

CIS1730CP06

LSC APP: APLICACIÓN MÓVIL PARA LA PRÁCTICA DE LA LENGUA DE
SEÑAS COLOMBIANA.

Marvin Daniel Cely Báez

Sergio Forero Gómez

José Darío Guerrero Aragón

PONTIFICIA UNIVERSIDAD JAVERIANA

FACULTAD DE INGENIERÍA

CARRERA DE INGENIERÍA DE SISTEMAS

BOGOTÁ, D.C.

2018

CIS1730CP06

LSC APP: APLICACIÓN MÓVIL PARA LA PRÁCTICA DE LA LENGUA DE SEÑAS
COLOMBIANA.

Autores:

Marvin Daniel Cely Báez

Sergio Forero Gómez

Jose Darío Guerrero Aragón

MEMORIA DEL TRABAJO DE GRADO REALIZADO PARA CUMPLIR UNO DE LOS
REQUISITOS Y OPTAR AL TÍTULO DE INGENIERO DE SISTEMAS

Directora

Ingeniera Luisa Fernanda Barrera León

Jurados del trabajo de grado

Sitio web del trabajo de grado

<http://pegasus.javeriana.edu.co/~CIS1730CP06/>

PONTIFICIA UNIVERSIDAD JAVERIANA

FACULTAD DE INGENIERÍA

CARRERA INGENIERÍA DE SISTEMAS

BOGOTÁ, D.C.

JUNIO, 2018

PONTIFICIA UNIVERSIDAD JAVERIANA

FACULTAD DE INGENIERÍA

CARRERA DE INGENIERÍA DE SISTEMAS

Rector Magnífico

Jorge Humberto Peláez Piedrahita, S.J.

Decano Facultad de Ingeniería

Ingeniero Jorge Luis Sánchez Téllez

Director de la Carrera de Ingeniería de Sistemas

Ingeniera Mariela Josefina Curiel Huérfano

Director Departamento de Ingeniería de Sistemas

Ingeniero Efraín Ortíz Pabón

Artículo 23 de la Resolución No. 1 de junio de 1946

“La Universidad no se hace responsable de los conceptos emitidos por sus alumnos en sus proyectos de grado. Sólo velará porque no se publique nada contrario al dogma y la moral católica y porque no contengan ataques o polémicas puramente personales. Antes bien, que se vean en ellos el anhelo de buscar la verdad y la Justicia”

AGRADECIMIENTOS

Gracias a la ingeniera Luisa Barrera por ofrecer su apoyo como directora de tesis, aprovechamos al máximo sus consejos y charlas en las reuniones con ella donde el proyecto tuvo un orden y un referente de gestión, sobre todo la realimentación ofrecida por la ingeniera, que permitió ir moldeando el proceso.

Gracias también al ingeniero Cesar Bustacara que nos ayudó en nuestro proceso para esclarecer la propuesta de grado. Inicialmente se tenía la idea de crear un traductor del LSC a lenguaje español, para ello el alcance de ese proyecto hubiera sido sumamente extenso para el contexto que se tenía en este trabajo de grado, por eso se fue explorando otras ideas alternativas encontrando la idea de una aplicación que permite la práctica del LSC. Debido a esas reuniones donde el grupo pasó horas y horas charlando con el ingeniero, se pudo lograr construir una propuesta clara e interesante. El grupo aprovechó al máximo esas reuniones obteniendo realimentación para construir una propuesta bien definida. Con estas aclaraciones guiadas por el ingeniero, el proyecto comenzó a dar forma de la mano de las investigaciones y el análisis de la problemática.

Damos gracias al grupo de Fenascal por recibirnos y darnos información referente al trabajo de grado, la reunión que tuvimos con ellos fue enriquecedora siendo un primer paso para construir las bases de nuestro proyecto.

Gracias a Lenily Báez y Melisa Hernández, profesoras en educación especial, por permitirnos entrevistarlas y ayudarnos a aclarar conceptos para practicar la lengua de señas colombiana, fue grato charlar con ellas. Gracias de nuevo a Melisa Hernández por ayudarnos en grabar la primera versión de videos donde ella realizó las señas, fue un trabajo muy agradable.

Finalmente, gracias a Miguel Mejía por ser nuestro modelo sordo el cual es líder en la comunidad sorda de Colombia, junto a Angélica Patiño por el acompañamiento en la interpretación entre nosotros los oyentes y la comunidad sorda, adicional por el apoyo en que la grabación fue lo esperado y terminó siendo contenido importante para la aplicación móvil.

Tabla de contenido

1.	OPORTUNIDAD, PROBLEMÁTICA Y ANTECEDENTES	2
1.1.	Contexto del problema	2
1.2.	Formulación del problema	3
1.3.	Propuesta de solución	3
1.4.	Justificación de la solución	4
2.	DESCRIPCIÓN DEL PROYECTO	5
2.1.	Objetivo general	5
2.2.	Objetivos específicos	5
2.3.	Entregables, estándares y justificación	5
1.	Conceptos fundamentales	7
2.	Trabajos importantes en el área	8
1.	Obtención de información	11
1.1.	Entrevista a profesionales con conocimiento del LSC	11
1.2.	Metodología de la entrevista y resultados	11
1.3.	Diseño del contenido	12
1.4.	Instructivo de técnicas de ludificación	14
1.4.1.	Características del juego	14
1.4.2.	Mecánica del juego	15
2.	Historias de usuario	21
3.	Restricciones de la aplicación	23
4.	Mockups	24
5.	Casos de uso	25
1.	Elección de tecnologías	27
1.1.	Algoritmo de reconocimiento de gestos	27
1.2.	Aplicación (front-end)	28
1.3.	Servidor (back-end)	30
1.3.1.	Diseño de la arquitectura	30

1.3.2.	Spring y Mongo DB	31
1.3.3.	Amazon Web Services	33
1.4.	Administrador (front-end)	34
2.	Arquitectura de la aplicación	35
2.1.	Arquitectura aplicación (front-end)	35
2.2.	Arquitectura servidor (back-end)	39
2.2.1.	Componentes	39
2.2.2.	Despliegue	42
2.3.	Arquitectura del administrador (front-end)	43
1.	Metodología	45
1.1.	Fase de iniciación	45
1.2.	Fase de construcción	48
1.3.	Fase de final	50
2.	Producto final	52
2.1.	Aplicación (front-end)	52
2.2.	Servidor (back-end)	56
2.2.1.	Despliegue en AWS	56
2.2.2.	Secuencia de despliegue	58
2.2.3.	Peticiones	59
2.3.	Administrador web (front-end)	60
1.	Control de calidad	62
1.1.	PRUEBAS DE LA APLICACIÓN	62
1.1.1.1.	Pruebas de instrumentación	62
1.1.1.2.	Pruebas de usuario	62
1.1.1.2.1.	Pruebas de usabilidad	63
1.1.1.2.2.	Pruebas de aceptación	65
1.2.	PRUEBAS DEL SERVIDOR	66
1.2.1.	Pruebas de API	66

1.	Análisis del impacto del proyecto	70
2.	Conclusiones	70
3.	Trabajo futuro	72
1.	Carta de autorización (licencia de uso)	76
2.	Formato descripción de trabajo de grado	79

TABLAS

Tabla 1: Comparación de medios de acercamiento al lenguaje de señas.	3
Tabla 2: Entregables y estándares.	6
Tabla 3: Niveles, lecciones y palabras	13
Tabla 6: Descripción de las recompensas.	19
Tabla 4: Historias épicas e historias de usuario.	21
Tabla 5: Ficha de restricciones.	23
Tabla 7: Descripción de los casos de uso.	25
Tabla 8: Basic Gesture Detection.	26
Tabla 9: Hand Gesture Detection AI With Convolutional Neural Networks.	26
Tabla 10: Microsoft Custom Vision Service.	27
Tabla 11: API Vision de Cloud.	27
Tabla 12: Amazon Rekognition Image.	28
Tabla 13: Comparación de diseños arquitecturales.	31
Tabla 14: Especificación de nodos.	43
Tabla 15: Mapeo de puertos por componente.	58
Tabla 16: Resultados interfaz de la aplicación.	63
Tabla 17: Resultados elementos de la aplicación.	63
Tabla 18: Resultado terminología e información del sistema.	63
Tabla 19: Aprendizaje.	64
Tabla 20: Resultado contenido.	64
Tabla 21: Resultados de pruebas de aceptación.	64
Tabla 22: Resultado de encuesta NPS.	65

FIGURAS

Figura 1: Nivel, lecciones y prácticas.	15
Figura 2: Diagrama de casos de uso.	24
Figura 3: Aplicación móvil componentes arquitectónicos	35
Figura 4: Paquetes principales en LSCApp.	36
Figura 5: Definición de API de comunicación con servidor AWS.	36
Figura 6: Entidades utilizadas en la aplicación.	37
Figura 7: Flujo de comunicación entre componentes a través de observables.	38
Figura 8: Paquetes principales ui	39
Figura 9: Diagrama de componentes servidor.	40
Figura 10: Diagrama de despliegue.	42
Figura 11: Diagrama de componente del administrador.	44
Figura 12: Metodología de investigación y desarrollo.	45
Figura 13: Fase de iniciación.	48
Figura 14: Fase de construcción.	50
Figura 15: Fase final.	51
Figura 16: Formulario iniciar sesión.	52
Figura 17: Formulario registrarse.	52
Figura 18: Validación de campos.	52
Figura 19: Selección de nivel.	53
Figura 20: Menú principal.	53
Figura 21: Diccionario.	53
Figura 22: Lecciones en nivel Abecedario y números.	53
Figura 23: Lecciones en nivel Sustantivos.	53
Figura 24: Listado de lecciones en nivel Predicados.	53
Figura 25: Práctica show-sign.	54
Figura 26: Botón de siguiente deshabilitado.	54
Figura 27: Botón siguiente habilitado.	54
Figura 28: Caso de fallo.	55
Figura 29: Caso exitoso.	55
Figura 30: Práctica translate-video.	55
Figura 31: Práctica take-sign.	55
Figura 32: Práctica discover-image.	55
Figura 33: logro obtenido al finalizar práctica.	56
Figura 34: Despliegue en AWS.	56
Figura 35: Secuencia de despliegue.	59

Figura 36: Ejemplo de petición REST.	60
Figura 37: Administrador web recibir datos.	61
Figura 38: Administrador web agregar video.	61
Figura 39: Distribución de personas que conocen el LSC.	62
Figura 40: Proceso de ejecución de pruebas.	68
Figura 41: Ejemplo de reporte para pruebas automatizadas de integración.	68
Figura 42: Desafío de agilidad.	71

ABSTRACT

LSC app is a mobile application designed and developed for people to practice the Colombian Sign Language (LSC). The application makes use of gamification techniques considering the coverage of the language lexicon. Additionally, the application integrates the functionality of an image recognition algorithm where the person captures the photo of their hand, making a static signal, and therefore the application determines the probability if the signal is correct or not. As a result, this playful tool allows the LSC to be disseminated to those people who do not know about it, favoring the approach of the deaf population by breaking the barriers of communication.

RESUMEN

LSC app es una aplicación móvil diseñada y desarrollada para que las personas practiquen la Lengua de Señas Colombiana (LSC). La aplicación hace uso de técnicas de ludificación teniendo en cuenta la cobertura del léxico de la Lengua. Adicionalmente, la aplicación integra la funcionalidad de un algoritmo de reconocimiento de imagen donde la persona captura la foto de su mano, realizando una seña estática, y por consiguiente la aplicación determina la probabilidad si la seña realizada es correcta o no. En consecuencia, esta herramienta lúdica permite difundir el LSC en aquellas personas que la desconocen favoreciendo el acercamiento de la población sorda rompiendo las barreras de comunicación.

I- INTRODUCCIÓN

La LSC, es una lengua viva en nuestro país que es desconocida por gran parte de la población Colombiana, lo que genera barreras de integración para las personas con discapacidad auditiva. Buscando disminuir esta barrera, nace la aplicación móvil **LSC App** que tiene el propósito de divulgar y fomentar la práctica del LSC usando un medio masivo de comunicación como son los teléfonos inteligentes. La aplicación permite conocer el LSC a las personas oyentes que lo desconocen y sirve como herramienta de practica a las personas que ya la conocen. Usando métodos de ludificación, la aplicación se divide en niveles cuya dificultad aumenta a medida que el usuario progresa y premia notificando los logros superados. Estos niveles están diseñados para que la persona logre entender el concepto para posteriormente identificarlo en las prácticas de los niveles siguientes. Adicionalmente, la aplicación tiene un nivel que utiliza el servicio *CustomVision* de Microsoft, como algoritmo de reconocimiento de gestos por medio de imágenes, donde el usuario captura una foto de la mano realizando una seña estática y a través del servicio de reconocimiento de gestos y el algoritmo que provee se indica la probabilidad de que la seña mostrada en la imagen sea correcta.

Como servicio adicional, la aplicación cuenta con un diccionario del LSC que contiene las palabras usadas en las prácticas de señas. Dicho diccionario puede ser actualizado ya que se cuenta con un servicio de administración, implementado en la web, donde el usuario puede agregar, actualizar y eliminar palabras. Este servicio es muy útil ya que dicho vocabulario va creciendo al pasar el tiempo y el aprendizaje del usuario es mayor.

