)** Los botones deben reflejar la posibilidad de hacer click ya sea con la forma del botón o con el texto de dentro.

**RNF(11)** Los mensajes y alertas deben ser fáciles de entender para cualquier usuario.

## 3.2 Diseño

En este apartado se va a desarrollar la etapa correspondiente al diseño de la aplicación. En concreto lo referente a la lógica del sistema, el diseño de los elementos interactivos, el diseño de la interfaz así como el diagrama de navegación por la aplicación. Al concluir la etapa de diseño se podrá iniciar la etapa de implementación.

### 3.2.1 Diseño de la arquitectura

La arquitectura general del sistema se puede ver representada en la **Figura 6**, como se observa, el usuario entra en contacto con el programa a partir de un dispositivo táctil, ya sea móvil o tableta, es decir, es el intermediario entre el usuario y el programa.

El programa es el que se ocupa de coger de la base de datos la información necesaria guardada anteriormente: usuarios registrados, último usuario activo en el sistema, preferencias de usuario, vocabulario e estadísticas de usuario. A partir de estos datos el programa crea las sesiones de juego. A medida que el usuario realiza acciones en la aplicación, este guarda los datos en la base de datos, ya sean ajustes de usuario, usuarios nuevos, estadísticas de sesiones jugadas, vocabulario etc.

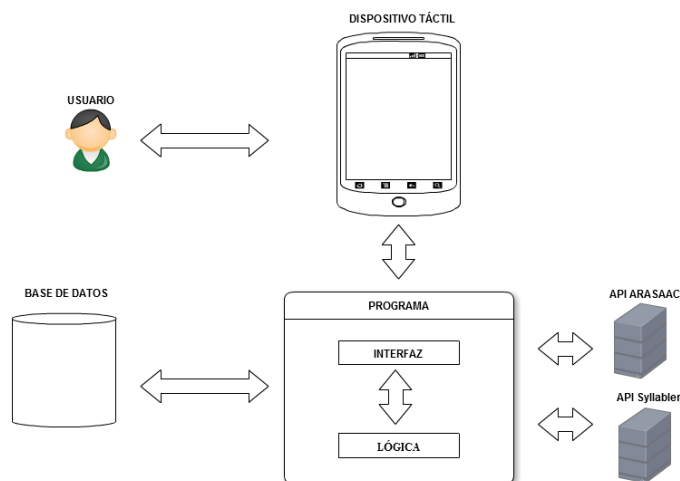
A continuación, se van a definir los elementos principales de la arquitectura de la aplicación:

- **Usuario:** Persona que a través de la tableta digital usará el programa.
- **Tableta digital:** Es el dispositivo encargado de poner en contacto al usuario con el programa.
- **Programa:** Parte de la arquitectura que está en contacto con la base de datos, las APIs y con el usuario por medio de la tableta digital, en el programa se implementa la lógica y la interfaz.
- **Base de datos:** La base de datos usa el modelo relacional, está pensada para almacenar los usuarios de la aplicación, las preferencias de cada usuario, las estadísticas de cada usuario, el vocabulario etc.
- **APIs:** El programa se comunicará con dos APIs. La primera API utilizada es la de ARASAAC, es la encargada de proporcionar los pictogramas y las palabras al programa, este los almacenará en la base de datos. La segunda API es la de Syllaber<sup>6</sup>, la cual se encargará de dividir las palabras en las sílabas necesarias para la fase tres, descomposición silábica, el programa guardará la cadena silabeada en la base de datos.

La base de datos se encontrará guardada dentro de la propia memoria local de la tableta digital, ya que no es necesario compartir los datos que se guardarán con el resto de usuarios de otras tabletas digitales.

---

<sup>6</sup> Web Oficial "Syllaber": <http://syllabler.fwok.org/> (último acceso:19/01/2017)



**Figura 6: Diagrama de la arquitectura del sistema.**

### 3.2.2 Diseño de la lógica del sistema

En este apartado se describe la lógica necesaria para el método de lectura global llevado a cabo, es decir, se describen las diferentes fases y sesiones que habrá que implementar.

Para comenzar vamos a definir los elementos y su funcionalidad:

- **Cuadro con rótulo de la palabra:** Se trata del cuadro que contiene el rótulo con el nombre de la palabra a asociar, se podrá deslizar a los diferentes contenedores.
- **Caja:** El sistema dispondrá de tres cajas por cada pantalla, una para cada palabra disponible en la sesión. No se tendrán por qué mostrar las tres cajas en todas las rondas, podrán aparecer una, dos o tres. Cada una podrá contener los siguientes elementos dependiendo del nivel:
  - **Pictograma:** Imagen que representa el objeto al que se refiere.
  - **Nombre pictograma:** Palabra con el nombre del pictograma.
  - **Contenedor:** Recuadro donde se añadirá el cuadro con rótulo de la palabra, es decir, donde se arrastra y se produce la asociación.

En algunas rondas se mostrará alguna caja de contenido nulo, es decir sin pictograma y sin nombre, solo con el contenedor.

En cuanto al juego de aprendizaje el sistema tendrá tres niveles o tipos de juego:

- Asociación imagen-palabra/palabra:

En la pantalla se presenta, en la parte superior las cajas de las diferentes palabras que se quieren mostrar, en cada caja aparecerá un pictograma, una palabra con el nombre del pictograma y el contenedor donde se arrastrará el cuadro con rótulo de la palabra. En la parte inferior el cuadro con el rótulo de la palabra a asociar arrastrando con su pictograma y nombre.

- Asociación imagen palabra:

Su funcionalidad es similar que el nivel de Asociación imagen-palabra/palabra, la única diferencia es que en la parte superior de la pantalla en las cajas de las diferentes palabras solo aparece el pictograma y el contenedor donde se realiza la asociación.

- Descomposición silábica:

Similar a los dos niveles anteriores, con la diferencia de que en vez de asociar un rótulo a un contenedor, habrá que asignar varios rótulos a varios contenedores, ya que la palabra se descompone en sílabas.

Cada uno de los niveles se compone de cuatro sesiones y un número de rondas escogidas en ajustes por el usuario:

- Entrenamiento:

Esta sesión es para que el alumno se familiarice con el juego, primero se muestra solo un contenedor de la palabra con el rótulo en la parte inferior para comprobar que el alumno identifica el pictograma y realiza la asociación. A continuación, mostraremos dos contenedores uno el de la palabra a asociar y otro de contenido nulo. Esto se hará con las tres palabras escogidas para la sesión.

- Aprendizaje y profundización:

En esta sesión el objetivo es que el alumno asocie la palabra con la imagen correcta. Para ello se mostrarán dos contenedores en cada ronda, tanto el cuadro con rótulo de la palabra como los contenedores irán cambiando en cada ronda entre el grupo de tres palabras escogido para la sesión.

- Discriminación:

Esta sesión es similar al aprendizaje y profundización, con la diferencia de que en vez de mostrar dos contenedores se mostrarán tres en cada ronda, aumentando la dificultad al alumno.

- Discriminación inversa<sup>7</sup>:

En esta sesión se trabaja una palabra de manera diferente. En cada ronda habrá un contenedor y dos rótulos de palabras en la parte inferior. El alumno deberá asociar el rótulo correspondiente con el pictograma.

---

<sup>7</sup> La descomposición silábica solo cuenta con una sesión.

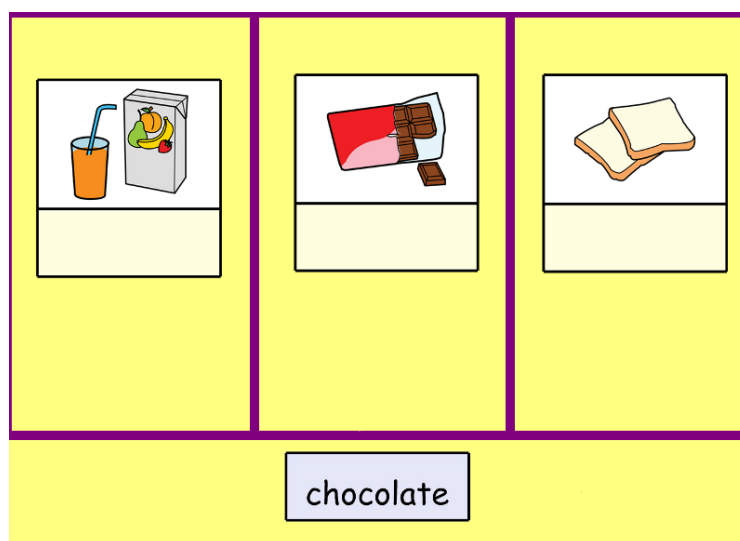


### 3.2.3 Diseño interfaz de usuario

La interfaz de usuario debe cumplir los requisitos descritos en el apartado 3.1, por ello el diseño escogido es minimalista, lo que facilitará el uso de la aplicación a los usuarios finales.

Las pantallas serán todas similares, se usarán bajo un sistema de botones con la misma estética, forma y color. Los diálogos y cuadros de texto serán simples.

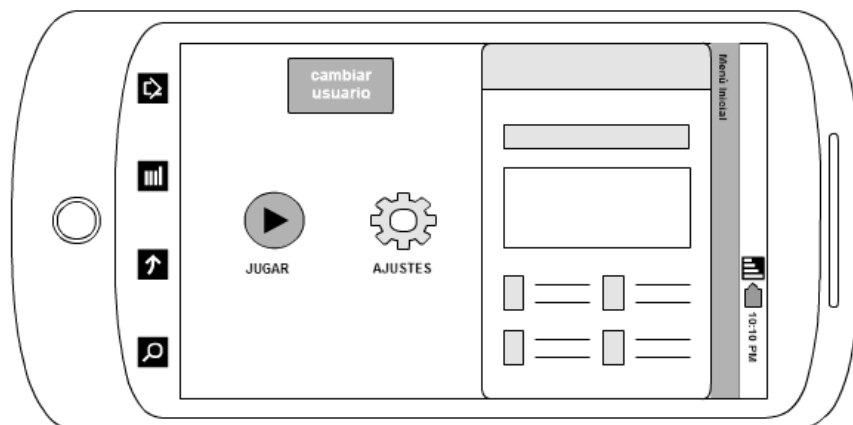
Para el diseño de la interfaz de las pantallas correspondientes a las sesiones de juego no se ha realizado prototipos ya que se sigue el diseño del material “Leo con Lula” para pizarra digital. Con esto se favorece el cambio de usuarios entre diferentes tecnologías, este diseño se puede ver en la **Figura 7**, en la cual se observan tres pictogramas y la palabra “chocolate”. El usuario tendrá que arrastrar la palabra “chocolate” debajo del pictograma que representa chocolate. Estas pantallas están diseñadas para evitar la distracción del usuario haciendo en la mayor medida posible que este centre su atención en el aprendizaje. Por ello se mostrarán solo los pictogramas y palabras necesarias para cada ronda de la sesión y solo se permitirá al usuario realizar la acción de arrastre de la palabra correspondiente al juego de aprendizaje.