En el presente documento se encuentra la descripción completa del trabajo realizado, mostrando los hitos logrados con respecto a la propuesta. La siguiente memoria de grado describirá los procesos llevados a cabo mostrando todos los artefactos que se necesitaron para el desarrollo y gestión del proyecto.

II- DESCRIPCIÓN GENERAL

1. OPORTUNIDAD, PROBLEMÁTICA Y ANTECEDENTES

1.1. Contexto del problema

Según el censo del DANE del año 2005, se identificó que el 1,1% de la población total cuenta con alguna limitación de escucha. Para el día 31 de agosto del año 2016, el censo de la población registró un total de 48.839.671 de personas, de las cuales, asumiendo una proyección lineal sobre el 1,1% del 2005, se identifica un aproximado de 537,235 personas con limitación para oír en la actualidad [1]. La población de personas sordas no se caracteriza únicamente por tener una limitación auditiva, sino porque en su necesidad por relacionarse con los otros adquieren y se comunican por medio una lengua de tipo viso-gestual. De acuerdo con el país hay una lengua de señas, por ejemplo, la Lengua de Señas Colombiana (LSC), el cual cuenta con una gramática propia, ágrafa, e identifica a sus usuarios como una comunidad lingüística minoritaria [2].

En el ámbito familiar, se conoce que el 95% de las personas sordas provienen de familias oyentes usuarias de lenguas vocal-auditivas, lo que los conduce a vivir en una condición de bilingüismo, siendo el español escrito, la segunda lengua como propuesta educativa para solucionar la necesidad de comunicarse con las demás personas [3].

Adicionalmente, en el ámbito educativo, el registro temporal de intérpretes de LSC del INSOR señala que, para el año 2011, 361 personas son reconocidas como intérpretes ‘certificados’. Esto en términos educativos no son cifras optimistas, pues al revisar el registro de matrícula de la población sorda, proveniente del Sistema Integrado de Matriculas (SIMAT) del Ministerio de Educación Nacional, para el año 2011, 9.887 estudiantes sordos fueron matriculados, por lo que el número de intérpretes para dichos sordos en promedio fue de 1 por cada 27. Relación que no cumple con la resolución 2565 de 2003 que señala que, “por lo menos un intérprete de LSC por cada grupo que integre hasta trece (13) estudiantes sordos en educación básica secundaria o media [4].

En conclusión, se puede observar que en los diferentes ámbitos sociales en los que se relacionan las personas sordas existen dificultades en la comunicación, debido a que aproximadamente el 98.9% de la población colombiana son personas con capacidad auditiva, donde la mayoría desconoce el LSC y no existe la cantidad suficiente de intérpretes que puedan dar solución a la demanda requerida.

1.2. Formulación del problema

Debido a la dificultad de comunicación de las personas con discapacidad auditiva con el resto de la población, generada por el desconocimiento del LSC ¿cómo las tecnologías de la información pueden contribuir para reducir dicho desconocimiento?

1.3. Propuesta de solución

Fuera de los salones de clase existen varios métodos que permiten el acercamiento al lenguaje de señas, vídeos de enseñanza en plataformas masivas, recursos y herramientas web, aplicaciones móviles, entre otros. Cada método concuerda en llegar a millones de personas, más difieren en el acceso a los datos y los modos de interacción.

	Presencia global	Acceso a recursos del dispositivo	Entretenimiento educativo	Conexión requerida
Videos de enseñanza	-	No	No	Si
Recursos y herramientas web	44.78% ordenadores	Si	Si	Si
Aplicaciones móviles	50.87% dispositivos móviles	Si	Si	No

Tabla 1: Comparación de medios de acercamiento al lenguaje de señas.

Las convenciones pertenecientes a la tabla 1 son “-”, no aplica, “Si”, si cumple la característica y “No”, no cumple la característica.

Como se muestra en la tabla 1, con información tomada de [5] realizando una breve comparación entre los medios de difusión, encontramos que las aplicaciones móviles lideran en accesibilidad, la cantidad de usuarios finales, el acceso a recursos de los dispositivos y la posibilidad de establecer ludificación en el contenido.

En el presente proyecto se propone realizar una herramienta a manera de aplicación móvil, con el fin de mejorar la comunicación de las personas con discapacidad auditiva, con el resto de la población, reduciendo el desconocimiento del lenguaje de señas, y promoviendo el incremento de intérpretes a nivel nacional. Se difunden actividades para la práctica del LSC, donde personas con cualquier nivel de conocimiento del lenguaje, con intención de aprendizaje autónomo, tengan la oportunidad de realizar ejercicios prácticos designando los tiempos y espacios para el desarrollo de estas.

1.4. Justificación de la solución

El proyecto está guiado a solucionar el problema del desconocimiento del LSC, por ende, la propuesta va enfocada a la divulgación de este lenguaje. Como existen muchas formas para difundir la información, ejemplo típico usando volantes, se ha llegado a la conclusión que la mejor forma para difundir el lenguaje es usando un medio masivo de comunicación.

Se considera elegir los teléfonos inteligentes, pues, se encontró que en Colombia 23'748.167 de las personas poseen teléfonos con acceso a internet, esto da a entender que son teléfonos inteligentes [5]. Y para precisar, de los teléfonos inteligentes el 89% de los usuarios en Colombia utilizan el sistema operativo Android diferente a los usuarios iOS que hacen parte del 7.7% [6]. Sintetizando, aproximadamente 21'135.868 de personas usan teléfonos inteligentes con Android, las cuales hacen referencia a la población potencial a la que puede ser difundida. Partiendo de ese hecho, se decide desarrollar una aplicación móvil porque

permite a las personas acceder de manera práctica y sencilla, diferente al uso de un computador.

Finalmente, para contribuir a la solución se genera mecanismos de difusión como la aplicación móvil que permite a las personas, sin discapacidad auditiva, revisar y practicar el lenguaje, que en consecuencia generaría cultura de autoaprendizaje y solucionaría la problemática de la comunicación reduciendo los niveles de analfabetismo del LSC.

2. DESCRIPCIÓN DEL PROYECTO

2.1. Objetivo general

Desarrollar una aplicación móvil que permita la práctica del diccionario básico de la lengua de señas colombiana, integrando técnicas de ludificación y el reconocimiento de gestos.

2.2. Objetivos específicos

- ✓ Definir la cobertura de léxico a utilizar con relación al diccionario básico de la lengua de señas colombiana.
- ✓ Diseñar la interfaz de usuario en la que se incluyan técnicas de ludificación para la motivación en la práctica del LSC.
- ✓ Definir un algoritmo de procesamiento de imágenes que permita identificar de manera acertada los gestos realizados por el usuario.
- ✓ Diseñar e implementar una aplicación móvil que integre la cobertura de léxico, la interfaz de usuario y el algoritmo de procesamiento de imágenes, con el fin de disponer una herramienta para la práctica del LSC.
- ✓ Evaluar la aplicación móvil aplicando una metodología de aceptación.

2.3. Entregables, estándares y justificación

A continuación, se presentan los entregables internos y externos durante el desarrollo del proyecto, algunos asociados a estándares, justificando su realización.

Entregable	Estándares asociados	Justificación
Cobertura de léxico	No aplica	Basado en el diccionario de la lengua de señas colombiana, gestos a incluir en la aplicación móvil.
Plan de ludificación	No aplica	Documento guía que especifica las técnicas de ludificación dentro de la aplicación móvil.
SPMP (Plan de Administración del Proyecto de Software)	ISO/IEC/IEEE 16326	Describe el cronograma, las actividades, recursos y metodología para el desarrollo del proyecto.
SRS (Especificación de Requisitos de Software)	ISO/IEC/IEEE 29148	Establece el alcance del proyecto, además de las restricciones y funcionalidades.
Plan de pruebas funcionales	IEEE 29119	Guía para probar el funcionamiento del sistema sin usuarios.
Plan de pruebas de aceptación	IEEE 29119	Guía para probar el funcionamiento del sistema con usuarios.
SDD (Descripción del Diseño del Software)	ISO/IEC/IEEE 1016	Diseño arquitectural y detallado del sistema.
Documento de cambios y versionamiento	IEEE 29119	Desarrollo documentado aplicando buenas prácticas de programación.
LSC App	No aplica	Aplicación móvil final desplegada en la tienda Google Play Store.
Documentación del resultado de las pruebas	No aplica	Refleja el proceso, resultados y conclusiones de las pruebas realizadas.
Manual de usuario	IEEE 1063	Guía para el uso de la aplicación móvil.

Tabla 2: Entregables y estándares.

III- CONTEXTO DEL PROYECTO

1. Conceptos fundamentales

Población sorda colombiana

Es un grupo heterogéneo, integrado por personas que por causas distintas tienen una deficiencia auditiva, circunstancia que puede presentarse en cualquier momento de la vida y que obliga a quien la posee a valerse primordialmente de su visión para relacionarse con el medio y comunicarse con otras personas [2].

LSC

Lenguaje de Señas Colombiano, es la lengua empleada por la población sorda colombiana para relacionarse y comunicarse. Es una lengua natural y puede estudiarse en todos los niveles lingüísticos: fonológico, morfológico, semántico y pragmático.

INSOR

El Instituto Nacional para Sordos (INSOR), es una entidad adscrita al Ministerio de Educación Nacional tiene como objetivo asesorar al Gobierno nacional, departamental y municipal en la formulación de políticas, planes, programas y proyectos para el desarrollo integral de la Población Sorda Colombiana [2].

Diccionario Básico de la Lengua de Señas Colombiana

Diccionario básico con un número de 1.200 entradas, elaborado por lingüistas y lexicógrafos del departamento de lexicografía del Instituto Caro y Cuervo. Se basa en el uso real de la lengua de señas del Valle y Bogotá; por lo tanto, describe dos variedades de la lengua de señas de dos regiones importantes de Colombia. Es dirigido a cualquier persona que esté interesada en aprender lengua de señas, profesores, padres e incluso personas sordas [7].

2. Trabajos importantes en el área

Después que la propuesta comenzó a surgir fue necesario buscar información de herramientas o programas que se parecerán a la propuesta, con el fin de cerciorarse que la idea no ha sido realizada. A continuación, se mostrarán los trabajos que se asemejaron a la propuesta.

Modelo de identificación y clasificación del lenguaje de señas colombiano [8]

Trabajo de grado el cual identifica que el LSC se ve afectado por neologismos o extranjerismos, los que resultan en la creación de términos específicos por región que conllevan a un crecimiento separado del lenguaje. El proyecto propone una solución tecnológica que permita efectuar un control sobre el crecimiento del léxico del LSC que además contribuya en la difusión, estandarización y aprendizaje de nuevos gestos.

Para lograr su objetivo, el sistema propuesto, es una aplicación web, la cual accede a un repositorio centralizado que contiene toda la información unificada del estándar del LSC junto con los nuevos gestos propuestos por la comunidad. Permite entonces a los usuarios: crear y almacenar nuevos gestos mediante la captura de video; consultar los gestos del repositorio y contribuir mediante opiniones y calificaciones; administrar la información asociada a cada gesto, como, modificación del movimiento del gesto, creación de nuevos gestos mediante la fusión con otros gestos y observar las trayectorias que componen los movimientos de cada gesto.

Este proyecto, plantea una solución útil en la gestión organizada del crecimiento del LSC, no obstante, carece en abarcar el procesamiento que se debe realizar para obtener la información del movimiento de una nueva seña propuesta. Únicamente define el formato del archivo con la información de la representación del gesto, por lo que una pieza fundamental para lograr el objetivo principal del proyecto está faltante.

Android based real-time static Indonesian sign language recognition system prototype [9]

Los autores del artículo se interesan en desarrollar un prototipo basado en Android que sea capaz de traducir lenguaje de señas de indonesia a lenguaje escrito con el fin de reducir la dificultad de comunicación para personas con discapacidad en la escucha y el habla. El proceso de traducción consta de los siguientes pasos: Eliminar el fondo de la imagen; Identificar la localización y contorno de la mano; Extraer el gesto realizado por la mano y clasificar el gesto para encontrar su correspondiente traducción en el alfabeto. Adicionalmente se realizan pruebas de rendimiento sobre el prototipo en tiempo real, las cuales indican que el sistema logra una tasa del 91.66% de éxito en el reconocimiento de las señas del usuario.

El documento explica de manera clara, pero en ocasiones muy general el procedimiento realizado para lograr el objetivo de reconocimiento del lenguaje de señas. Brinda una perspectiva global de las técnicas necesarias para lograr este objetivo y es un buen punto de partida para avanzar con la investigación. En la parte técnica, sobre librerías e implementación de los algoritmos, el artículo no profundiza, lo que nos obliga a realizar futuras investigaciones para clarificar esta información. Finalmente presenta una clara exposición del procedimiento realizado para probar el sistema y sus resultados, lo cual también nos sirve como guía para nuestro proyecto.