**Figura 7: Pantalla sesión “Leo con Lula” para pizarra digital.**

En cuanto al diseño de los menús, se han realizado diferentes prototipos. A continuación, describimos los más relevantes para el entendimiento de cómo se diseñó el sistema. Estos son: menú inicial, menú de ajustes, pantalla de gestión de vocabulario así como el prototipo de un diálogo del sistemas.

El menú inicial seguirá el prototipo que se puede observar en la **Figura 8**, está compuesto de tres botones, en la parte izquierda de la pantalla, y un texto, en parte derecha de la pantalla. El botón de la parte superior izquierda de la pantalla sirve para cambiar de usuario, al presionarlo desplegará una lista con los usuarios en el sistema. Los dos de la parte inferior izquierda, serán para entrar en la pantalla de selección de vocabulario para jugar y de cambio de ajustes. En la parte derecha se mostrará información de en qué consiste el proyecto “Leo con Lula”.



**Figura 8: Prototipo menú inicial.**

En cuanto al menú de ajustes, como se puede ver en la **Figura 9**, se compone de trece botones. Los botones de mayúsculas/minúsculas, cambio de fuente, ayuda visual, sonido en acierto, sonido en error, ayuda lectura, añadir usuario, eliminar usuario, número de juegos y cambio de usuario. Al pulsarlos mostrarán un diálogo. Dependiendo de la funcionalidad del botón, el diálogo incluirá una lista de usuarios u opciones. En el caso de cambiar de usuario, eliminar usuario y número de rondas se incluirá una lista. Un cuadro de texto se añadirá en el diálogo del botón añadir usuario. El diálogo que mostrará el resto será de unos botones con las opciones disponible tal como se puede ver en la **Figura 10**. En este caso se trata del botón mayúsculas/minúsculas, el cual muestra qué tipo de letra prefiere el usuario junto a dos botones con las opciones disponibles.

En cuanto a los botones jugar, estadísticas y gestión de vocabulario nos llevarán a la pantalla que realice la funcionalidad correspondiente. En la **Figura 11**, se puede ver el prototipo de la pantalla de gestión de vocabulario. Consiste en un buscador que irá conectado a la API de ARASAAC, al buscar una palabra se mostrará en una lista las opciones disponibles.

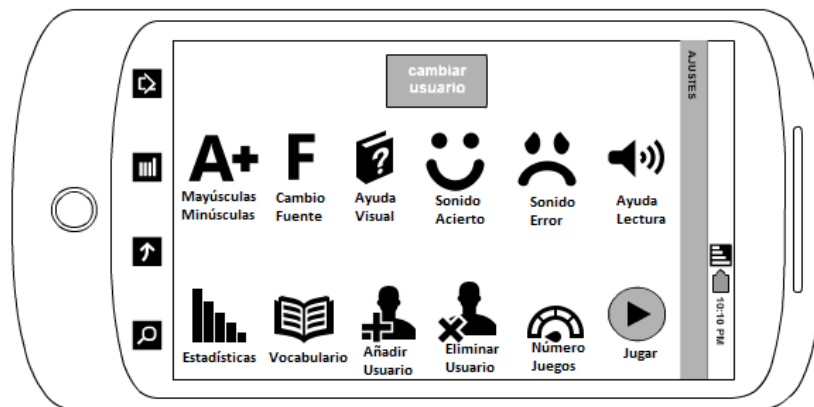


Figura 9: Prototipo pantalla ajustes.

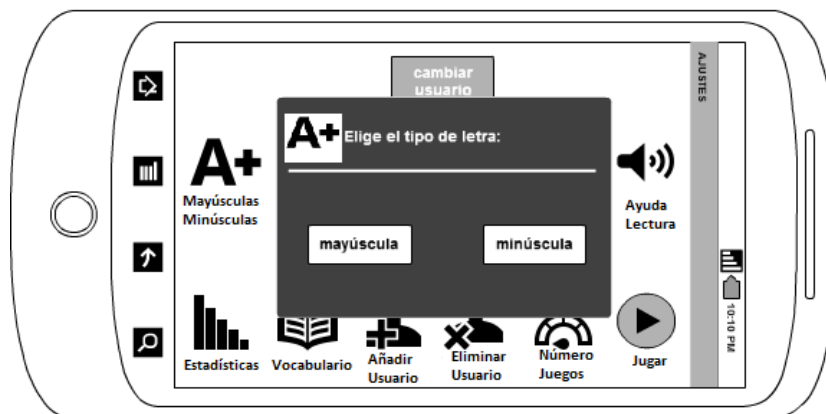


Figura 10: Prototipo diálogo selección ajustes.

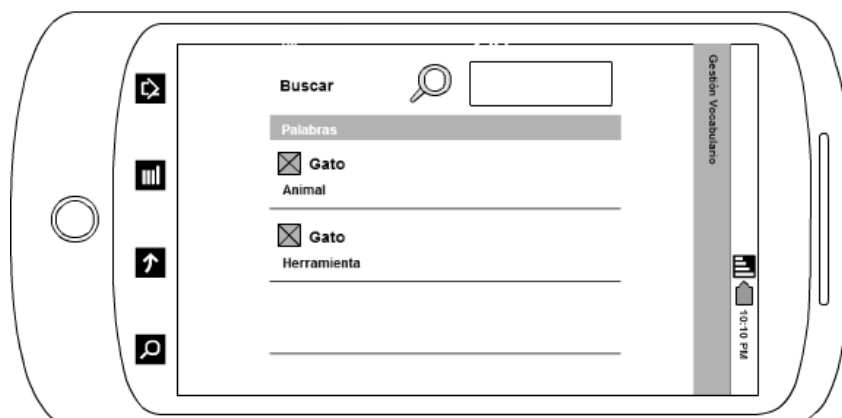
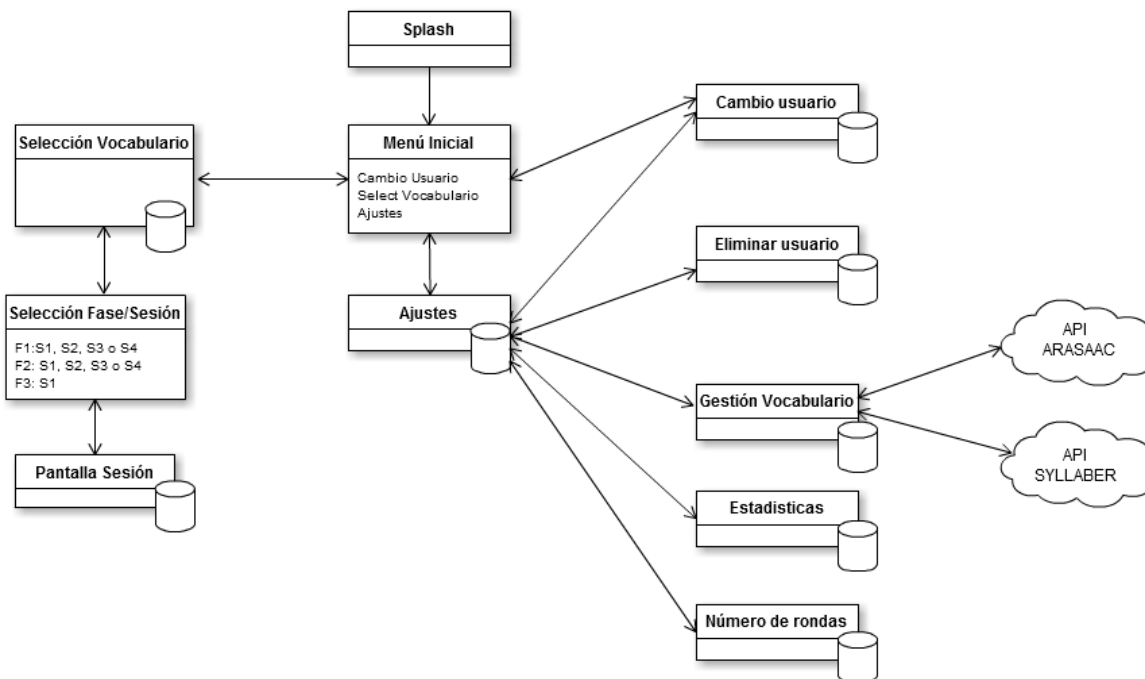


Figura 11: Prototipo pantalla gestión vocabulario.

### 3.2.4 Estructura de la aplicación

La aplicación está estructurada en diferentes pantallas, cada pantalla está formada por una parte lógica y una parte gráfica. El diseño de la estructura se puede ver en el diagrama de actividad de la **Figura 12**, en él se pueden observar las diferentes pantallas.

- **Splash:** Pantalla de presentación inicial, muestra logo del sistema y da paso al menú inicial.
- **Menú inicial:** Desde el menú inicial se puede acceder a cambiar de usuario, seleccionar vocabulario y a la pantalla de ajustes.
- **Selección Vocabulario:** Pantalla para seleccionar las tres palabras para el juego de aprendizaje de la base de datos, antes de seleccionar la fase y sesión a jugar.
- **Pantalla de selección de fase/sesión:** Tres pantallas deslizantes una por cada fase. En cada una se mostrarán las diferentes sesiones en forma de botones, al pulsar sobre los botones se pasará a la pantalla de la sesión pulsada.
- **Pantalla sesión:** Pantalla donde se realiza el juego de aprendizaje, habrá una para cada tipo de sesión explicadas en el apartado **3.2.2**.
- **Ajustes:** Pantalla de ajustes compuesta por botones que al pulsarlos mostrarán un dialogo para cambiar el ajuste o darán paso a las pantallas de cambio usuario, eliminar usuario, número de rondas, gestión de vocabulario o estadísticas.
- **Cambio usuario:** Pantalla en la que por medio de una lista se mostrará los usuarios de la base de datos y se podrá seleccionar al que se quiera cambiar.
- **Eliminar usuario:** Pantalla en la que por medio de una lista se mostrará los usuarios de la base de datos y se podrá seleccionar al que se quiera eliminar.
- **Numero de rondas:** Pantalla en la que por medio de una lista se mostrará el número de rondas y se podrá seleccionar cuantas rondas durará cada sesión.
- **Gestión vocabulario:** Esta pantalla se conecta con las APIs, mediante un buscador y una lista donde se mostrarán los resultados de la búsqueda, se podrá añadir pulsando sobre la lista la palabra que se quiera a la base de datos, esta guardará la palabra silabeada que la conseguirá de la conexión con la API Syllaber.
- **Estadísticas:** En esta pantalla se mostrarán las estadísticas del usuario.



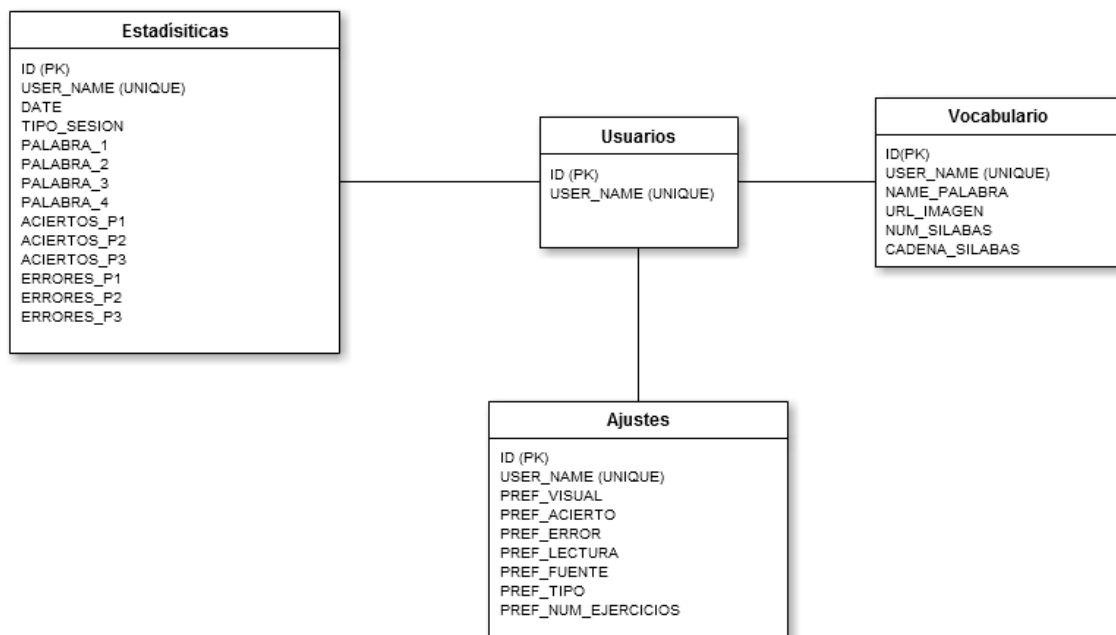
**Figura 12: Diagrama de actividad pantallas.**

En el diagrama de actividad las pantallas que muestran el icono de la base de datos, cilindro, son las que tienen conexión ya sea para extraer o para actualizar información de esta. Las dos nubes son las APIs que se utilizan en el sistema para facilitar la implementación.

### 3.2.5 Diseño de la base de datos

La base de datos se guarda a nivel local del dispositivo. Esta contiene una tabla usuarios, donde se guardan todos los usuarios que se añaden al sistema. El nombre de usuario será único. Cada uno de los usuarios guardará sus propias estadísticas, vocabulario y ajustes en sus respectivas tablas, que se pueden ver en el diagrama de la **Figura 13**.

Se crearán unas tablas auxiliares con los datos guardados comunes para todos los usuarios del dispositivo, en concreto: usuario activo, palabras partida y evaluación. Estas nos permitirán facilitar la implementación. La tabla usuario activo guardará el último usuario activo en la aplicación cuando esta se cierra, al iniciarla de nuevo comenzará con este usuario y todos sus ajustes. Para asignar un id de evaluación al dispositivo la primera vez que se inicie la aplicación se creará la tabla evaluación. En cuanto a la tabla palabras partida servirá para facilitar el traspaso de las palabras una vez iniciado el juego de una sesión a la siguiente, o a la hora de reiniciar el juego.



**Figura 13: Diagrama de la base de datos.**

## 4 Implementación

---

Después de concluir la fase de análisis del sistema y teniendo claras las decisiones tomadas en la etapa de diseño, se puede comenzar a realizar la fase de implementación del sistema. En esta sección se describirá la organización seguida en la implementación del proyecto, los diferentes módulos, las librerías y APIs utilizadas para el desarrollo del sistema, así como las partes del sistema que se consideran más relevantes.

### 4.1 Librerías utilizadas

En relación al cumplimiento de los requisitos marcados en el proyecto, se han utilizado diferentes librerías que han facilitado la implementación que se pueden ver descritas en este apartado.

#### 4.1.1 Volley

Los pictogramas usados en el las diferentes sesiones de juegos son los proporcionados por la API de ARASAAC. Para obtener estos pictogramas se tuvo que programar las peticiones Http a esta API con ayuda de la librería Volley, desarrollada por Google. La conexión con Volley se implementó creando una cola y añadiendo posteriormente peticiones a esa cola, en la **Figura 14** se puede ver la implementación de VolleySingleton.java, destacar el objeto RequestQueue, el cual es una cola donde añaden las conexiones que queremos que realice la librería.

```
public class VolleySingleton {

    private static VolleySingleton mVolleyS = null;
    //Este objeto es la cola que usará la aplicación
    private RequestQueue mRequestQueue;

    private VolleySingleton(Context context) { mRequestQueue = Volley.newRequestQueue(context); }

    public static VolleySingleton getInstance(Context context) {
        if (mVolleyS == null) {
            mVolleyS = new VolleySingleton(context);
        }
        return mVolleyS;
    }

    public RequestQueue getRequestQueue() { return mRequestQueue; }
}
```

**Figura 14:** Código clase VolleySingleton.java.

Para integrar el VolleySingleton.java se implementó un fragmento base el cual extenderemos en los fragmentos donde vayamos a hacer la conexión a internet. Este fragmento base es InterfazServidorActivity.java, en la **Figura 15**, se muestra el método onCreate() del fragmento donde se instancia el Singleton y se inicializa la RequestQueue, a su vez se puede ver el método addToQueue(Request request) encargado de añadir las peticiones a la cola de manera elegante.

```

public class InterfazServidorActivity extends Fragment {

    protected RequestQueue fRequestQueue;
    private VolleySingleton volley;

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        volley = VolleySingleton.getInstance(getActivity().getApplicationContext());
        fRequestQueue= volley.getRequestQueue();
    }

    public void addToQueue(Request request) {
        if (request != null) {
            request.setTag(this);
            if (fRequestQueue == null)
                fRequestQueue = volley.getRequestQueue();
            request.setRetryPolicy(new DefaultRetryPolicy(
                60000, 3, DefaultRetryPolicy.DEFAULT_BACKOFF_MULT
            ));
            onStartConnection();
            fRequestQueue.add(request);
        }
    }
}

```

Figura 15: Código de la clase InterfazServidorActivity.java.

### 4.1.2 Picasso

Los pictogramas que obtenemos de la API de ARASAAC, se guardan en la base de datos, se almacena la URL de la imagen. A la hora de usar estos pictogramas se dio importancia al tema de guardar las imágenes en la memoria interna de la tableta, es decir, creando una caché, debido a que se requiere que el usuario pueda usar la aplicación sin necesidad de internet, este solo sea necesario para gestionar el vocabulario. Para facilitar esta implementación se usó la librería de gestión de imágenes **Picasso** creada por Square Up, la cual se encarga de almacenar y recuperar de la memoria interna así como del proceso de descargar o de eliminar los pictogramas cuando no se utilizan más. En la siguiente **Figura 16** se puede ver con la simplicidad que se realiza la carga de la imagen usando el método:

- *Picasso.with("Contexto").load("URL").into(imageView).*

```

imagen_izq = (ImageView) findViewById(R.id.cuadro_izq);
Picasso.with(imagen_izq.getContext()).load(palabras_juego.get(0).getUrlImage()).into(imagen_izq);

```

Figura 16: Código carga de imagen a través de la librería Picasso.



### 4.1.3 Mail

En cuanto al desarrollo del proceso de evaluación de los usuarios, se implementó un sistema para enviar los logs de las acciones que se consideran más relevantes vía email en segundo plano cuando el usuario utilizara la beta que se les suministró, siempre y cuando aceptara las condiciones de privacidad al iniciar la aplicación. Para ello se usó la librería **Mail** desarrollada por SUN, la cual se encarga del proceso de envío de email desde la aplicación. Se implementó en la clase Mail.java.

## 4.2 Esquema de la implementación

La implementación llevada a cabo está dividida de forma modular para facilitar la inclusión de nueva funcionalidad, interfaces, implementaciones etc. en el futuro, así como para simplificar la etapa de mantenimiento del sistema. Se ha dividido este apartado en lógica del sistema e interfaz.

### 4.2.1 Lógica del sistema

La lógica está dividida en diferentes módulos teniendo en cuenta la funcionalidad que realizan dentro del sistema. Se va a explicar su implementación en este apartado.

#### Gestión de Usuarios

La gestión de usuarios se implementa cumpliendo los requisitos funcionales **RF(01)** referentes a añadir, eliminar, cambiar usuario, así como de los ajustes de usuario y sus estadísticas. Las clases que se dedican de estas funciones son:

- UsuariosActivity.java
- UsuariosElimActivity.java
- AjustesActivity.java
- NumPartidas.java
- SesionActivity.java

UsuariosActivity.java es la clase que realiza la funcionalidad de cambiar de usuario **RF(12)**, se implementó como una lista de los usuarios registrados en la base de datos, la cual se muestra en forma de diálogo al pulsar el botón de cambio de usuario ya sea del menú de ajustes o del menú inicial, gracias al estilo de Android "@android:style/Theme.Dialog".

Al pulsar un elemento de la lista, en este caso un usuario, mostrará un diálogo de alerta preguntado si se estaba seguro cambiar de usuario con los botones de sí y no.

La clase `UsuariosElimActivity.java` y `NumPartidas.java` están implementadas de la misma forma que `UsuariosActivity.java` pero en vez de cambiar de usuario `UsuariosElimActivity` al pulsar el elemento de la lista muestra un diálogo preguntando si se quiere eliminar el elemento o no, en cualquiera de los casos se actualiza la información en la base de datos cumpliendo el requisito **RF(08)**. En el caso de `NumPartidas.java` esta clase muestra una lista con el número de rondas por sesión, el rango es desde 10 hasta 25 rondas, al pulsar sobre el número de rondas escogido se preguntará mediante un diálogo de alerta si se quiere cambiar el número de rondas por sesión esta funcionalidad cumple el requisito **RF(11)**.

La clase que se encarga de mostrar las estadísticas de usuarios es `SesionActivity.java`, la cual muestra una tabla con las sesiones realizadas por el usuario, con la fecha, número de aciertos y fallos de cada palabra jugada, satisfaciendo el requisito **RF(09)**.

Para finalizar el módulo de gestión de usuarios tenemos la clase `AjustesActivity.java`, esta podría haberse situado en el módulo de menús, ya que se puede considerar un menú de ajustes, pero debido a que desde esta clase se cambian las preferencias de usuario de la base de datos se decidió introducirla en el módulo de gestión de usuarios. Esta pantalla implementa la lógica de los botones para cambiar las preferencias de tipo de letra mayúscula o minúscula, fuente de la letra, refuerzo ante acierto, refuerzo ante error, lectura de palabras, añadir usuarios. Todas funcionan de la misma manera al pulsar sobre el botón muestra un cuadro de dialogo con las opciones disponibles en forma de botones, por ejemplo en el caso de tipo de letra mostrará si se quiere cambiar entre mayúsculas o minúsculas. En el caso de añadir usuario el diálogo añadirá un cuadro de texto para escribir el nombre de usuario a registrar en la base de datos.