Single-frame hand gesture recognition using color and depth kernel descriptors [10]

Este artículo presenta un método de reconocimiento de gestos de la mano, a través de imágenes estáticas. Para lograr el reconocimiento de la mano utilizan la información de color y profundidad de las imágenes, que fusionan para extraer los descriptores del *kernel*. Para la predicción de la clasificación aplican *Support Vector Machine (SVM)* reconociendo con precisión los datos (88.94%) que en este caso hacían parte de la Lengua de Signos Americana (ASL).

Este método, basado en el análisis de imágenes en 2D y extracción de información 3D, enfrenta el problema de reconocimiento a través de procesos de tratamiento de imágenes, usando herramientas como espacios de color, reconocimiento de patrones, correlación de imágenes, máscaras en el espacio y tracking motion, por lo que para lograr un seguimiento eficiente, se deben variar las condiciones de iluminación, complejo del fondo y el color de la piel para la detección de la mano, detección de bordes, descripción de patrones y extracción de características.

Reconocimiento del lenguaje de señas manuales con el Kinect [11]

En este documento se presenta el trabajo realizado para la identificación y el reconocimiento del abecedario del lenguaje de señas colombiano que se compone de letras con y sin movimiento y de palabras ejecutadas con una sola mano. Se hace énfasis especial en el reconocimiento de la forma de la mano a través del procesamiento de las imágenes de color y profundidad dadas por la captura del esqueleto que brinda el dispositivo Kinect.

Para la clasificación de las letras sin movimiento utiliza un algoritmo de aprendizaje supervisado (SVM), y para la clasificación de letras o palabras con movimiento utiliza Modelos Ocultos de Márkov (HMM), los cuales, para el reconocimiento de letras estáticas logra una precisión del 94.7%, en las letras con movimiento un 93.8% y en las palabras un 99.1% de aciertos.

Aunque el trabajo muestra resultados satisfactorios en cómo lograr el reconocimiento de algunas letras del LSC con y sin movimiento, al extraer el modelo de color personalizado y características importantes por persona, obliga que, para el reconocimiento de palabras sea obligatorio el uso de un guante, para que se pueda diferenciar cuando la palabra se ejecuta cerca de la cara. Adicionalmente solo se tiene en cuenta el movimiento de una sola mano.

IV- ANÁLISIS DEL PROBLEMA

1. Obtención de información

1.1. Entrevista a profesionales con conocimiento del LSC

Se realizó una entrevista semiestructurada a dos profesionales seleccionados debido a su conocimiento y dominio del LSC. La profesora Lenily Báez aprendió el LSC dentro del proceso de formación profesional como educadora especial y Melisa Hernández tiene el LSC como “lengua materna” y es profesional como educadora especial.

Para la realización de la entrevista se recolectó información de expertos en la enseñanza e igualmente, con conocimientos previos en el LSC, para establecer los criterios básicos de la implementación de una APP que permita la práctica de vocabulario elemental. Adicionalmente, se presentaron bocetos iniciales de la aplicación lo cual generó una realimentación y lluvia de ideas.

1.2. Metodología de la entrevista y resultados

Se realizaron tres preguntas fundamentales para obtener un conocimiento base sobre el funcionamiento del LSC, las cuales se definieron como:

- ¿Cuál es la estructura para practicar frases del LSC?
- ¿Es prioritario la enseñanza de categorías como números y letras (abecedario)?
- ¿Cuál considera usted que podría ser la estructura para practicar este vocabulario básico?

Con respecto a lo anterior, los profesionales respondieron en primer lugar, que la estructura básica del LSC utiliza sustantivo, verbo y predicado con la combinación de éstas, es importante mencionar que, en la LSC no se hace uso de artículos (la, el, los, las). Significa que la traducción del español al LSC es solamente conceptual.

En segundo lugar, se afirma la enseñanza de categorías de números y letras no se prioriza puesto que hace parte de la enseñanza lecto-escritura que se da después de que la persona

tenga un bagaje amplio en el vocabulario. Además, es muy similar a la lecto-escritura de los oyentes que parte de un vocabulario esencial.

El siguiente ejercicio realizado fue mostrarles bocetos a las entrevistadas, para obtener diversas ideas y sugerencias sobre el contenido de estos. Para ello, se tuvo en cuenta esta pregunta central.

- ¿Cómo creería usted que se podría categorizar a la secuencia didáctica?

El resultado fue categorizar la secuencia de una oración por sustantivo, verbo y predicado con la finalidad de hacer combinaciones partiendo de oraciones simples hasta crear una oración completa como:

- **Sustantivo y Verbo:**

Ej. Yo como

- **Sustantivo, Verbo y predicado (Lugar):**

Ej. Tu caminas en el centro comercial

- **Sustantivo, Verbo y predicado (Lugar y Tiempo):**

Ej. Nosotros estudiamos en la universidad todos los días

1.3. Diseño del contenido

En la tabla 3 se apreciará cómo se repartió el contenido de los niveles con sus lecciones.

Nivel	Lección	Palabras	Cantidad de palabras
Abecedario y números	Abecedario	A - Z	27
	Números	0 - 9	10
Sustantivos	Pronombres	<ul style="list-style-type: none"> • Yo • Tú • Él/Ella • Ellos • Nosotros 	5
	Relaciones	<ul style="list-style-type: none"> • Abuelo(a) • Mamá • Hermano (a) 	11

		<ul style="list-style-type: none"> • Papá • Familia 	<ul style="list-style-type: none"> • Hijo(a) • Amigo(a) 	
Verbos	Actividades cotidianas	<ul style="list-style-type: none"> • Reír • Saltar • Cantar • Bailar • Correr 	<ul style="list-style-type: none"> • Caminar • Jugar • Dormir • Comer 	9
	En el estudio	<ul style="list-style-type: none"> • Leer • Escribir • Recordar • Olvidar 	<ul style="list-style-type: none"> • Pensar • Ver • Ir 	7
Predicados	Lugares	<ul style="list-style-type: none"> • Casa • Edificio • Apartamento • Universidad • Centro comercial 	<ul style="list-style-type: none"> • Hospital • Parque • Cine • Biblioteca • Baño 	10
	Tiempos	<ul style="list-style-type: none"> • En la mañana • En la tarde • En la noche • Ayer • Hoy • Mañana • Todos los días 		7

Tabla 3: Niveles, lecciones y palabras

De esta manera si combinamos la cantidad de palabras de los sustantivos con los verbos y predicados encontramos que:

- **Nivel verbos:**
 $(16 \text{ sustantivos}) \times (16 \text{ verbos}) = 256 \text{ oraciones}$
- **Nivel predicados:**

- **Nivel lugares:**
(16 sustantivos) × (16 verbos) × (10 lugares) × (7 tiempos) = 2.560 oraciones
- **Nivel tiempos:**
(16 sustantivos) × (16 verbos) × (10 lugares) × (7 tiempos) = 17.920 oraciones

Entonces se encontró que al crear oraciones aleatoriamente en cada nivel desde el nivel de verbos hasta el nivel de predicados es posible obtener un total de *20.736 oraciones*. Esto significa que el usuario tendrá la posibilidad de utilizar una cantidad amplia de oraciones cada vez que quiera utilizar la aplicación para practicar el LSC.

Además, se agregó un cuarto nivel que hace uso del algoritmo de reconocimiento de señas estáticas, señas como el abecedario y los números. Cabe resaltar que el nivel del abecedario y números no hace parte de la estructura sustantivo, verbo y predicado expuesta anteriormente.

1.4. Instructivo de técnicas de ludificación

Ludificación es el término que se utiliza para designar el uso de mecánicas de juegos en sistemas que no lo son, para mejorar la experiencia e implicación del usuario [12]. La popularidad en el uso de elementos de ludificación en ámbitos educativos y empresariales sigue creciendo, según el ciclo de Gartner para tecnologías emergentes del 2013, que proyecta alcanzar el tope de productividad en 5 a 10 años, y esto se muestra con ejemplos de la vida real que han sido probado exitosos como los programas de incentivos para clientes basados en puntos, o el uso de insignias o medallas en sitios web educativos, para involucrar mejor a los usuarios [13].

1.4.1. Características del juego

Dado que la ludificación es una herramienta poderosa utilizada en el ámbito empresarial y educativo, en nuestra aplicación, adoptamos las dinámicas de juego como herramienta para mantener la atención del usuario e incentivar su uso reiterativo. Al considerar la definición formal de juego en el contexto de la aplicación, y basados en Kapp [14], fue necesario tener en cuenta las siguientes características principales de los juegos:

Conducido por metas

Tener metas que añadan propósito al uso de la aplicación, donde para lograrlas el usuario deba seguir ciertas reglas que delinear sus acciones. Las reglas pueden ser de “instrucción” que condensan la información que el usuario debe “aprender”, es decir las señas del LSC, y “de interacción” que evalúan y muestran la forma correcta de uso y progreso en la aplicación. Para esto la retroalimentación direccionada da al usuario información sobre si su acción fue correcta o incorrecta y adicionalmente ofrecen guía hacia la acción correcta.

Niveles e Historias

Los niveles se caracterizan por involucrar al usuario en un conjunto de metas que debe lograr, donde la dificultad de alcanzarlas aumenta y marcan el progreso del usuario. Este progreso visto como una historia, es un hilo narrativo que genera una percepción de flujo y desarrollo, lo que mantiene el interés del usuario.

Motivación

La motivación surge cuando se dan recompensas al lograr metas específicas y su obtención es notificada de forma inmediata al usuario. Las recompensas se clasifican en “principales”, como las dadas al finalizar los niveles, y las de tipo “alternas” que invitan a realizar acciones creativas y usualmente se logran luego de finalizar las principales.

1.4.2. Mecánica del juego

Luego de determinar las características de juego, se procedió a definir las mecánicas de juego, que son la forma de cómo dentro de la aplicación se utilizaron las características previas.

Niveles y lecciones

El contenido para practicar fue clasificado en niveles, donde cada uno contiene subniveles, que denominamos lecciones y cada lección se compone de prácticas como se ve en la figura 15.



Figura 1: Nivel, lecciones y prácticas.

Tanto niveles como lecciones y practicas cumplen con la característica de hilo narrativo al obligar que para continuar con el siguiente se debe haber completado el anterior, además que, al progresar la dificultad aumenta.

Prácticas:

En esta se muestra el contenido que el usuario va a practicar, palabras o frases de la LSC. Existen dos tipos principales de práctica, las de exposición y las de evaluación. Las de exposición, tienen como propósito mostrar nuevas señas al usuario para que las conozca. Las de evaluación cumplen la tarea de revisar la correcta identificación de las señas previamente expuestas. Cabe anotar que, para las prácticas de evaluación, una retroalimentación direccionada es dada al finalizar la práctica, es decir que, según la acción realizada, se indica si la acción fue correcta o no y en caso incorrecto la dirección hacia la acción correcta es mostrada. A continuación, se detallan cada una de las practicas definidas para la aplicación especificando para cada una:

- **Tipo:** el tipo de práctica, si es de exposición o evaluación.
- **Código:** Abreviatura para referenciar en otros documentos y el desarrollo.
- **Descripción:** Texto que describe la mecánica de la práctica.
- **Fuente:** De donde se obtuvo la idea de la mecánica de la práctica. En caso de ser una idea propia, se denomina original.

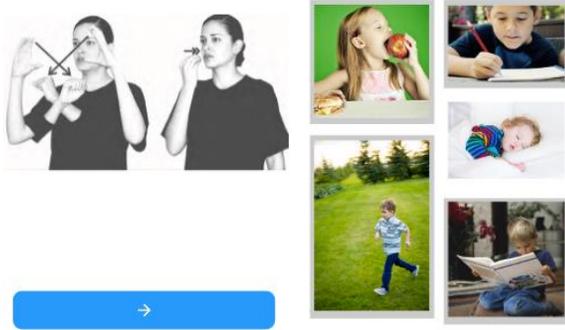
Mostrar palabra					
Tipo:	exposición	Código:	show-sign	Fuente:	Original
 <div style="border: 1px solid blue; display: inline-block; padding: 2px 10px; margin: 5px auto;">Yo</div>					
Descripción:	Se muestra un video de la seña y abajo la palabra que representa				

Cual de esta es (palabra a video)					
Tipo:	evaluación	Código:	which-one-videos	Fuente:	Duolingo
<p>¿Cuál de estos es "Yo"?</p> <div style="display: flex; flex-wrap: wrap; justify-content: center; gap: 10px;"> <div style="border: 1px solid gray; padding: 5px;"></div> <div style="border: 1px solid gray; padding: 5px;"></div> <div style="border: 1px solid gray; padding: 5px;"></div> <div style="border: 1px solid gray; padding: 5px;"></div> </div>					
Descripción:	Pregunta por una palabra (ej. cual de esta es "Yo") y muestra cuatro videos, se debe seleccionar el correcto				

Cual de esta es (video a palabra)					
Tipo:	evaluación	Código:	which-one-video	Fuente:	Duolingo
<p>Selecciona la opción correcta</p>  <div style="display: flex; justify-content: center; gap: 10px; margin-top: 10px;"> <div style="background-color: #ADD8E6; padding: 5px 15px; border: 1px solid #ADD8E6;">Yo</div> <div style="background-color: #ADD8E6; padding: 5px 15px; border: 1px solid #ADD8E6;">Nosotros</div> <div style="background-color: #ADD8E6; padding: 5px 15px; border: 1px solid #ADD8E6;">Ellos</div> </div>					
Descripción:	Muestra un video describiendo una palabra, y varias palabras en texto para seleccionar la correcta				

Traduce esta oración (video)					
Tipo:	evaluación	Código:	translate-video	Fuente:	Duolingo
<p>Traduce esta oración en el mismo orden que se muestra el video</p>  <div style="display: flex; justify-content: center; gap: 10px; margin-top: 20px;"> <div style="border: 1px solid #ccc; padding: 2px 5px;">Yo</div> <div style="border: 1px solid #ccc; padding: 2px 5px;">Comer</div> <div style="border: 1px solid #ccc; padding: 2px 5px;">Nosotros</div> <div style="border: 1px solid #ccc; padding: 2px 5px;">Estudiar</div> <div style="border: 1px solid #ccc; padding: 2px 5px;">Jugar</div> <div style="border: 1px solid #ccc; padding: 2px 5px;">Tu</div> <div style="border: 1px solid #ccc; padding: 2px 5px;">Centro comercial</div> <div style="border: 1px solid #ccc; padding: 2px 5px;">Carne</div> </div>					
Descripción:	Muestra a video describiendo una oración y diferentes palabras en español, donde unas corresponden a la oración mostrada. Se deben seleccionar las palabras correspondientes en el mismo orden que se muestran en el video.				