A su vez la clase `AjustesActivity` enlaza por un sistema de botones a las clases descritas en esta sección:

- `UsuariosActivity.java`,
- `UsuariosElimActivity.java`,
- `NumPartidas.java`,
- `SesionActivity.java`

Así como a las clases que se describirán en el módulo **Gestión de vocabulario**, `VocabularioActivity.java` y `ListaVocabularioActivity.java`.

Con la implementación de `AjustesActivity.java` conseguiremos cumplir los requisitos funcionales **RF(02)**.

## Gestión de Conexión

El módulo de gestión de conexión se encarga de establecer el enlace con las APIs. Está compuesto por las clases VolleySingleton.java e InterfazServidor.java, las cuales se explicaron en el apartado 4.1.1 que describe la librería Volley, así como de MainFragment.java. Esta clase extiende a InterfazServidor.java y es la encargada de obtener las palabras y pictogramas en formato JSON de la API de ARASAAC, así como de recuperar la cadena de la palabra silabeada de la API de Syllaber.

Su método principal es makeRequest() que se puede ver en la **Figura 17**, el cual realiza la petición al servicio web mediante JsonObjectRequest y recibe la respuesta mediante el método onResponse().

```
private void makeRequest(){
    AssistLog.addMessage(AssistLog.LEVEL.INFO, demo_id,
        AssistLog.ACTION_TYPE.CONFIG, "SEARCH", "Busca la palabra" + label.getText().toString());

    buscar= label.getText().toString();
    url=getUrl();
    items=new ArrayList<Palabra> ();

    JsonObjectRequest request= new JsonObjectRequest(url, new Response.Listener<JSONObject>() {
        @Override
        public void onResponse(JSONObject response) {

            JSONArray palabras = null;
            try {
                palabras = response.getJSONArray("symbols");
            } catch (JSONException e) {
                e.printStackTrace();
            }
            for (int i = 0; i < palabras.length(); i++) {

                JSONObject jsonObject = null;
                try {
                    jsonObject = palabras.getJSONObject(i);
                } catch (JSONException e) {
                    e.printStackTrace();
                }

                try {
                    word = new Palabra(jsonObject);
                    items.add(word);
                } catch (JSONException e) {
                    e.printStackTrace();
                }
            }
        }
    });

    ItemVocabularioAdapter adapter = new ItemVocabularioAdapter(getActivity() , items, 0);
    list.setAdapter(adapter);
}
```

**Figura 17:** Código del método makeRequest de la clase MainFragment.java.

La respuesta JSON obtenida de la API de ARASAAC, **Figura 18**, está compuesta por un array de símbolos que recuperamos con `response.getJSONArray("symbols")`, el cual recorreremos para obtener los objetos `JSONObject`, usando el método `getJSONObject(i)`. Para crear una nueva Palabra usamos el constructor de la clase Palabra, el cual recibe el objeto `JSONObject`, esta clase es la encargada de parsear el objeto y almacenarlo en la base de datos, la cual se explica en el apartado de **Gestión de vocabulario**.

Destacar que el sistema detecta el idioma del dispositivo mediante `getURL`, permitiendo recuperar las palabras de ARASAAC en los veinte idiomas disponibles en la API.

```
{ "symbols": [ { "imagePNGURL": "http://arasaac.org/repositorio/originales/7202.png", "name": "perro", "wordTYPE": "2", "CreationDate": "2009-01-19 13:23:19", "ModificationDate": "2009-01-19 13:23:19", "thumbnailURL": "http://arasaac.org/classes/img/thumbnail.php?i=c216ZT01MCZydXRhPS4uLy4uL3JlcG9zaXRvcmlvL29yaWdpbmFsZXMvNzIwMi5wbmc=" }, { "imagePNGURL": "http://arasaac.org/repositorio/originales/2517.png", "name": "perro", "wordTYPE": "2", "CreationDate": "2007-12-17 09:56:09", "ModificationDate": "2008-07-29 09:22:35", "thumbnailURL": "http://arasaac.org/classes/img/thumbnail.php?i=c216ZT01MCZydXRhPS4uLy4uL3JlcG9zaXRvcmlvL29yaWdpbmFsZXMvMjUxNy5wbmc=" } ], "itemCount": 100, "page": 0, "totalItemCount": 2, "pageCount": 0 }
```

**Figura 18: Respuesta JSON obtenida de la API de ARASAAC.**

El JSON con la palabra silabeada se obtiene de la API Syllaber, **Figura 19**, se implementó mediante el método `apiSilabeo`, **Figura 20**, el cual realiza la petición al servicio web mediante `JsonObjectRequest` y recibe la respuesta mediante el método `onResponse`, al igual que `makeRequest`. Del JSON obtenido solo nos interesa el array de sílabas que extraemos con el método `response.getJSONArray("syllables")`, este se convierte en una cadena del formato necesario para almacenar en la base de datos, mediante el método `convertArrayToString(cadena)` de la clase Palabra.

```
{ "word": "tomate", "syllables": ["to", "ma", "te"], "stressed": "te", "stressedType": 1, "stressedLetter": -1, "numSyllables": 3, "hasTl": "false", "hasPrefix": "false" }
```

**Figura 19: Respuesta JSON obtenida de la API de Syllaber.**

```
private void apiSilabeo(String palabra) {

    String url_silabeo="http://syllabler.fwok.org/?json=1&word=" + palabra;

    JsonObjectRequest request= new JsonObjectRequest(url_silabeo, new Response.Listener<JSONObject>() {
        @Override
        public void onResponse(JSONObject response) {

            JSONArray silabas = null;

            try {
                silabas = response.getJSONArray("syllables");
            } catch (JSONException e) {
                e.printStackTrace();
            }

            cadena_silabeo=word.convertArrayToString(silabas);
            numsilabas=silabas.length();

        }
    }, (volleyError) -> { onConnectionFailed(volleyError.toString()); });
    addToQueue(request);
}
```

**Figura 20: Código del método `apiSilabeo` de la clase `MainFragment.java`.**

Este módulo junto al de **Gestión de vocabulario**, nos permite cumplir el requisito funcional **RF(10)**.

## Menús

El módulo de menús está compuesto por las clases MenuInicialActivity.java, ScreenSlidePageFragment, MyFragmentPageAdapter y ListaVocabularioActivity.

La clase MenuInicial.java implementa la funcionalidad del menú inicial, está compuesta por tres botones: el botón jugar que enlaza con la clase de selección de vocabulario ListaVocabularioActivity.java, que se explicará en este apartado, el botón de cambio de usuario que te muestra en forma de diálogo la clase UsuariosActivity.java de gestión de usuarios y el botón de ajustes el cual enlaza con la clase AjustesActivity.java, las cuales se explicaron en el apartado de **Gestión de Usuarios**. Antes de enlazar con AjustesActivity.java se muestra un diálogo con una pregunta de seguridad de manera aleatoria, de esta forma, se evita que los usuarios no avanzados puedan acceder al menú de configuración.

Los tres botones tienen una programación similar, se puede ver en la **Figura 21**, en la cual se muestra la funcionalidad del botón de selección de usuario. Se comprueba que haya un usuario activo, si no lo hay, es decir, db.getUserActivo() devuelve null, muestra el diálogo añadir usuario para registrar el primer usuario en el sistema, una vez añadido, inicia la clase de selección de usuarios, UsuariosActivity.java .

```

selec_usu.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View arg0) {

        AssistLog.addMessage(AssistLog.LEVEL.INFO, demo_id,
            AssistLog.ACTION_TYPE.INTERACTION, "TAP", "Pulsa botón selecciona usuario");

        if (usuario==null) {

            AssistLog.addMessage(AssistLog.LEVEL.INFO, demo_id,
                AssistLog.ACTION_TYPE.GENERAL, "SHOW", "Muestra dialogo añadir primer usuario");

            Dialog añadir = DialogoAñadirUsuario();
            añadir.show();
            db.open();
            usuario=db.getUserActivo();
            db.close();
            if(usuario!=null){
                selec_usu.setText(usuario);
            }

        } else {
            AssistLog.addMessage(AssistLog.LEVEL.INFO, demo_id,
                AssistLog.ACTION_TYPE.GENERAL, "START", "Inicia actividad de lista de usuarios [UsuariosActivity]");

            Intent intent = new Intent(MenuInicialActivity.this, UsuariosActivity.class);
            startActivity(intent);
        }
    }
}

```

**Figura 21: Código del botón de cambiar de usuario de la clase MenuInicial.java.**

Otra parte a destacar del código de `MenuInicialActivity.java` es cómo se programó la pregunta de seguridad, en las **Figura 22** y **Figura 23** se observa parte del método `crearCaptcha()` el cual devuelve un diálogo, en la **Figura 22** se ve como se elige la pregunta de manera aleatoria con la función `Math.floor` y un `switch`. En cada `switch` se establece la pregunta a mostrar y la respuesta correcta. En la **Figura 23** se muestra cómo se comprueba que el `textoBusqueda` sea igual a la respuesta, si es igual la respuesta es correcta por lo tanto se enlazará con la pantalla de ajustes, en caso de no ser la respuesta correcta, se mostrará un diálogo de error de captcha.

```
private Dialog crearCaptcha(){
    AlertDialog.Builder builder = new AlertDialog.Builder(this);

    final EditText textoBusqueda = new EditText(this);
    String respuesta="";
    //Pregunta aleatoria
    int valorEntero = (int) Math.floor(Math.random()*(6-1+1)+1);

    switch(valorEntero){

        case 1:
            builder.setTitle("¿Cuál es el adjetivo en la siguiente frase?");
            builder.setMessage("Nacho tiene un coche verde.");
            builder.setView(textoBusqueda);
            respuesta="verde";

            break;
```

**Figura 22:** Parte 1 de código para crear captcha de la clase `MenuInicial.java`.

```
final String finalRespuesta = respuesta;

builder.setPositiveButton("Aceptar", (dialog, whichButton) -> {

    AssistLog.addMessage(AssistLog.LEVEL.INFO, demo_id,
        AssistLog.ACTION_TYPE.INTERACTION, "TAP", "Pulsa botón aceptar dialogo captcha");

    if (textoBusqueda.getText().toString().equals(finalRespuesta)) {

        AssistLog.addMessage(AssistLog.LEVEL.INFO, demo_id,
            AssistLog.ACTION_TYPE.GENERAL, "INTENT", "Inicia actividad de ajustes [AjustesActivity]");

        Intent intent2 = new Intent(MenuInicialActivity.this, AjustesActivity.class);
        startActivity(intent2);
        startActivity(intent2);
    } else {

        AssistLog.addMessage(AssistLog.LEVEL.ERROR, demo_id,
            AssistLog.ACTION_TYPE.GENERAL, "ERROR_CAPTCHA", "Error captcha");
        Toast.makeText(MenuInicialActivity.this, "Respuesta incorrecta.", Toast.LENGTH_SHORT).show();
    }

});
```

**Figura 23:** Parte 2 de código para crear captcha de la clase `MenuInicial.java`.

Las clases `ListaVocabularioActivity.java` junto a `ScreenSilidePageFragment.java` y `MyFragmentPageAdapter.java`, son las encargadas de implementar el menú de selección de fase y sesión, con el que se cumple el requisito **RF(14)**. Las fases se dividen en tres pantallas. Se puede pasar de una a otra realizando la acción de deslizar el dedo por la pantalla. En cada una de las pantallas se muestran los botones que al pulsarlos iniciarán la sesión correspondiente.

### Lógica del juego

En el módulo de lógica nos encontramos con la funcionalidad del juego de aprendizaje. Esta se implementa en diferentes clases una por cada nivel de juego disponible en el sistema.

En la primera fase asociación imagen-palabra/palabra tenemos cuatro clases:

- `PrimeraFaseActivity.java`
- `PrimeraFaseApProfActivity.java`
- `PrimeraFaseDiscActivity.java`
- `PrimeraFaseDiscInvActivity.java`

La segunda fase asociación imagen/palabra está programada en las clases:

- `SegundaFaseActivity.java`
- `SegundaFaseApProfActivity.java`
- `SegundaFaseDiscActivity.java`
- `SegundaFaseDiscInvActivity.java`

La tercera y última fase actualmente solo dispone de un nivel el cual se implementó en la clase `TerceraFaseActivity.java`.

Aunque cada nivel sigue una lógica diferente la base de la programación es similar por ello vamos a describir las partes más relevantes de código de manera general a todas.

Para inicializar la pantalla lo hacemos dentro del método `onCreate()`, en el cual inicializamos los listener como se puede ver en la **Figura 24**. En los métodos `setOnDragListener()` y `setOnLongClickListener()` se implementa la funcionalidad de arrastrar el cuadro que contiene el rótulo palabra al contenedor correspondiente, cumpliendo el requisito **RF(15)**.

```
findViewById(R.id.cuadro_wd).setOnLongClickListener(this);
    register drag event listeners for the target layout containers
findViewById(R.id.iz_cont_center_bottom).setOnDragListener(this);
findViewById(R.id.center_cont_center_bottom).setOnDragListener(this);
findViewById(R.id.der_cont_center_bottom).setOnDragListener(this);
```

**Figura 24:** Código inicialización de los listener de los contenedores y del rótulo.

Cuando se realiza una pulsación larga sobre el cuadro que contiene el rótulo con la palabra, se empieza a arrastrar dicho cuadro, esto se implementó en el método `onLongClick()`, que se puede ver en la **Figura 25**, primero se crea una sombra de la vista con el método `View.DragShadowBuilder` y se empieza el arrastrado mediante el método `StartDrag`. La `imageView` inicial del cuadro se pone invisible.

```

@Override
public boolean onLongClick(View imageView) {

    AssistLog.addMessage(AssistLog.LEVEL.INFO, demo_id,
        AssistLog.ACTION_TYPE.INTERACTION, "LONG_CLICK", "Cuadro palabra");

    //Se crea clip, el cuadro es tocado
    ClipData clipData = ClipData.newPlainText("", "");

    //empieza drag, se crea la sombra
    View.DragShadowBuilder shadowBuilder = new View.DragShadowBuilder(imageView);
    imageView.startDrag(clipData, shadowBuilder, imageView, 0);
    //cuadro invisible
    imageView.setVisibility(View.INVISIBLE);
    return true;
}

```

**Figura 25:** Código método `onLongClick` común a todas las sesiones de juego.

Una vez creada la sombra y empezado el arrastrado se llama al método `onDrag`, este se implementó mediante un `switch` donde se recibe el `DragEvent` y diferentes `case` para comprobar cuando empieza el arrastre, cuando finaliza etc. Nos vamos a centrar en el `case DragEvent.ActionDrop`, ya que nos interesa saber dónde se suelta el cuadro, lo podemos ver en la **Figura 26**, dentro del `case` nos encontramos otro `switch`, en este caso recibe la `id` del `layout` donde hemos soltado el cuadro, el `case` que se muestra en la figura X es el correspondiente al contenedor de la parte izquierda de la pantalla, este pantalla se divide en tres contenedores izquierda, centro y derecha, uno por cada palabra de la sesión. En este `case` se comprueba que la palabra y pictograma del contenedor izquierdo sea igual al cuadro que contiene el rótulo de la palabra mediante la siguiente sentencia del código `IF ( wd_izq.getText().equals(wd_cuadro.getText())`), en caso de que se cumpla la condición, se actualiza la barra de estado de la sesión y el número de aciertos, se muestra una animación de un tick verde, en modo de refuerzo acierto, mediante la llamada al método `move()`, se reproduce un sonido de acierto, en caso de que el usuario tenga activada la opción sonido ante acierto, con el método `reproducirSonido(id_acierto)` y se cambia a la siguiente ronda de la sesión mediante el método `sesionDriscriminacion(progressJuego)`. En el caso de no que no se cumpla la condición se hace lo contrario, se actualiza el número de errores, se devuelve el cuadro de palabra a su posición inicial, en caso de tener activado el sonido ante error se reproduce con el método de `reproducirSonido(id_error)` y si se tienen activado las ayudas visuales, en caso de ser el primer fallo de la ronda se producirá un flash en la pantalla el cual se realiza mediante una animación `startAnimation(set)` jugando con el alfa del `imageView` del contenedor, en caso de ser el segundo fallo o más se dejará el contenedor donde se falló en color gris.



```

case DragEvent.ACTION_DROP:

    switch (receivingLayoutView.getId()) {

        case R.id.iz_cont_center_bottom:

            draggedImageViewParentLayout
                = (ViewGroup) draggedImageView.getParent();
            draggedImageViewParentLayout.removeView(draggedImageView);
            bottomLinearLayout = (LinearLayout) receivingLayoutView;
            bottomLinearLayout.addView(draggedImageView);
            draggedImageView.setVisibility(View.VISIBLE);

            if(wd_izq.getText().equals(wd_cuadro.getText())){

                reproducirSonido(id_acierto);
                progressStatus++;
                progressJuego++;
                progressBar.setProgress(progressStatus);
                aciertosSesion++;
                acierto_p0++;
                num_fallos = 0;
                move();

                Thread thread = run() -> {
                    try {
                        synchronized (this) {
                            wait(1000);

                            runOnUiThread(() -> {

                                bottomLinearLayout.removeView(draggedImageView);
                                draggedImageViewParentLayout.addView(draggedImageView);
                                sesionDiscriminacion(progressJuego);

                            });
                        }
                    } catch (InterruptedException e) {
                        e.printStackTrace();
                    }
                };
            }
            thread.start();
        }else{
            reproducirSonido(id_error);
            erroresSesion++;
            error_p0++;
            num_fallos++;
            bottomLinearLayout.removeView(draggedImageView);
            draggedImageViewParentLayout.addView(draggedImageView);
            db.open();
            if (db.getUserAyudaVisual(db.getUserActivo()).equals("1")) {
                if(num_fallos==1) {
                    findViewById(R.id.top_container).startAnimation(set);
                }else{
                    findViewById(R.id.izq_container).setAlpha(0.3f);
                }
            }
            db.close();
            sesionDiscriminacion(progressJuego);
        }
    }
}

```

Figura 26: Parte de Código del método OnDrag, case correspondiente a soltar el rótulo.

Otra parte importante del código de las fases de juego es cargarPantallaInicial(), la cual se encarga de cargar las palabras y sus pictogramas, así como representar la pantalla con los ajustes de usuario almacenados en la base de datos, en la **Figura 27** se puede ver cómo comprueba el tipo de fuente del usuario, así lo hace con todos los ajustes.

```

if(db.getUserAyudaFuente(db.getUserActivo()).equals("1")) {

    AssistLog.addMessage(AssistLog.LEVEL.INFO, demo_id,
        AssistLog.ACTION_TYPE.EXECUTION, "FUENTE", "Chalkboard");

    wd_izq.setTypeface(chalkboard);
    wd_cent.setTypeface(chalkboard);
    wd_der.setTypeface(chalkboard);
    wd_cuadro.setTypeface(chalkboard);

}else{

    AssistLog.addMessage(AssistLog.LEVEL.INFO, demo_id,
        AssistLog.ACTION_TYPE.EXECUTION, "FUENTE", "Hélvetica");

    wd_izq.setTypeface(helvetica);
    wd_cent.setTypeface(helvetica);
    wd_der.setTypeface(helvetica);
    wd_cuadro.setTypeface(helvetica);

}

```

**Figura 27:** Parte de código del método cargarPantallaInicial de las sesiones de juego.

En cuanto al cambio de ronda, se realiza con el método que da nombre a la fase, en este caso explicado el método es sesionDiscriminacion(int progress), el cual coloca la pantalla en función de la ronda correspondiente mediante switch-case. En caso de que el número de rondas de ajustes seleccionado sea mayor que el número de rondas disponibles se irá cambiando la pantalla aleatoriamente a partir de ese punto.

Una funcionalidad que se implementó en las diferentes clases de las sesiones es la correspondiente a lectura de palabra, para ello se usó el método speak de la clase incluida en Android TextToSpeech, lo que hace cumplir el requisito funcional **RF(07)**.

Para finalizar las clases correspondientes a las sesiones de juego satisfaciendo el **RF(19)**, se guardan las estadísticas, fecha, palabras, número de aciertos y de errores de la sesión en la base de datos. A su vez se calcula la puntuación obtenida por el usuario **RF(18)** en la partida y se muestra al usuario mediante un diálogo.

El módulo de lógica del juego cumple los requisitos ya mencionados en referencia a ajustes de usuario **RF(02)**, ya que aparte de poder modificar las preferencias estas tienen que ser cumplidas durante el juego.

## Gestión de vocabulario

El módulo gestión de vocabulario incluye las clases encargadas de la funcionalidad correspondiente a los requisitos **RF(10)** **RF(13)** añadir, eliminar o seleccionar vocabulario, estas clases son:

- EliminarVocabularioActivity.java
- VocabularioActivity.java
- SeleccionarVocabulario.java

La clase VocabularioActivity.java se encarga de la búsqueda de vocabulario por medio del fragmento del módulo de gestión de conexión MainFragment.java, que establece la conexión con la API y que se explicó en el apartado de **Gestión de Conexión**.

La clase EliminarVocabulario.java muestra una lista de las palabras almacenadas por el usuario en la base de datos, en caso de querer eliminarla se pulsará sobre la palabra de la lista y se le mostrará un diálogo de confirmación.