Capturar seña					
Tipo:	evaluación	Código:	take-sign	Fuente:	Original
<p>Realiza la seña con la mano y tomale una foto.</p> 					
Descripción:	Debe tomar una foto de la letra o número que se indique. La foto es analizada y se da respuesta si corresponde con la indicación.				

Descubrir la Imagen					
Tipo:	evaluación	Código:	discover-image	Fuente:	Original
<p>Pon mucha atención al video que esta describiendo?</p> <p>Selecciona la imagen correcta segun el video anterior</p> 					
Descripción:	Se muestra video donde mediante señas se describe algo. Luego se exponen imágenes donde el usuario debe seleccionar la correcta.				

Progreso

Durante el transcurso de las prácticas, la aplicación indica al usuario con una barra el progreso alcanzado. Adicionalmente se lleva un progreso general de la aplicación que indica la cantidad de lecciones completadas, este progreso se puede observar en el perfil del usuario.

Recompensas

En la aplicación las recompensas se definieron como logros, representados como insignias otorgadas que se muestran inmediatamente cuando el usuario alcanza el logro. Basados en la clasificación de recompensas, se definieron los logros principales, dados al terminar cada lección y que habilitan poder continuar con siguientes lecciones, y los logros alternativos que se describen en la siguiente tabla:

Nombre	Descripción	Insignia
¡A deletrear!	Dado al practicar todas las señas del abecedario	
¡Estás en racha!	Dado al completar una lección sin equivocarse	
¡Llegaste lejos!	Dado al completar todos los niveles	
Conteo	Dado al terminar la lección de números	
Desbloqueador	Dado al desbloquear todos los niveles	
Determinado	Dado al completar el ultimo nivel sin equivocarse	
Estratégico	Dado al completar el nivel de verbos	
Lección realizada	Dado al completar la primera lección	

Tabla 6: Descripción de las recompensas.

2. Historias de usuario

Previo a la planeación de la primera iteración, se propuso la elaboración de historias de usuario, que especifica las funcionalidades necesarias para la interacción de los usuarios con el sistema. Cada historia de usuario se describió relacionando una historia épica, un estado, un escenario, unos supuestos y varios criterios de aceptación.

A continuación, se explica cada componente de una historia de usuario:

- **Historia épica:** agrupación de historias de usuario divididas por funcionalidades.
- **Estado:** describe la situación actual de la historia en el desarrollo. Los posibles estados son “Por hacer”, “En progreso”, “Pruebas”, “En corrección” y “Finalizado”. Cada historia inicia en el estado por hacer, al momento de su desarrollo pasa a en progreso, donde al finalizar se transita a pruebas, y de tener fallas se clasifica como en corrección, seguido nuevamente por pruebas; o en caso contrario, se da como finalizado.
- **Escenario:** escrito de forma “**Como** tipo de usuario, **quiero** funcionalidad, **para poder** justificación”, detalla el qué de una funcionalidad esperada del sistema, para un usuario específico.
- **Supuestos:** condiciones previas que debe cumplir el usuario.
- **Criterios de aceptación:** describe los requisitos que se deben cumplir para dar una historia de usuario por hecha.

A partir del esquema anterior descrito se propusieron 3 historias épicas y 8 historias de usuario, explicadas y detalladas utilizando la herramienta AirTable, que permitió describir y compartir cada historia con el grupo de trabajo y la directora de trabajo de grado. Además, estas historias de usuario quedaron directamente relacionadas con los casos de uso y también con los posteriores mockups (bocetos) realizados donde cada uno se clasificó con un código de orden y nombre específico.

En la tabla 4 se nombran las historias épicas e historias de usuario que contienen.

Historia épica	Historias de usuario	Mock-up relacionado	Caso de uso relacionado
Gestión de usuarios	Crear cuenta	<ul style="list-style-type: none"> ● Pantalla de inicio ● Crear cuenta 	Crear cuenta
	Iniciar sesión	<ul style="list-style-type: none"> ● Pantalla de inicio ● Iniciar sesión 	Iniciar sesión
	Cerrar sesión	<ul style="list-style-type: none"> ● Barra de menú 	
	Ver perfil	<ul style="list-style-type: none"> ● Barra de menú ● Perfil 	Ver perfil
	Editar perfil	<ul style="list-style-type: none"> ● Configuración 	Editar perfil
Práctica de señas	Desarrollar nivel	<ul style="list-style-type: none"> ● Sustantivo ● Verbos ● Predicados ● Práctica lúdica 	Desarrollar prácticas
	Ver diccionario de señas	<ul style="list-style-type: none"> ● Diccionario de señas ● Diccionario de señas - Definición 	Ver diccionario de señas
Construcción del sistema	Gestionar recursos	<ul style="list-style-type: none"> ● Administrador web 	Gestionar recursos

Tabla 4: Historias épicas e historias de usuario.

Para el proceso de validación de historias de usuario se utilizó el modelo INVEST que caracteriza una historia de usuario bien definida [15]. En este proceso se analizó las descripciones de cada historia de usuario con las siguientes características del modelo que son:

- **Independiente:** La historia de usuario deben poder implementarse, probarse y entregarse por sí mismas sin depender de otras funcionalidades.
- **Negociable:** La historia de usuario puede ser discutida según la necesidad del negocio.
- **Valiosa:** La historia de usuarios proveerá valor a los clientes, usuarios y demás stakeholders del producto.
- **Estimable:** La historia de usuario se puede priorizar y planificar.
- **Pequeña (Small):** La historia de usuario debería ser lo suficientemente pequeña como para poder ser implementada en una iteración.
- **Verificable (Testable):** Las historias de usuario debería poder escribir pruebas que verifiquen que el software de la historia funcione adecuadamente.

Un criterio que se tuvo en cuenta fue el de aceptar una historia de usuario si todas las características del modelo INVEST quedaban aceptados.

Después de tener definidas las historias de usuario se inició el proceso de desarrollo, y se realiza el diseño de las pruebas, asegurando que cada requisito se cumpliera con un caso positivo y un caso alternativo (sección Control de calidad).

Para tener más detalle ver [Especificación de las historias de usuario](#).

3. Restricciones de la aplicación

Para las restricciones de la aplicación se tomaron tres elementos principales: los dispositivos móviles, el algoritmo de reconocimiento de imágenes y la aplicación para administración de las palabras a mostrar en la aplicación móvil. Se identificaron las restricciones principales las cuales son descritas en la tabla 5, permitiendo definir lo que la aplicación puede y no puede hacer.

Componente	Descripción
Aplicación móvil	La versión del Android debe ser 6.0 o más.
	El contenido de la aplicación solo podrá servir con conexión a internet.
	Para el nivel de reconocimiento de imágenes se recomienda: <ul style="list-style-type: none"> • La foto tiene que ser solamente de la mano realizando la seña. • La foto debe estar iluminada y bien definida. • El fondo de la foto tiene que contrastar con la mano. • La mano tenga artefactos como manillas, mangas o relojes. • El peso de la foto debe ser menor a 4MB.
Administrador web	Al momento de cargar o editar contenido de la aplicación se debe tener en cuenta: <ul style="list-style-type: none"> • Las dimensiones de los videos deben ser de 720 pixeles de lados iguales. • El formato del video debe ser “.mp4”. • El formato de las imágenes debe ser “.jpg”. • Las imágenes deben pesar menos de 1MB.

Tabla 5: Ficha de restricciones.

4. Mockups

Diseñar los mockups representaron el punto de partida para crear las historias de usuario y casos de uso. Ayudaron a identificar las funcionalidades además de dar una primera impresión al momento que se desarrolló la aplicación. Se crearon *mockups* para la aplicación móvil y para el administrador web con la herramienta Adobe XD CC que permitió visualizar la posible interacción del usuario con la aplicación. Los colores y los iconos usados fueron tomados de una base de datos de iconos llamada Flaticon [16]. Para ver el detalle de los *mockups* de la aplicación referirse al [anexo](#).

5. Casos de uso

Con el fin de una representación gráfica de la interacción de los actores con el sistema, en la figura 16 se evidencia un diagrama de casos de uso, compuesto por 10 casos de uso y 2 actores. Los 2 actores definidos por el sistema son:

- **Usuario:** persona con un perfil en la aplicación, cuya intención es practicar el LSC.
- **Administrador:** persona con permisos para gestionar el contenido de la aplicación.

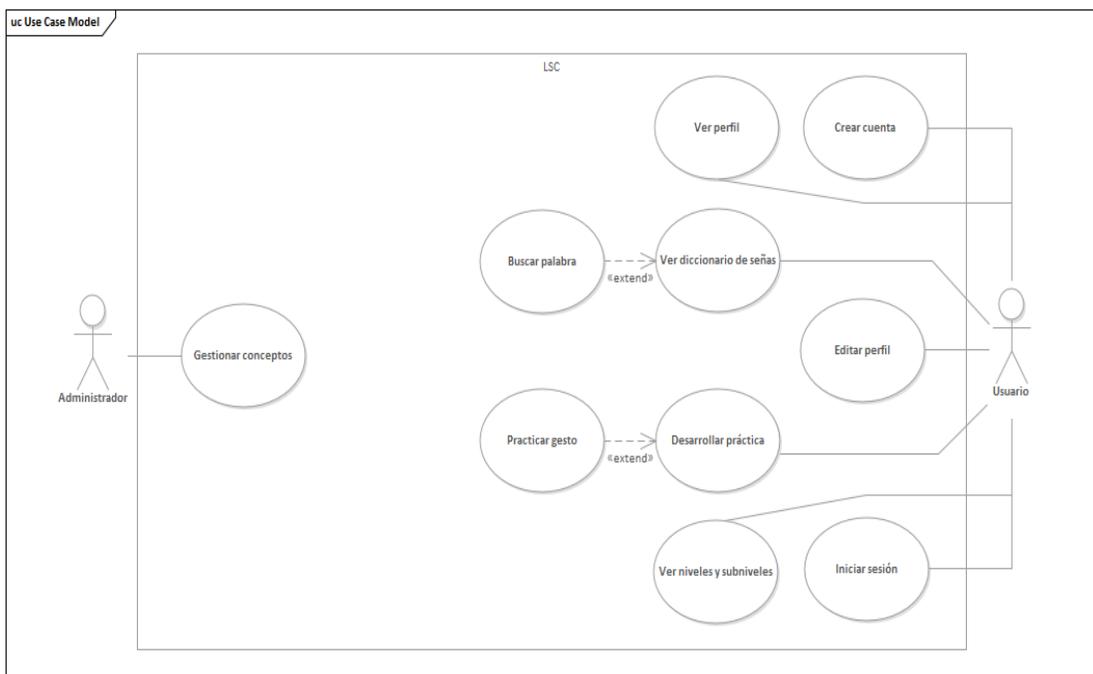


Figura 2: Diagrama de casos de uso.

En la tabla 7 se describen en más detalle los casos de uso del sistema descritos en la figura previa

Caso de uso	Descripción
Crear cuenta	Permite crear un perfil en la aplicación.
Iniciar sesión	Accede a la aplicación mediante una cuenta registrada.
Ver perfil	Detalla los datos asociados a una persona, además del avance de este dentro de los ejercicios lúdicos.
Editar perfil	Actualiza los datos de un usuario en un perfil.
Ver niveles y subniveles	Describe las categorías contenidas en la aplicación, basados en técnicas de ludificación.
Ver diccionario de señas	Muestra todas las palabras incluidas en el sistema.
Buscar palabra	Filtra la lista de palabras del CU Ver diccionario de señas, por un criterio especificado.
Desarrollar práctica	Interacción del usuario en las diferentes prácticas del LSC.
Practicar gesto	Compara el gesto realizado por un usuario a partir de una foto, mediante el algoritmo de reconocimiento de gestos.
Gestionar conceptos	Posibilita la creación, edición o supresión de nuevas palabras en el sistema.

Tabla 7: Descripción de los casos de uso.

V- DISEÑO DE LA SOLUCIÓN

1. Elección de tecnologías

1.1. Algoritmo de reconocimiento de gestos

Durante la fase de iniciación del proyecto se realizó una investigación cuya finalidad fue encontrar la solución a la implementación del algoritmo del reconocimiento de gestos estáticos. Para cada uno de los algoritmos candidatos seleccionados se determinaron las características de tipo de reconocimiento, oferta de servicio en la nube y clasificación por tags personalizados, las cuales se tuvieron en cuenta para efectuar la elección final. En la [tabla de comparación](#) se puede observar el detalle de clasificación y evaluación para cada una de las siguientes soluciones:

- *Basic Gesture Detection* [17]
- *Hand Gesture Detection AI with Convolutional Neural Networks* [18]
- *Microsoft Custom Vision Service* [19]
- *API Vision de Cloud* [20]
- *Amazon Rekognition Image* [21]

Conclusión

El algoritmo que se ajustó a las necesidades del proyecto fue *Microsoft Custom Vision* ya que el servicio computo que ofrece es muy completo. Este servicio permite crear conceptos propios como una seña y entrenarlos como imágenes de entrenamiento. También da la facilidad de exponer servicios REST para ser consumidos por el servidor de la aplicación del proyecto.

Por último, este servicio almacena una lista de predicciones realizadas donde luego pueden ser almacenados como imágenes de entrenamiento por medio de servicios REST, esto permite seguir entrenando el proyecto para obtener una predicción más precisa [22]

1.2. Aplicación (front-end)

La aplicación **LSC App**, fue desarrollada en lenguaje Java utilizando el IDE oficial Android Studio 3.1.2 y el Android SDK API 23, que soporta como mínimo dispositivos con el sistema operativo Android 6.0 (Marshmallow), lo que implica que la aplicación puede correr en aproximadamente 61.5% de los dispositivos móviles, según las tablas de distribución basadas en estadísticas de Google Play [23].

Android Architecture Components

Lanzadas durante el evento de Google I/O 2017 [24] basadas en el patrón MVVM, son un conjunto de librerías que sirven como recomendación oficial de buenas prácticas y arquitectura para construir aplicaciones robustas. Dado que son un conjunto de componentes que ofrecen múltiples servicios como persistencia e implementación de observables, se decidió utilizar esta librería y adoptar el patrón de diseño que promueve, reemplazando el patrón MVP inicialmente implementado. Los componentes utilizados para el proyecto fueron:

- **Lifecycle:** permite crear objetos que reaccionan automáticamente a cambios en el ciclo de vida de otros componentes Android, por ejemplo, los objetos que manejan los videos de la aplicación.
- **LiveData:** Clase que sirve para envolver la información en objetos observables lo que facilita la actualización de las pantallas y comunicación entre componentes.
- **ViewModel:** Clases independientes a los controladores de interfaz de usuario, permitiendo separar la lógica de la aplicación y la lógica de presentación
- **Room persistence Library:** sirve como capa de abstracción a la base de datos SQLite embebida en los dispositivos Android.

Comunicación con servidor a través de peticiones REST

Desarrollar servicios en Android que se comuniquen con un servidor a través de peticiones REST implica crear tareas asíncronas fuera del hilo principal de ejecución que hagan el llamado y manejen las peticiones y respuestas con el formato adecuado [25]. Para evitar este nivel de complejidad, se utilizó la librería Retrofit que facilita las tareas de conexión encapsulando el manejo de peticiones, cargue, hilos y sincronización [26]. El criterio de selección fue principalmente el factor de popularidad en la comunidad de desarrolladores y que se basa en las librerías OkHttp y Okio, también muy reconocidas en el medio.

Inyección de dependencias

Promover que las clases obtengan las instancias que necesitan a partir de una clase de configuración es el objetivo de la inyección de dependencias. Se utilizó en el proyecto ya que promueve el desacoplamiento, reutilización y facilita la ejecución de pruebas [27], Dagger 2 fue la librería escogida para este propósito, principalmente porque es la librería estándar utilizada mantenida por el equipo de Google [28].

Reproductor de video

Para esta tarea se utilizó la librería ExoPlayer respaldada por Google y utilizada en sus aplicaciones de YouTube y Play Movies [29], provee una alternativa a la opción nativa de Android, que facilitó la construcción del reproductor de video personalizado utilizado en la aplicación.

Gestión de Imágenes

El manejo de imágenes en Android es otra de las tareas que son complejas, ya que toca tener muchos factores en cuenta como, manejo de hilos, memoria y caches entre otros. Por lo cual para esta funcionalidad fue necesario utilizar otra librería, que de las investigadas se encontraron Glide y Picasso como las más utilizadas. Glide es mucho más poderosa ya que ofrece mayor personalización, pero para la aplicación del proyecto, requiere una gestión de imágenes sencilla, Picasso fue seleccionada por ser de menor tamaño [30].

1.3. Servidor (back-end)

1.3.1. Diseño de la arquitectura

Según la información consignada en [31], se realizó una comparación frente a una arquitectura orientada a microservicios, un monolito o una arquitectura guiada a servicios (SOA).

En la tabla 13 se presenta la comparación de las características de cada una de las correspondientes arquitecturas.

	Monolito	SOA	Microservicios
Independencia	Baja Todas las funcionalidades se encuentran bajo el mismo sistema	Media Los servicios se orquestan por medio de un ESB (Enterprise Service Bus)	Alta Las funcionalidades se comunican por protocolos ligeros
Acoplamiento	Fuerte No se tiene un concepto de servicio	Débil La lógica está separada en servicios	Desacoplado Cada servicio expone una funcionalidad de negocio específica e independiente
Despliegue	Lento Se debe desplegar el sistema completo	Lento Se deben desplegar todos los servicios y su correspondiente bus	Rápido Los servicios pueden ser desarrollados y desplegados por separado
Escalabilidad	Baja El sistema entero debe ser replicado	Baja Todos los servicios deben ser replicados	Alta Se puede replicar cada microservicio

Almacenamiento	Compartido	Compartido	Independiente
	El sistema tiene una misma unidad de persistencia	Si bien la lógica se separa por servicios, estos comparten los datos	Cada microservicio persiste su propia información

Tabla 13: Comparación de diseños arquitecturales.

Dado que para el proyecto se requerían las características contempladas anteriormente, se optó por realizar una arquitectura guiada por microservicios, ya que esta obtuvo mejor desempeño en comparación a las arquitecturas monolito y SOA.

Otras características por considerar fueron:

- La inclusión de una puerta de entrada *API Gateway*, encargada de recibir y redireccionar las peticiones a los correspondientes servicios en el *backend*.
- Los cambios en el servidor pueden darse sin afectar al o a los clientes.
- Cada microservicio puede ser desplegado de forma independiente, aceptando la realización de *roll back*, además de reducir los riesgos al momento de corregir *bugs*.
- No es necesario compartir la tecnología, debido a que se exponen servicios, ocultando la implementación de la lógica.

1.3.2. Spring y Mongo DB

Para el desarrollo del servidor se contempló el uso de los lenguajes de programación propuestos por AWS en [33], .NET, Java, Python, Ruby, PHP y NodeJS, todos ellos ofreciendo la posibilidad de crear servicios web tal como los microservicios requieren.

Se descartaron las opciones .NET, PHP y NodeJS, debido a la disposición de la arquitectura que propone el sistema operativo Ubuntu, la recomendación de usar un lenguaje orientado a objetos y la poca documentación y librerías, respectivamente.

Finalmente, teniendo en cuenta los *frameworks* Spring, Django y Ruby on rails, de los lenguajes de programación Java, Python y Ruby, respectivamente, se optó por el uso de Spring debido a:

- Experiencia y conocimiento previo del lenguaje de programación Java por parte del equipo de trabajo.
- Mayor facilidad en la configuración inicial, permitiendo iniciar el servidor de aplicaciones más rápidamente.
- Extensa documentación, provista por la comunidad.
- Tal como se requiere para cada microservicio, Spring presenta funcionalidades para cubrir peticiones web, seguridad y gestión de datos.

La idea de desarrollar una arquitectura guiada por microservicios promueve el uso de bases de datos no relacionales. Según MobgoDB [34] existen 4 tipos de bases de datos NoSQL (*Not only SQL*) que son:

- **Key-value stores (almacenamiento llave – valor):** cada elemento es almacenado con una llave única.
- **Wide-column stores (almacenamiento por columnas):** almacena múltiples datos en vez de filas.
- **Document databases (bases de datos por documentos):** enlaza una llave con un documento; cada documento puede contener almacenamiento llave – valor o almacenamiento por arreglos.
- **Graph databases (bases de datos por grafos):** almacenan información por redes.

Se optó por incluir una base de datos NoSQL basada en documentos. A partir de la información de Mongo DB [34], que posiciona a MongoDB como la base de datos por documentos más usada para el año 2018, se opta por utilizar dicha tecnología en conjunto con los microservicios.

Algunas ventajas de MongoDB incluyen:

- Relación con programación orientada a objetos, permitiendo fácil integración con Java, lenguaje de programación seleccionado para el desarrollo.
- Rendimiento alto para largos volúmenes de datos.
- No existe un esquema, es posible adicionar o remover cualquier cantidad de atributos en cualquier momento, de forma dinámica.

1.3.3. Amazon Web Services

Para los servicios de computación en la nube se tuvieron en cuenta 3 tecnologías; *Microsoft Azure*, *Amazon Web Services* y *Google Cloud*. Todas ofrecen múltiples servicios en la nube con posibilidad de personalizar varios de sus componentes.

Como se expresa en [35], de acuerdo con la habilidad de ejecución y a la completitud de los servicios ofrecidos, *Amazon Web Services* lidera la competencia, seguido por *Microsoft Azure* y *Google Cloud*. AWS ofrece IaaS (Infraestructura como Servicio) y PaaS (Plataforma como Servicio), lo que lo lleva a ofrecer posibilidades como instancias de almacenamiento, contenedores de servicios, computación con y sin servicios, entre otros.

En [AWS Cloud computing] se enuncia que AWS cuenta actualmente con 55 zonas de disponibilidad, en 18 distintas regiones del mundo, conectadas mediante redes de alta seguridad, desempeño y redundancia; razón por la cual sus servicios cuentan con alta disponibilidad y baja latencia.

Para el presente proyecto, AWS se presentó como la mejor opción para un sistema escalable, con alta disponibilidad y tolerancia a errores. Además, el servicio en la nube ofrece una capa gratuita por 12 meses, que incluye 750 horas para instancias EC2, 5GB de almacenamiento para S3 y miles de peticiones disponibles al mes, que en su conjunto son suficientes para el desarrollo y despliegue del servidor.

1.4. Administrador (front-end)

Angular 5

Angular usa el patrón MVVM que es útil para la creación de aplicaciones web altamente interactivas. Además, permite a los desarrolladores trabajar por separado en la misma sección de la aplicación utilizando el mismo conjunto de datos [36].

Angular utiliza la inyección de dependencia de las características relacionadas con los componentes con módulos y modularidad en general, esto permite que la página desarrolle por angular solo cargará los datos de módulo, en vez de recargar toda la página.

React JS

Es una biblioteca de *JavaScript*, abierta por Facebook en 2013, que es ideal para la construcción de grandes aplicaciones web donde los datos se pueden cambiar de forma periódica [36]. Expresa las estructuras HTML a través de JSX, las tendencias recientes también tienden a poner la administración de CSS dentro de *JavaScript*. Este enfoque tiene sus propios beneficios, pero en términos de estructurar el código no es tan conveniente para desarrollar [37].

React JS tiene falta de documentación oficial, el desarrollo súper rápido de *ReactJS* no deja lugar para la documentación adecuada. También que requiere de mucho tiempo para dominar, lo que significa que *ReactJS* requiere un conocimiento profundo de cómo integrar la interfaz de usuario en el marco de MVC [37].

Conclusiones

Además, para el desarrollo era ideal encontrar herramientas rápidas para el desarrollo, donde la curva de aprendizaje no fuera demasiada alta y además ofreciera servicios para comunicarse con el servidor de la aplicación. Por ese motivo se eligió *Angular 5* ya que es ideal para programadores con conocimiento a Programación Orientada a Objetos (POO) que utiliza estructuras detalladas fáciles de determinar.

2. Arquitectura de la aplicación

La solución desarrollada se divide en la aplicación móvil y el administrador web. Estas dos tienen como base un servidor central quien se encarga de manejar la lógica de negocio y almacenar el contenido de la aplicación. Cada una de las aplicaciones maneja una arquitectura propia de acuerdo a la tecnología en que fue desarrollada y manejan una comunicación entre ellas vía peticiones REST. A continuación se detalla cada una de las partes.