La funcionalidad para seleccionar el vocabulario con el que se va a trabajar la sesión se implementó en la clase SeleccionarVocabulario.java. Se programó dos listas una que muestra el vocabulario disponible en la base de datos del correspondiente usuario. Como se puede ver en la **Figura 28**, se devuelve un cursor de las palabras con el método getAllPalabras(db.getUserActivo()) y se crea un adaptador mediante el método new ItemVocabularioAdapter(...), el cual se relaciona con la lista list.setAdapter(adapter). Para seleccionar una palabra de la lista se programó el método onClickListener de la primera lista, cuando se toca un elemento de la primera lista se pasa a la segunda lista, en la que se van añadiendo las palabras seleccionadas y se actualiza su adapter.

```

db.open();
demo_id=db.getIdPruebas();
Cursor c= db.getAllPalabras(db.getUserActivo());

if (c.moveToFirst()) {
    //Recorremos el cursor hasta que no haya más registros
    do {
        String nombre_palabra= c.getString(0);
        String url_imagen = c.getString(1);
        String tipo = c.getString(2);
        int numsilbas= c.getInt(3);
        String cadenasilabas = c.getString(4);
        Palabra palabra=new Palabra(nombre_palabra,url_imagen,tipo,numsilbas,cadenasilabas);
        items.add(palabra);
    } while(c.moveToNext());
}

db.reiniciarPalabraPartida();
db.close();

adapter = new ItemVocabularioAdapter(this , items, 1);
list.setAdapter(adapter);

adapter_selec = new ItemVocabularioAdapter(this , items_selec, 1);
list_selec.setAdapter(adapter_selec);

```

**Figura 28:** Código para extraer las palabras de la base de datos de la clase SeleccionarVocabulario.java

## **Gestión de evaluación**

El proceso de evaluación necesitaba de una funcionalidad correspondiente por ello se decidió implementar este módulo que incluye las clases AgreementDialog, AssitLog y BaseActivity.

AgreementDialog es la encargada de crear un diálogo con las condiciones de uso para la beta que se enviará para realizar la evaluación. AssitLog es la clase encargada de implementar el log de actividad para saber toda la interacción del usuario. BaseActivity simplemente usara los métodos de estas clases, para inicializar el diálogo, el log y enviar el log al cerrar la aplicación.

## **Base de datos**

La base de datos creada sigue el esquema realizado en la etapa de diseño. Se ha implementado usando el sistema de gestión de bases de datos SQLite, la creación y métodos de gestión de estas fueron implementados en la clase DatabaseAdapter.java, del módulo database.

### **4.2.2 Interfaz del Sistema**

La implementación de la interfaz del sistema es la que nos permite posicionar los elementos gráficos en la posición que queremos de la pantalla, es decir, crear cada pantalla del sistema.

Para implementar la interfaz de cada pantalla del sistema se programó en XML los diferentes layouts, los cuales se llaman en el onCreate() de la actividad donde se quieren mostrar.

Las cadenas de texto de la interfaz se incluyeron en String.xml, y se tradujeron al inglés, con lo que se tiene una aplicación completa en español y en inglés, ya que como se contó en **Gestión de Conexión**, el sistema esta implementado para detectar el idioma del teléfono y recuperar en este lenguaje las palabras de ARASAAC. Con esto conseguimos tener una aplicación en múltiples idiomas solo traduciendo el String.xml a los idiomas disponibles en la API.

En el **anexo A** se encuentra el manual de usuario, donde se pueden observar capturas de las diferentes pantallas de la aplicación.

## 5 Evaluación del proyecto

El objetivo de este apartado es validar el diseño, análisis e implementación del proyecto. Para ello, se realizó una evaluación del sistema con un grupo de expertos en el tema de la lectura global para niños con necesidades especiales.

### 5.1 Proceso de evaluación

El proceso de evaluación que se llevó a cabo con usuarios expertos, nos va a permitir validar nuestro sistema. Para realizar la evaluación se preparó el sistema, es decir se creó una beta, la cual nos permitiera obtener todos los datos de interacción sin necesidad de estar presente. Por ello se creó el módulo de **Gestión de evaluación** en el sistema, el cual añade al sistema un diálogo de condiciones de uso, un id de usuario de evaluación y log de todo lo realizado por el usuario que se envía por correo de manera invisible.

Para la realización del proceso de evaluación se envió un email a un grupo de expertos en la materia, explicándoles en qué consistía el proyecto y ofreciéndoles que se apuntaran al proceso de evaluación del sistema. A los expertos que contestaron positivamente al email se les añadió a la beta privada la cual esta subida en Google Play y se les envió un segundo email con el proceso de evaluación que tenían que realizar, con los plazos del proceso y un manual de instalación y uso del sistema que se puede ver en el **anexo A**.

En el plazo fijado se les proporcionó un cuestionario de evaluación de la aplicación realizado en Google Forms, el cual se puede ver en el **anexo B**. Este cuestionario está dividido en tres partes: datos de usuario, cuestionario estándar de usabilidad System Usability Scale (SUS) y preguntas sobre la experiencia con la aplicación.

La parte de datos de usuario pregunta al evaluador por la relación que tiene con el TEA y por su profesión, con el fin de saber si es padre, profesor o experto en la materia. También se pregunta por el código de usuario que se le asigna a la beta que se le envió, el cual se le proporciona en ajustes de la aplicación, esto se hace con el fin de saber datos que log de interacción corresponde a cada usuario. Los logs de interacción se envían al correo de manera invisible como se describió en el apartado de **Gestión de evaluación**.

Se escogió un cuestionario estándar de usabilidad, System Usability Scale (SUS) [8], el cual se ha traducido al español. Está formado por diez preguntas con cinco opciones de respuesta. Las opciones como se puede ver en la **Figura 29**, van desde totalmente en desacuerdo hasta totalmente de acuerdo. Si el usuario no se ve capaz de contestar una pregunta se señalará el valor central de la escala 3.

Strongly Disagree 1	2	3	4	Strongly Agree 5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

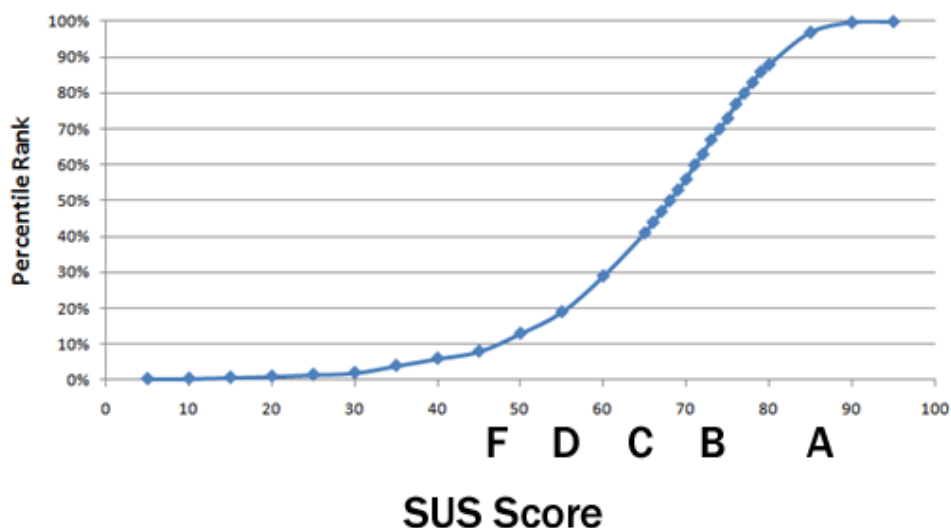
Figura 29: Opciones respuesta cuestionario SUS.

Para interpretar los datos del SUS hay que seguir los siguientes pasos:

- En las preguntas impares: restar uno del valor de la respuesta del usuario.
- En las preguntas pares: restar a 5 el valor de la respuesta del usuario.
- Esta escala da valores entre 0-4, siendo 4 la respuesta más positiva.
- Sumar las respuestas convertidas mediante los puntos 1 y 2. Multiplicar el total por 2.5, con lo que conviertes el rango de valores posibles 0-100 y no de 0-40.

Según diversos estudios la media de los SUS realizados es de 68, por tanto una puntuación superior a 68 sería mayor que la media.

Para interpretar el valor obtenido la mejor forma es convertirlo en un rango percentil, el proceso se llama normalización. En el **Figura 30**, se puede observar el gráfico del rango percentil asociados al de las puntuaciones SUS y a las letras para clasificar el sistema. Por ejemplo, si obtuviéramos un valor de 74, se convertiría en rango percentil del 70%, lo que querría decir que se ha obtenido mejor puntuación de uso que el 70% de los sistemas probados con este cuestionario, esto se interpreta con clasificación del sistema B-. Una puntuación superior a 80.3 obtendría la clasificación más alta la A, una puntuación de 68 recibe una C y una de 51 recibe una F.



**Figura 30:** Gráfico rangos del percentil asociados a las puntuaciones SUS. *Fuente:* [9]

Para finalizar el cuestionario de evaluación, se realizaron preguntas abiertas sobre la experiencia con la aplicación que se pueden ver en el **anexo B**, con lo que se pretende evaluar la facilidad de uso y la interfaz del sistema.

Para concluir el proceso de evaluación estamos a la espera de recibir las respuestas del grupo de expertos a los cuestionarios. Se analizarán los resultados obtenidos en las próximas semanas como se detalló en el apartado **6.2**.

## **6 Conclusiones y trabajo futuro**

---

### **6.1 Conclusiones**

Este proyecto tenía como objetivos proporcionar los materiales utilizados en el proyecto “Leo con Lula” para pizarra digital de manera estructurada y ordenada para facilitar el aprendizaje. En segundo lugar obtener información de la progresión de los usuarios por parte de los educadores. Estos objetivos principales se han cumplido, así como los objetivos presentados en el apartado 1.2.

Para cumplir los objetivos propuestos, se realizó un análisis centrado en el usuario experto. En el que mediante reuniones se extrajeron todos los requisitos necesarios para que el sistema fuera útil para el usuario final. A continuación del análisis, se realizaron las etapas de diseño e implementación, se diseñó la arquitectura del sistema, la base de datos y los prototipos necesarios que se evaluaron con el usuario experto. Una vez terminada la etapa de diseño, se implementó el sistema de manera modular para facilitar la integración de nuevos módulos en el futuro. Para finalizar, se inició un proceso de evaluación con un grupo de usuarios expertos.

En conclusión, el proyecto se realizó con éxito ya que se consiguió un sistema completo, centrado en los requisitos propuestos por el usuario experto y que cumple la funcionalidad de facilitar el proceso de evaluación al grupo de educadores, así como facilitar el proceso de enseñanza y aprendizaje teniendo el material necesario para llevar a cabo el método de lectura global “Leo con Lula” de manera agrupada.

### **6.2 Trabajo futuro**

En este apartado se describe el trabajo futuro que se realizará para el proyecto. En primer lugar, se terminará con el proceso de evaluación en las próximas semana, analizando los datos obtenidos en el cuestionario SUS que se envió al grupo de expertos, el cual hemos definido en el apartado de evaluación. Estos resultados se esperan publicar a lo largo de 2017 en un foro de impacto del área. A su vez, las opiniones de este grupo de expertos se tendrán en cuenta para futuras mejoras del sistema.

Para finalizar, se llevará un proceso de evaluación centrado en los usuarios finales, es decir, en los alumnos que utilizarán “Leo con Lula”. En este proceso de evaluación se quiere comprobar que esta aplicación sirve como herramienta en el aprendizaje de la lectura por el método global para niños con necesidades especiales en un marco real. Esto se llevará a cabo en los meses sucesivos, ya que se trata de un proceso largo.