2.1. Arquitectura aplicación (front-end)

LSC App define su arquitectura basándose en la Guía de arquitectura de *Android Jetpack* [38] y las librerías de *Architecture Components*, cuya recomendación se refleja en el siguiente diagrama:

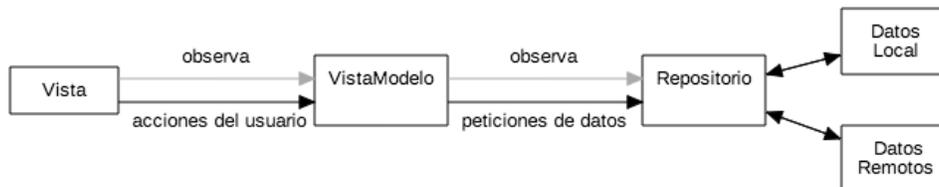


Figura 3: Aplicación móvil componentes arquitectónicos.

Esta arquitectura promueve la separación de responsabilidades la cual reflejamos en paquetes en nuestra aplicación:

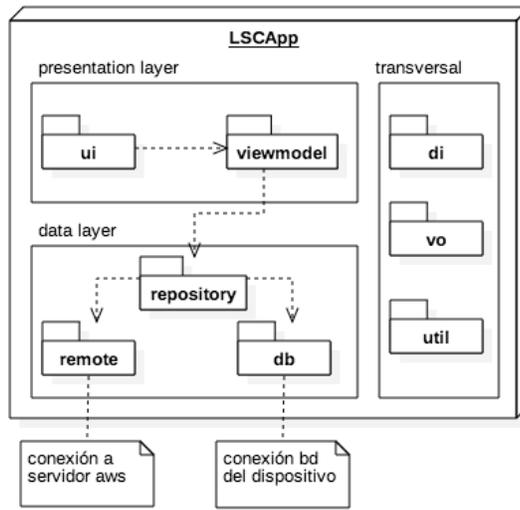


Figura 4: Paquetes principales en LSCApp.

Cada paquete cumple una responsabilidad específica la cual detallamos en el documento SDD de Android. A continuación, se muestra una descripción breve para cada uno:

Paquete remote

La fuente original de información de la aplicación proviene del servidor AWS, el cual ofrece servicios específicos para consulta y ejecución de acciones. Tomando como base estos, se definió una interface central que se encarga de hacer los llamados a dichos servicios

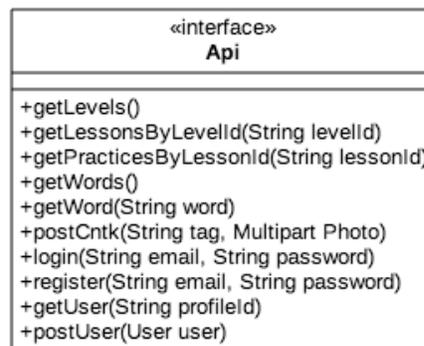


Figura 5: Definición de API de comunicación con servidor AWS.

Paquete db:

La información recolectada de las peticiones al servidor es almacenada en la base de datos del dispositivo la cual genera el siguiente modelo de entidades que pueden ser accedidas por el resto de la aplicación

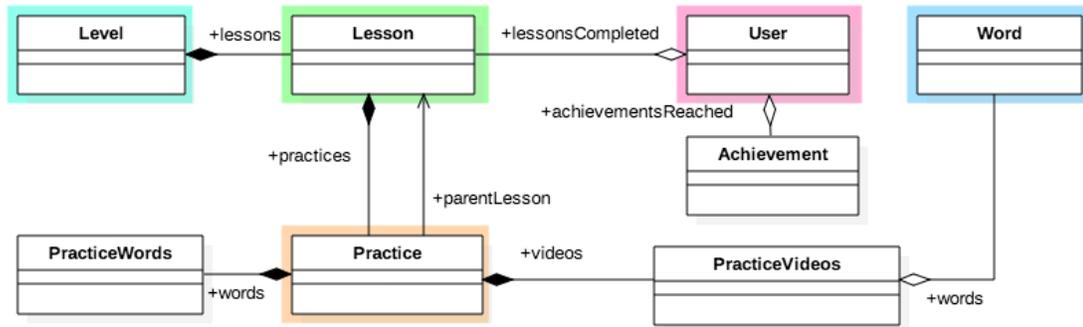


Figura 6: Entidades utilizadas en la aplicación.

Las entidades marcadas con colores son las entidades principales, las cuales representan secciones importantes en la aplicación.

Paquete repository

Se maneja una clase repositorio por cada una de las entidades principales de la aplicación y son el punto de entrada a la información de la aplicación que será mostrada al usuario. Sirve como *facade*, proveyendo la información solicitada y ocultando la gestión de la fuente. Contiene la lógica para determinar la fuente de datos, si proviene de la base de datos o del servidor aws.

Paquete viewmodel

Clases responsables de administrar información relacionada con la interface de usuario y capturar eventos del usuario, organizadas también por las entidades principales. Son intermediarios entre la interfaz y el repositorio utilizando objetos observables (*LiveData*), que encapsulan información de las entidades y notifican automáticamente a cualquier cambio

realizado. El flujo de comunicación en la aplicación entre la interfaz, viewmodels y repositorio se observa en el siguiente diagrama

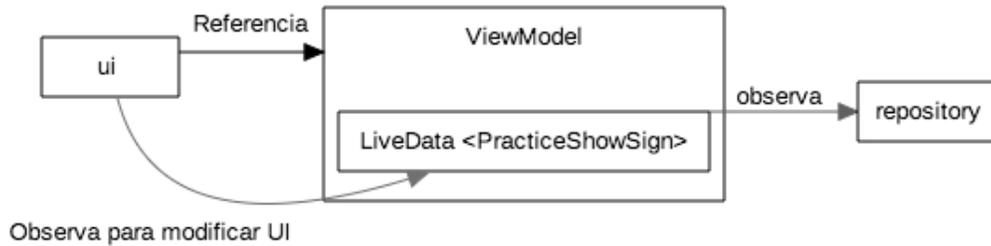


Figura 7: Flujo de comunicación entre componentes a través de observables.

Paquete ui

Maneja la interfaz de usuario y la navegación, **LSC App** utiliza varios de los componentes ofrecidos por el *framework* de Android. Basados en la utilización de estos, se clasificaron y se organizaron de la siguiente manera

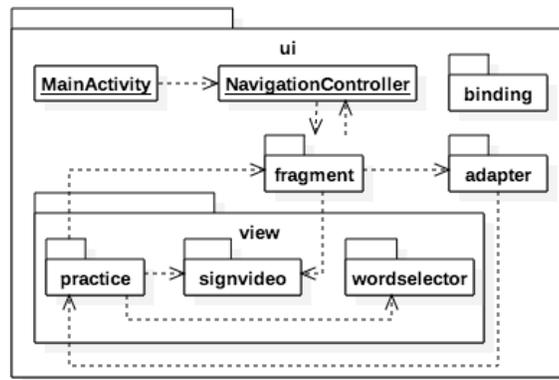


Figura 8: Paquetes principales ui

La aplicación utiliza el patrón de navegación por fragmentos⁹ por lo cual se utiliza una única actividad, denominada *MainActivity*, que muestra el menú principal y con el *NavigationController* dispone que fragmento mostrar según la acción del usuario. Los fragmentos de la aplicación tienen el contenido principal de la aplicación:

- Selección de nivel
- Selección de lección
- Desarrollo de prácticas según lección seleccionada
- Listado de palabras disponibles a consultar
- Formularios de iniciar sesión y de registro
- Perfil del usuario

A sí mismo el paquete *practice* contiene las vistas para las diferentes practicas definidas en la sección de ludificación.

2.2. Arquitectura servidor (back-end)

2.2.1. Componentes

En la figura 24 se ilustran los componentes definidos para el servidor. Se incluyen componentes externos como LSC Móvil y LSC Web, desarrollados por el equipo de trabajo, descritos en las **(secciones Arquitectura servidor (back-end))**.

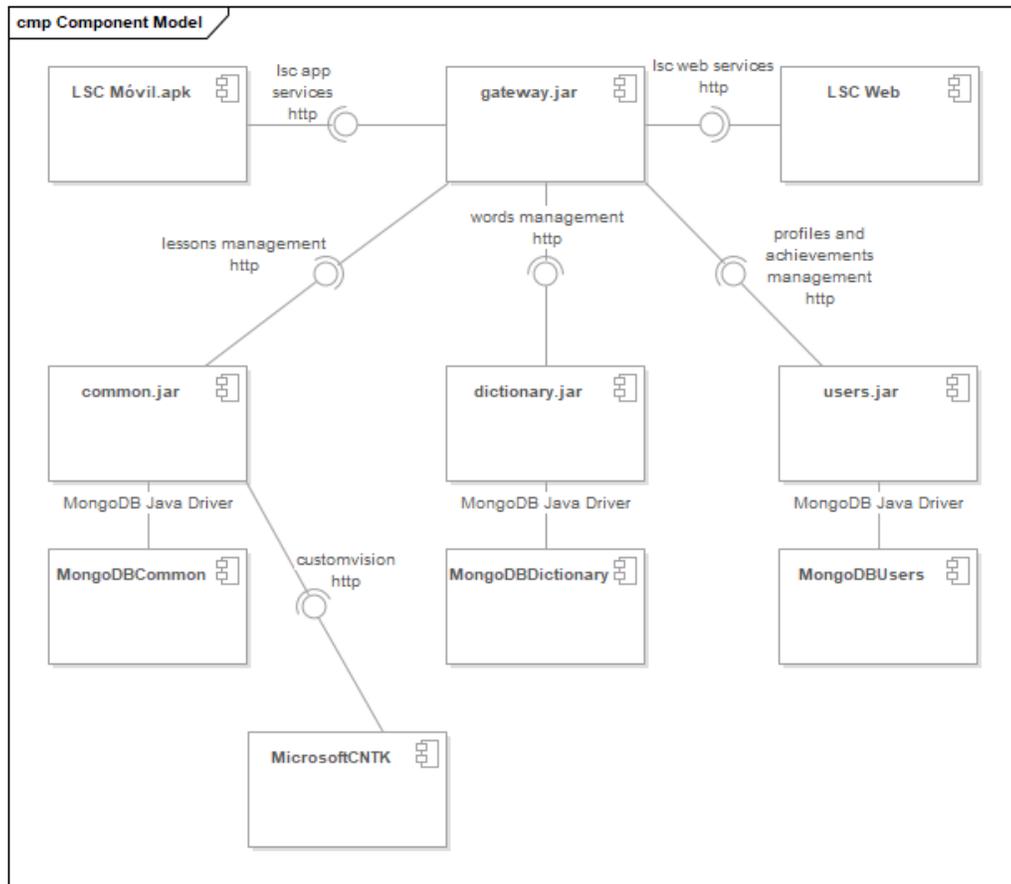


Figura 9: Diagrama de componentes servidor.

Para el desarrollo del servidor se propuso realizar una arquitectura guiada por microservicios, razón por la cual se diseñaron 4 componentes Java que gestionan la lógica e integración, y 3 bases de datos no relacionales para persistir los datos.

Los componentes Java definidos fueron:

- **Gateway**

Componente fachada encargado de recibir todas las solicitudes realizadas por medio de peticiones HTTP, a los servicios detallados en las interfaces expuestas *lsc app services* y *lsc mobile services*; y de redireccionar cada solicitud acorde a su descripción.

- **Common**

Gestiona la selección de niveles, lecciones y prácticas realizados por la aplicación, expuestos por medio de la interfaz *lessons management*.

Cabe resaltar que para el desarrollo de la práctica “*take-sign*”, definida en el documento de ludificación, solicita predicciones a los servicios expuestos por Microsoft, mediante su interfaz de *customvision*.

- **Dictionary**

Almacena y ofrece servicios en la interfaz *words management* para gestionar palabras del diccionario de señas, correspondientes a los niveles y lecciones propuestos.

- **Users**

Permite la gestión de perfiles y sesiones, además de la administración de logros, expuestos en la interfaz *profiles and achievements management*.

Las comunicaciones entre los componentes Java y con sistemas externos, incluyendo LSC Móvil y LSC Web son por medio de peticiones REST.

Adicionalmente, los componentes MongoDB, comunicados por medio de MongoDB Java Driver, se encargan de la persistencia y obtención de entidades, almacenando las clases de la siguiente manera:

- **MongoDBCommon:** *LessonEntity* y *LevelEntity*.
- **MongoDBDictionary:** *WordEntity*.
- **MongoDBUsers:** *AchievementEntity* y *ProfileEntity*.

En el anexo [SDD](#) se detallan los paquetes, clases y conexiones entre clases por cada componente.

2.2.2. Despliegue

De acuerdo con la figura 25, se especifica la relación de los componentes correspondientes a la estructura física del sistema. Dichos componentes se encuentran alojados dentro de nodos, los cuales referencian plataformas de hardware de la aplicación.

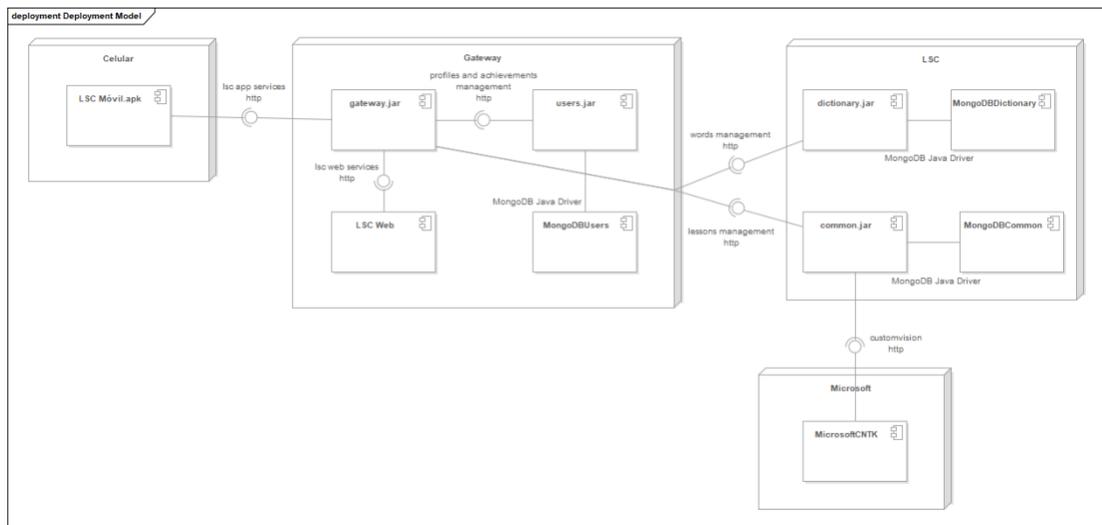


Figura 10: Diagrama de despliegue.

El sistema está compuesto por 4 nodos, 3 propios y 1 externo; dividiendo el servidor en 2 máquinas, además del nodo que hace referencia al dispositivo móvil. La configuración de los nodos del servidor es explicada en la (sección **Amazon Web Services**). El 4º nodo hace referencia a una máquina perteneciente a Microsoft, que nos permite la resolución de peticiones de aprendizaje de máquina provistas por la interfaz *customvision*.

De igual forma en la tabla 14 se especificaron las siguientes características correspondientes a cada nodo: procesador, memoria RAM, sistema operativo y disco duro.

Nodo	Procesador	RAM	Sistema operativo	Disco duro
Celular	Snapdragon 410 1.2GHz	2GB	Android 6.0	16GB
Gateway y	Intel Xeon Family 2.5GHz	1GiB	Ubuntu 16.04	25GB
LSC	Intel Xeon Family 2.5GHz	1GiB	Ubuntu 16.04	25GB

Tabla 14: Especificación de nodos.

Las características de los nodos Gateway y LSC son como se encuentran actualmente desplegados, las del celular son características mínimas del dispositivo, en el que se aseguró el funcionamiento de la aplicación.

2.3. Arquitectura del administrador (front-end)

El administrador web consume servicios RESTful del servidor de la aplicación para realizar las operaciones CRUD [40]. Como el administrador web fue desarrollado con el *framework* de Angular 5, este *framework* posee un servicio denominado Angular CLI que permitió crear un proyecto base donde se desarrolló la lógica de la página web [39].

Los bloques de construcción básicos de usados son los NgModules, que proporcionan un contexto de compilación para los componentes. NgModules recopilan código relacionado en conjuntos funcionales; toda la aplicación está definida por NgModules 41. Aplicación Angular tiene un módulo raíz que habilita el arranque, este tiene otros módulos, en el caso del Administrador el módulo raíz se conecta con el módulo *components* como se ve en la figura 26.

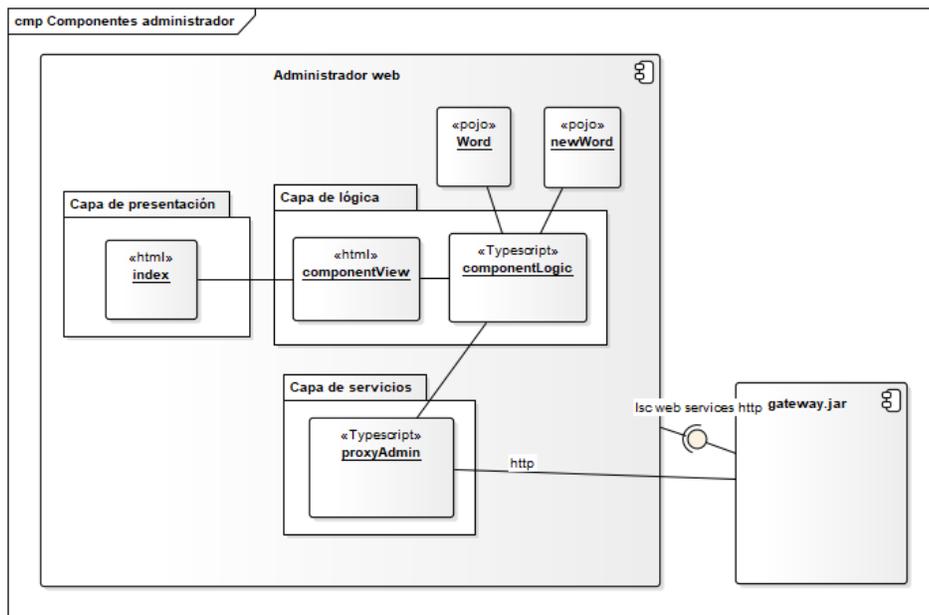


Figura 11: Diagrama de componente del administrador.

Dentro de ese componente se define la vista del administrador, en *task.component.html*. Estos elementos son conjuntos en la pantalla que Angular puede elegir y modificar.

Los metadatos de una clase componente (*task.component.ts*) lo asocian con una plantilla (*task.component.html*) que define una vista [39].

El administrador tiene el componente denominado *services* que proporciona la funcionalidad de proveer servicios, estos se pueden inyectarse en componentes como dependencias, haciendo que su código sea modular, reutilizable y eficiente. En el caso del administrador se usan servicios con operaciones CRUD.

Tanto los componentes como los servicios son simplemente clases, con decoradores que marcan su tipo y proporcionan metadatos que le dicen a Angular cómo usarlos. Los componentes hacen uso de las clases *Work* para guardar las palabras enlistadas y *newWork* para crear una palabra y enviarla al servidor de la aplicación.

VI- DESARROLLO DE LA SOLUCIÓN

1. Metodología

Para este proyecto se utiliza una metodología de investigación y DAD (*Disciplined Agile Delivery*) como marco de referencia. En su definición, DAD se propone como un *framework* híbrido que adopta estrategias de otros métodos ágiles como *Scrum*, *Extreme Programming (XP)*, *Agile Modeling (AM)*, *Kanban* entre otros [8]. A continuación, se muestra un diagrama que describe las fases del proyecto:

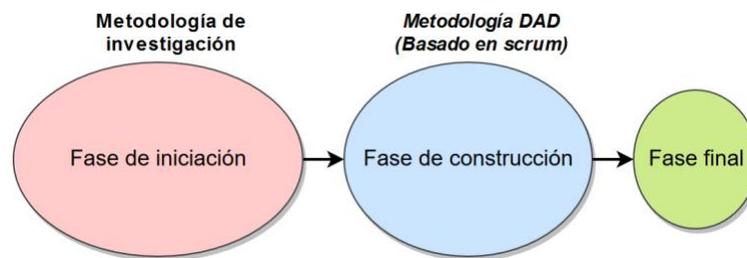


Figura 12: Metodología de investigación y desarrollo.

1.1. Fase de iniciación

Esta fase pretendió plantear, evaluar y corregir la problemática y los problemas a abordar, para esto, se realiza una búsqueda de información y una elaboración de la propuesta. Posteriormente, se realiza la planeación del proyecto, la investigación de tecnologías y herramientas a utilizar, para realizar la especificación de requisitos y el diseño de las pruebas.

Actividades

- **Planteamiento del problema:** Fue la que definió la problemática y los problemas que se quieren resolver.
- **Definir propuesta de trabajo:** Se realizó la investigación acerca de la problemática y los problemas, trabajos similares, y conceptos relevantes. Se plantea una solución y su justificación, el alcance y las restricciones para tener en cuenta.

Nota: Después de esto se realizaron varias reuniones cada 2 semanas, corrigiendo y mejorando la propuesta inicial hasta tener una propuesta mejor elaborada.

Entregables:

✓ Propuesta del trabajo de grado, [Ver anexo](#).

- **Elaboración del marco teórico:** En esta actividad se investigó conceptos relevantes, trabajos similares en el área a abarcar y posibles soluciones o soluciones parciales ya desarrolladas.
- **Evaluar el valor potencial:** En esta actividad el grupo se reunió constantemente para discutir el impacto de la solución ante la problemática y los problemas planteados, se analiza si se da solución, y si esta tiene un impacto suficiente.
- **Precisar restricciones del sistema:** Para ello se estableció el alcance, los supuestos y las restricciones sobre las que se realizará el trabajo de grado, la planeación, la implementación y la fase final.
- **Planeación del proyecto:** Se realizó un documento de planeación, que contiene los objetivos, la justificación, los recursos, el cronograma de las actividades y los entregables, la metodología de la implementación y los lenguajes y herramientas a utilizar.

Entregables

✓ Plan de Administración del Proyecto de Software, [ver anexo](#).

- **Selección del algoritmo:** En esta etapa buscó y evaluar algoritmos de reconocimiento de imágenes que identifique gestos de la mano de acuerdo con las restricciones de la implementación.
- **Técnicas de ludificación:** Etapa de análisis de implementaciones de algoritmos de reconocimiento de gestos adaptables de acuerdo con las restricciones de la implementación. Se establece cómo incorporar técnicas de ludificación en el producto funcional.

Entregables

- ✓ Algoritmo de reconocimiento de gestos, (**sección Algoritmo de reconocimiento de gestos**).
- ✓ Instructivo de técnicas de ludificación, (**ver sección Instructivo de técnicas de ludificación**).
- **Especificación de requisitos:** En esta etapa se realizaron requisitos, después de la elaboración se pasaron a historias de usuario como una descripción más precisa orientada al desarrollo. Estas historias de usuario se crearon y validaron apoyándose de los *mockups*.

Entregables

- ✓ Historias de usuario
- ✓ Casos de uso
- ✓ *Mockups*

Diseño de las pruebas

Desarrollo de las pruebas de la implementación requeridas para validar la arquitectura y el producto final, además, se establecen las pruebas a realizar después de la fase de construcción para determinar el impacto de la solución.

Entregables

- ✓ Documento de control de calidad, [ver anexo](#).

En la figura 28 Se puede apreciar el orden de desarrollo de cada tarea.

Metodología de investigación

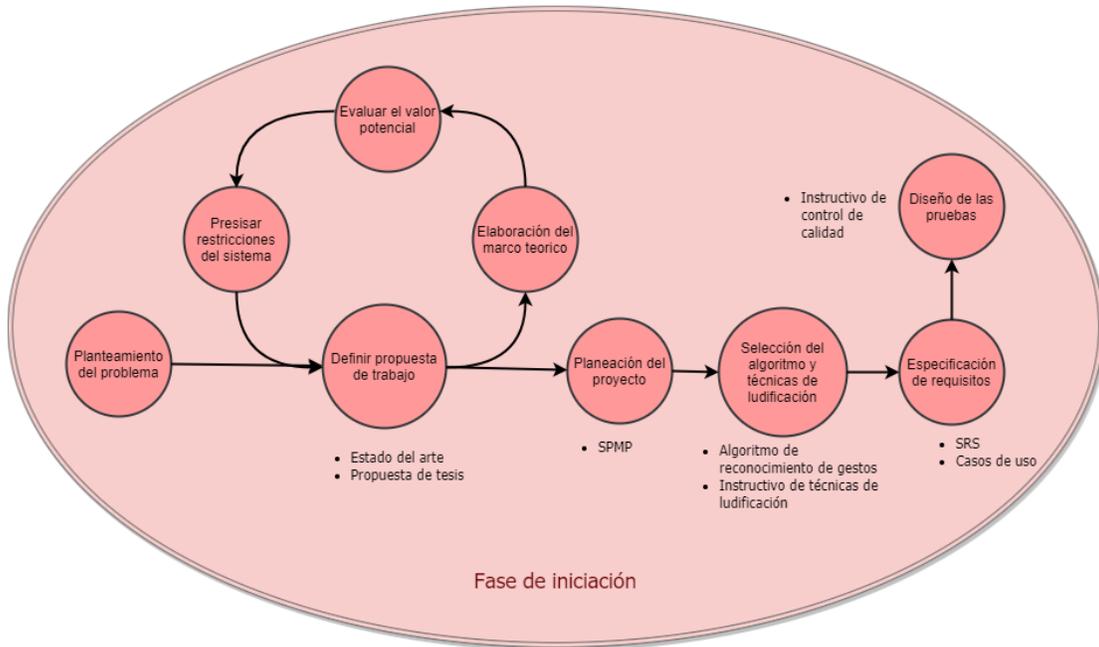


Figura 13: Fase de iniciación.