## Referencias

---

- [1] Monfort, M., & Juárez, A. (2004). Leer para hablar. *La adquisición del lenguaje escrito en niños con alteraciones del desarrollo y/o del lenguaje*. Madrid: Entha Ediciones.
- [2] Esteban, E. "El autismo y el aprendizaje de los niños. Educación especializada para niños con autismo".
- [3] Roncero, M. R. V. (2003). "La lectoescritura como instrumento de aprendizajes relevantes en autismo"
- [4] Segers, J.E. (1958). "La enseñanza de la lectura por el método global"
- [5] Rodríguez, J. M. H., Pérez, A. M., & Casas, B. R. (2007). *Déjame que te hable de los niños y niñas con autismo de tu escuela*. Teleno.
- [6] Ayala, J. C. R. "Las TICs como herramientas mediadoras en las interacciones sociales de niños con autismo de alto rendimiento en el aula regular".
- [7] García, F., Fonoll, J., & Fernández, J. (2011). Accesibilidad, TIC y educación. *Madrid: Centro Nacional de Información y Comunicación Educativa, Ministerio de Educación y Ciencia*.
- [8] Brooke, J. (1996). SUS-A quick and dirty usability scale. *Usability evaluation in industry*, 189(194), 4-7.
- [9] Sauro, J. (2011). Measuring Usability with the System Usability Scale (SUS). Recuperado de <http://www.measuringu.com/sus.php>



## Glosario

---

API	Application Programming Interface.
ASD	Autism spectrum disorder.
ICT	Information and communications technology.
PDF	Portable Document Format.
RF	Requisito funcional.
RNF	Requisito no funcional.
SUS	System Usability Scale.
TEA	Trastornos del Espectro Autista.
TIC	Tecnología de información y comunicaciones.
XML	eXtensible Markup Language

## Anexos

---

### A. Manual de instalación y uso

#### 1. Instalación

Para instalar “Leo con Lula” se debe acceder a Google Play a través del teléfono o el ordenador, con la cuenta de Gmail que se facilitó al equipo de desarrollo de Leo con Lula y buscar “leo con lula”.

#### 2. Arranque de aplicación

Una vez instalada, podemos arrancarla desde el lanzador de aplicaciones, tocando sobre el siguiente icono:



Una vez lanzada, se nos mostrará un diálogo con las condiciones de uso. Si no se está de acuerdo, basta con pulsar en cancelar para cerrar la aplicación.

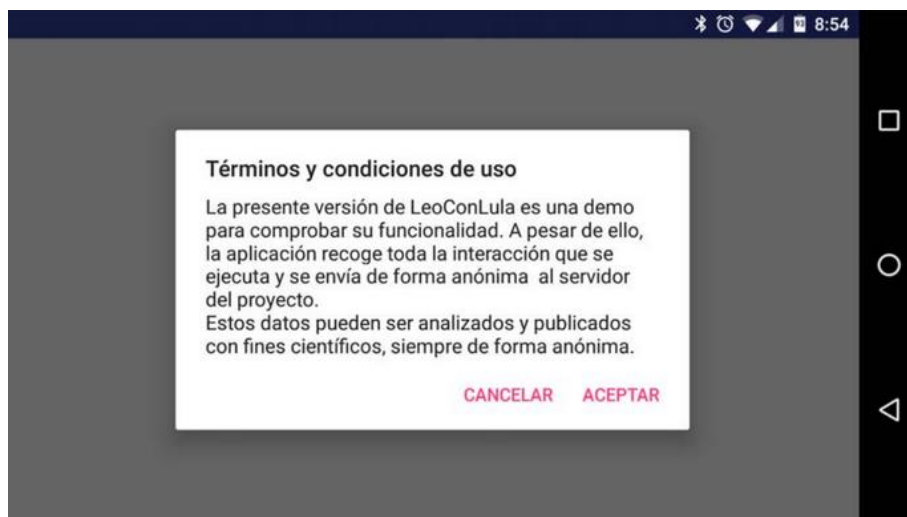


Figura 31: Captura manual uso, diálogo “Leo con Lula” condiciones de uso.

### 3. Pantalla de inicio

La pantalla de inicio cuenta con tres elementos básicos: selección de usuario, comenzar juego y acceder a los ajustes



Figura 32: Captura manual uso, menú inicial.

### 4. Ajustes de la aplicación

La primera vez que accedamos a los ajustes, si no hay ningún usuario creado, se nos preguntará por el nombre del usuario a crear

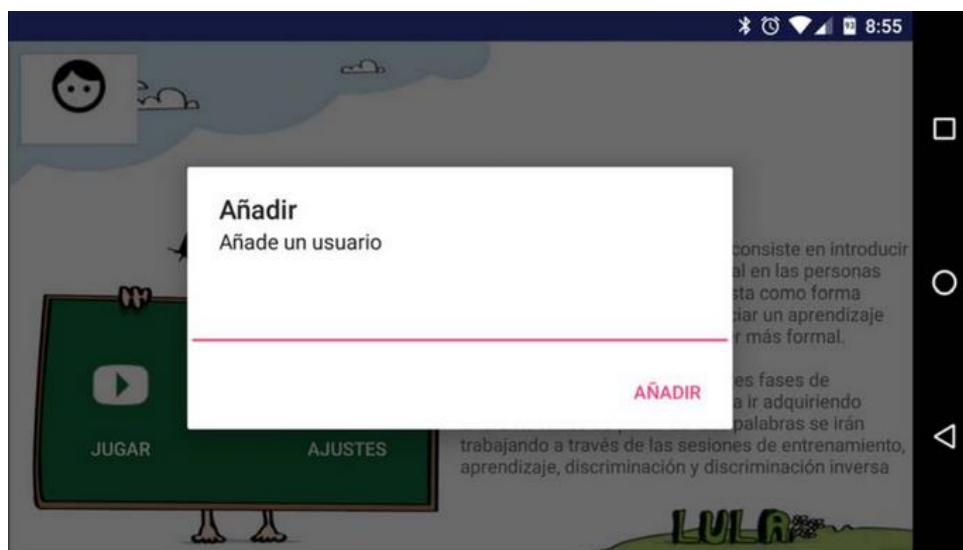


Figura 33: Captura manual uso, añadir usuario.

Una vez lo creamos, se nos mostrará otro diálogo con la pregunta de seguridad. De esta forma, se evita que los usuarios no avanzados puedan acceder al menú de configuración.

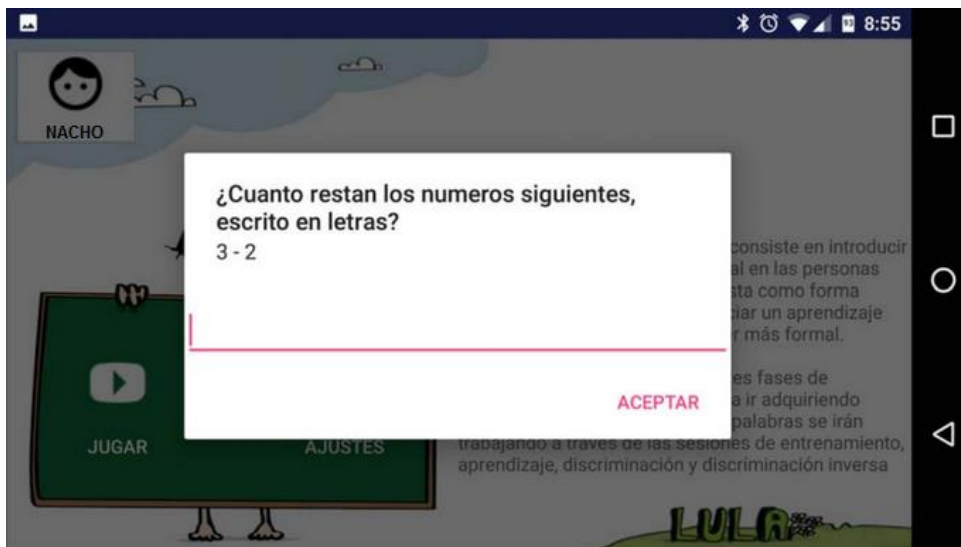


Figura 34: Captura manual uso, diálogo pregunta seguridad.

Una vez dentro de la pantalla de configuración se nos mostrará, en la esquina superior izquierda, el icono del usuario que hemos creado, en la zona central nuestro código de “beta tester” (este dato será importante, para realizar el cuestionario de evaluación) y, por último, la pizarra de opciones.



Figura 35: Captura manual uso, menú ajustes.

Los parámetros de configuración son:

- Tipo de letra: mayúsculas o minúsculas.
- Fuente: helvética o chalkboard.
- Ayudas visuales: sí o no.
- Sonido en caso de éxito: sí o no.
- Sonido en caso de fallo: sí o no.
- Leer palabras al tocar: sí o no.
- Estadísticas: en futuras versiones se podrá acceder a una sección con estadísticas de rendimiento del usuario.
- Gestionar vocabulario: para añadir palabras al conjunto del vocabulario con el que se va a trabajar.
- Añadir y eliminar usuarios.
- Número de juegos: es el número de “turnos” de cada sesión.
- Jugar.

##### 5. Gestión de vocabulario

“Leo con Lula” trabaja con el conjunto de pictogramas de ARASAAC (<http://arasaac.org/>). En la caja superior podemos teclear la palabra a buscar y en la caja inferior se mostrarán los pictogramas que nos ofrece ARASAAC relacionados con esa palabra. Para añadir la palabra al vocabulario basta con tocar sobre el picto.



**Figura 36: Captura manual uso, pantalla gestión vocabulario.**

## 6. Juego

Lo primero que tenemos que hacer para comenzar un juego es seleccionar el vocabulario con el que se va a trabajar. Para ello, se nos mostrará un selector con todo el vocabulario disponible (columna izquierda) y las palabras seleccionadas (columna derecha). Tendremos que seleccionar 3 palabras para empezar a jugar.



**Figura 37: Captura manual uso, pantalla selección palabras juego.**

Existen 3 niveles o tipos de juego:

- Asociación imagen-palabra/palabra.
- Asociación imagen palabra.
- Descomposición silábica.

Cada uno de los niveles se compone de cuatro sesiones:

- Entrenamiento.
- Aprendizaje y profundización.
- Discriminación.
- Discriminación inversa<sup>8</sup>.

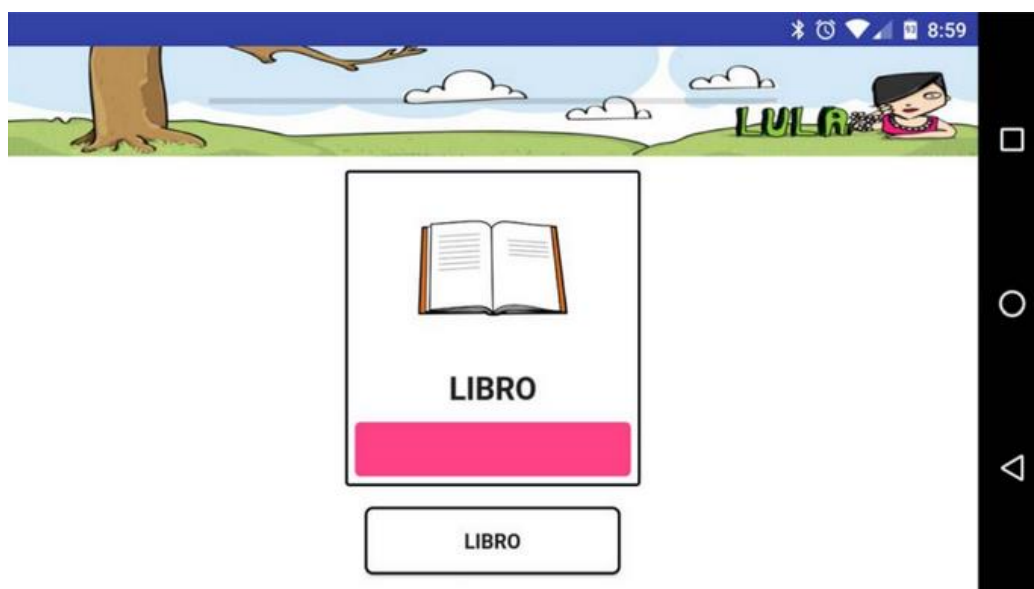
Los niveles se tienen que dar por orden. Y, cada sesión, cuenta con el número de turnos que se haya configurado en el menú de ajustes.

<sup>8</sup> La descomposición silábica solo cuenta con una sesión.



**Figura 38: Captura manual uso, pantalla selección sesión.**

Las siguientes capturas se corresponden con diferentes turnos de las sesiones I y II del primer nivel. Al completar cada sesión se muestra un mensaje con una valoración (en formato de estrellas) y se ofrece la opción de continuar con la siguiente sesión.



**Figura 39: Captura manual uso, turno de la sesión I.**

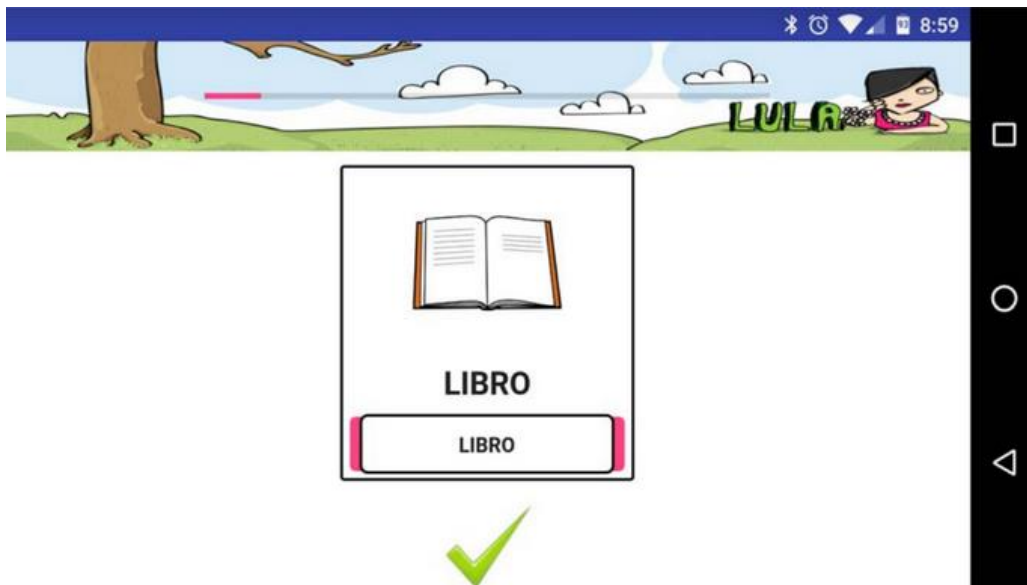


Figura 40: Captura manual uso, refuerzo ante acierto.

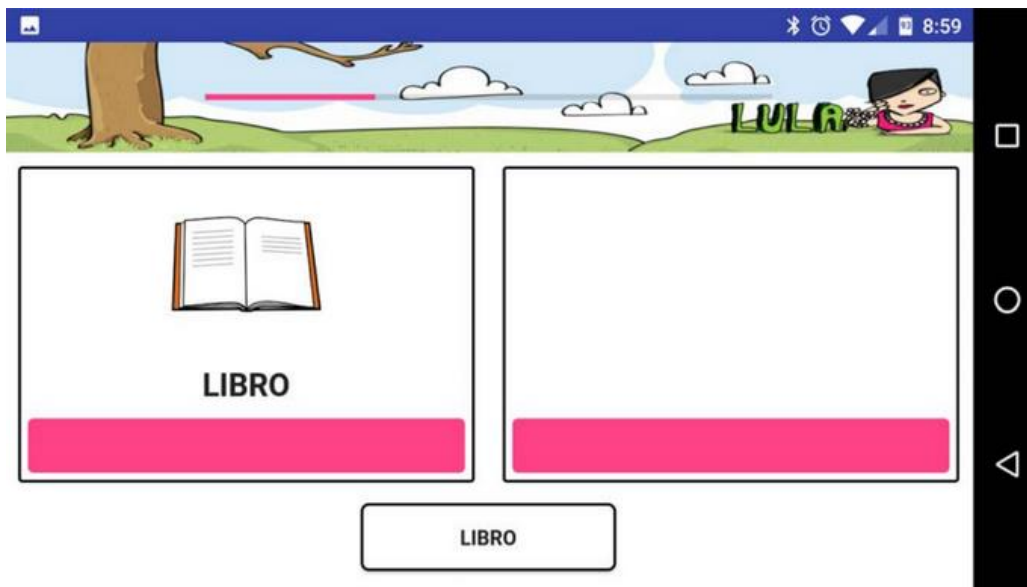


Figura 41: Captura manual uso, asociar sesión I ante contenedor contenido nulo.



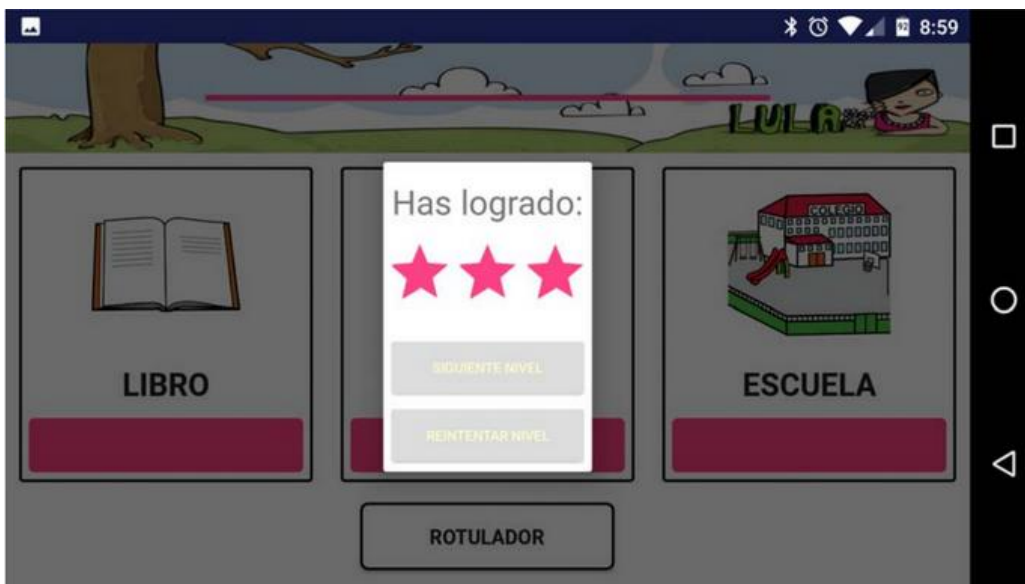


Figura 42 Captura manual uso, puntuación al finalizar sesión.



## B. Cuestionario evaluación “Leo con Lula”.

En este apartado se pueden ver las capturas de pantallas correspondientes al cuestionario de evaluación enviado al grupo de expertos.

### Evaluación Leo Con Lula

El siguiente cuestionario pretende recoger tus impresiones sobre la herramienta Leo con Lula, que has probado durante las últimas semanas. Este feedback nos será de gran utilidad para mejorar el producto

\*Obligatorio

#### Datos del usuario

Nombre

Tu respuesta

Edad

Tu respuesta

¿Cuál es tu relación con el TEA? \*

Tu respuesta

¿Cuál es tu profesión? \*

Tu respuesta

¿Cuál es tu código de usuario? (Lo puedes encontrar en la configuración de la aplicación) \*

Tu respuesta

Figura 43: Captura cuestionario evaluación, datos del usuario.

## Cuestionario de usabilidad

Las siguientes preguntas forman parte de un cuestionario de usabilidad estándar

Creo que yo usaría este sistema \*

	1	2	3	4	5	
En desacuerdo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Totalmente de acuerdo

Encuentro el sistema innecesariamente complejo \*

	1	2	3	4	5	
En desacuerdo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Totalmente de acuerdo

Creo que el sistema fue fácil de usar \*

	1	2	3	4	5	
En desacuerdo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Totalmente de acuerdo

Creo que necesitaría el soporte de un técnico para usar el sistema \*

	1	2	3	4	5	
En desacuerdo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Totalmente de acuerdo

Creo que las numerosas funciones del sistema están bien integradas \*

	1	2	3	4	5	
En desacuerdo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Totalmente de acuerdo

Figura 44: Captura cuestionario evaluación, cuestionario usabilidad parte I.

Creo que hay mucha inconsistencia en el sistema \*

	1	2	3	4	5	
En desacuerdo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Totalmente de acuerdo

Creo que la mayoría de la gente aprendería a usar el sistema rápidamente \*

	1	2	3	4	5	
En desacuerdo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Totalmente de acuerdo

Creo que es muy engorroso usar el sistema \*

	1	2	3	4	5	
En desacuerdo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Totalmente de acuerdo

Me he sentido muy seguro usando el sistema \*

	1	2	3	4	5	
En desacuerdo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Totalmente de acuerdo

Necesito aprender muchas cosas antes de empezar a utilizar el sistema \*

	1	2	3	4	5	
En desacuerdo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Totalmente de acuerdo

Figura 45: Captura cuestionario evaluación, cuestionario usabilidad parte II.

## Preguntas sobre la experiencia de uso de Leo con Lula

¿Qué es lo que más te ha gustado?

Tu respuesta

---

¿Y lo que menos?

Tu respuesta

---

Desde el punto de vista de la persona de apoyo, ¿qué te ha parecido lo más fácil?

Tu respuesta

---

¿Y lo más difícil?

Tu respuesta

---

Finalmente, en este espacio nos puedes dejar cualquier tipo de idea u opinión que te parezca relevante

Tu respuesta

---

ENVIAR

Nunca envíes contraseñas a través de Formularios de Google.

**Figura 46:** Captura cuestionario evaluación, preguntas sobre la experiencia de uso de Leo con Lula.