1.2. Fase de construcción

En esta fase se utiliza el *framework* DAD basado en conceptos de desarrollo ágil *Scrum*, esto consistió en hacer reuniones cada 2 semanas (*sprints*) para asignar las responsabilidades trabajo donde se investigó, diseñó, se hizo pruebas de conexión de la arquitectura e implementó la solución.

Actividades

- **Diseño de la arquitectura:** En esta actividad se asignaron tareas para el bosquejo y en consecuencia la elección de tecnologías que conformen la arquitectura describiendo sus funcionalidades e interacciones entre sus componentes como:
 - *Back-end* servidor y otros sistemas externos
 - *Front-end* móvil
 - *Front-end* Administrador Web

Entregables:

- ✓ Descripción del Diseño del Software, [ver anexo](#).

- **Elaboración del diseño detallado:** Se realizó el diseño de la aplicación móvil, y su relación con los *mockups* teniendo en cuenta las técnicas de ludificación descritas, obteniendo diagramas como:
 - Diagrama de clases.
 - Diagrama de persistencia.

- **Planeación de la iteración:** EL grupo realizó reuniones cada dos semanas con la directora de grado para distribuir tareas y planear los objetivos del *sprint*, teniendo en cuenta la lista de actividades descritas en el SPMP, asignando responsabilidades y fechas de entrega de cada actividad a cada miembro del equipo.

Entregables

- ✓ Acta de reunión, [ver anexo](#).
 - ✓ *Product backlog*, [ver anexo](#).
- Validación y verificación:** En esta actividad se completó un documento de control de calidad para la aplicación haciendo una serie de pruebas para validar la funcionabilidad y verificar el manejo de la información dentro del sistema de la aplicación.

Entregables

- ✓ Documento de control de calidad

En la Figura 29 Se puede apreciar el orden de desarrollo de cada tarea.

Metodología DAD (Basado en Scrum)

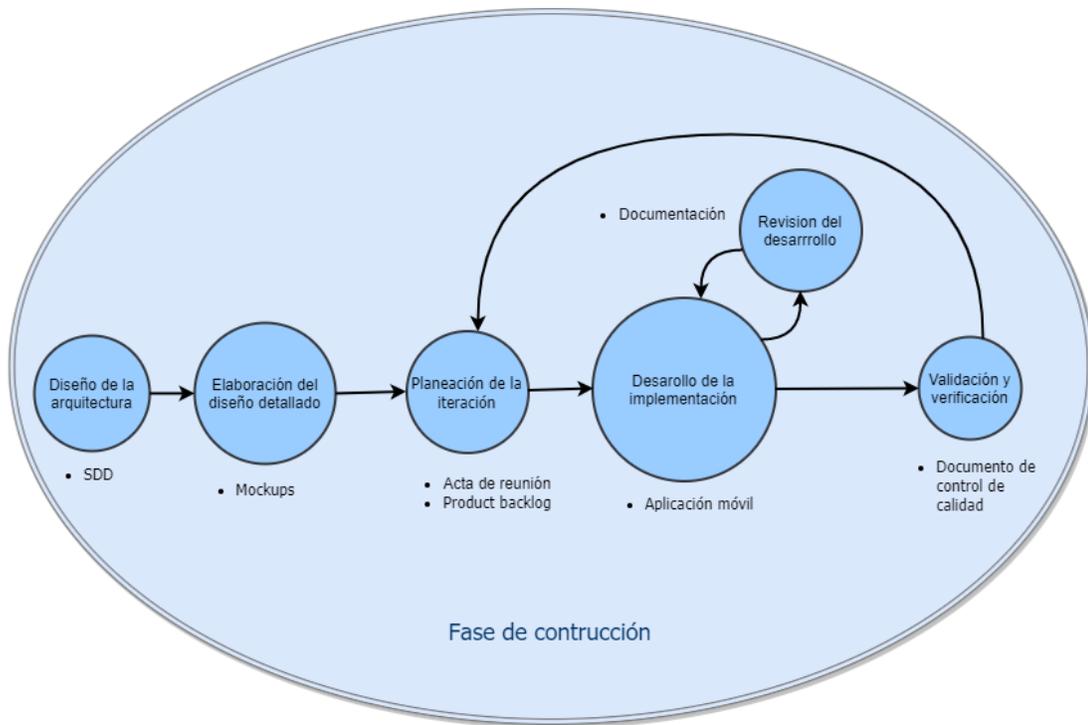


Figura 14: Fase de construcción.

1.3. Fase de final

En esta fase se realizó un seguimiento completo a la versión final de la aplicación de la mano de las pruebas en el control de calidad. Además, se realizó actividades relacionado a la documentación y pruebas de usuario.

Actividades

- **Lanzamiento y pruebas de impacto:** En esta etapa, después de haber tenido la aplicación y el servidor probado en su totalidad se creó una versión beta de la aplicación para luego pasar a la actividad de pruebas de usuario y aceptación (**sección Pruebas de usuario**). Y luego se realizó la tarea de añadirlo a la tienda Google Play Store siguiendo las instrucciones de carga del APK.

Entregables:

- ✓ Informe de valoración, [ver anexo](#).
- **Post mortem:** En esta etapa del proyecto el grupo se reunió para hacer revisiones de la documentación y de los anexos que la componen. Sacando conclusiones y organizando todos los elementos para estar preparados en el día de la sustentación.

Entregables:

- ✓ Memoria de trabajo de grado.

En la figura 30 Se puede apreciar el orden de desarrollo de cada tarea.

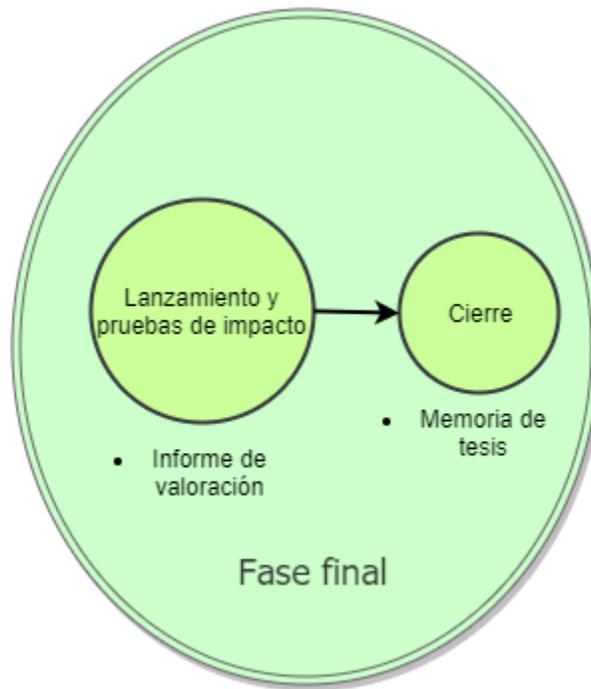


Figura 15: Fase final.

2. Producto final

2.1. Aplicación (front-end)

Al iniciar la aplicación por primera vez, se solicita al usuario que se inicie sesión o que se registre, presentando el formulario respectivo. Se implementaron validaciones básicas en los campos, como correcta escritura de correo electrónico.

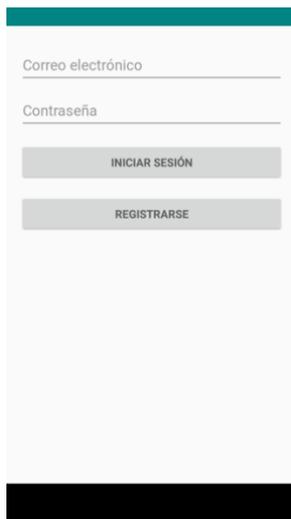
Este formulario de inicio de sesión presenta un campo de texto para el correo electrónico y otro para la contraseña. Debajo de los campos hay dos botones: 'INICIAR SESIÓN' y 'REGISTRARSE'.

Figura 16: Formulario iniciar sesión.

Este formulario de registro incluye campos para el correo electrónico, la contraseña y un campo para confirmar la contraseña. Los botones 'REGISTRARSE' e 'INICIAR SESIÓN' están situados al final del formulario.

Figura 17: Formulario registrarse.

Esta captura de pantalla muestra un mensaje de error 'Este correo es incorrecto' en un recuadro negro que aparece sobre el campo de correo electrónico. El correo 'correomalformado.com' está resaltado en rojo con un ícono de advertencia.

Figura 18: Validación de campos.

Al registrarse o iniciar sesión, la aplicación guarda en la base de datos el id provisto por el servidor, de tal modo que cuando vuelva a ingresar no tiene que realizar este paso de nuevo.

Luego de la autenticación exitosa, la aplicación muestra la actividad principal, que permite acceder al menú que dan acceso a los fragmentos principales:

- Selección de nivel, mostrado por default al iniciar aplicación.
- Diccionario.
- Perfil.



Figura 19: Selección de nivel.

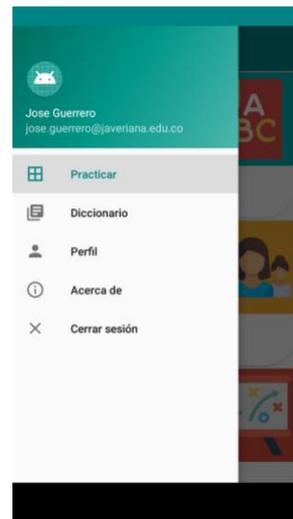


Figura 20: Menú principal.

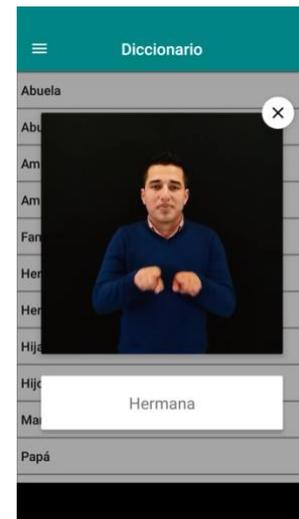


Figura 21: Diccionario.

Al seleccionar un nivel se puede observar el contenido de cada una:



Figura 22: Lecciones en nivel Abecedario y números.



Figura 23: Lecciones en nivel Sustantivos.



Figura 24: Listado de lecciones en nivel Predicados.

Al seleccionar una lección se muestra el listado de prácticas que contiene. Cada práctica se muestra una a la vez y el usuario debe presionar el botón siguiente para proseguir. Para las

prácticas de tipo evaluación el botón de siguiente está inactivo hasta que el usuario haya seleccionado una respuesta. Adicionalmente en la parte superior se muestra la barra de progreso que indica al usuario la posición en que se encuentra:

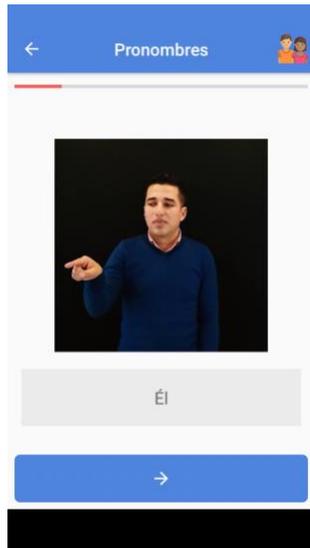


Figura 25: Práctica show-sign.

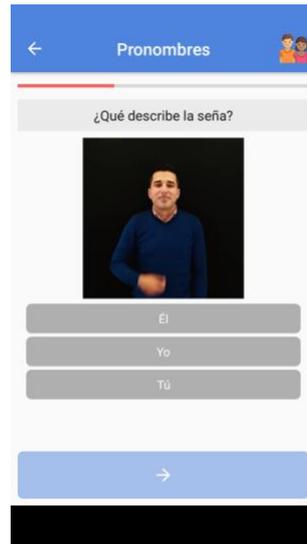


Figura 26: Botón de siguiente deshabilitado.



Figura 27: Botón siguiente habilitado.

Al presionar siguiente, se evalúa la respuesta y se da la retroalimentación, indicando la opción correcta en caso de fallo.

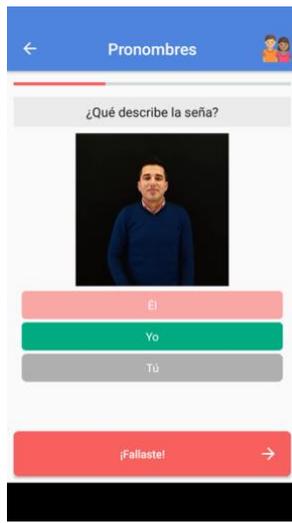


Figura 28: Caso de fallo.

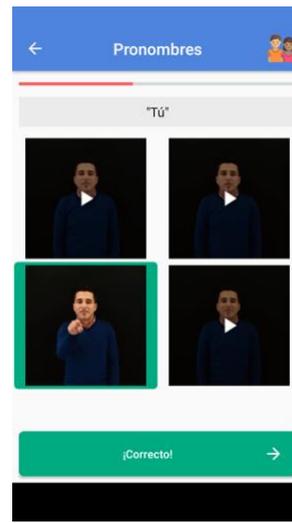


Figura 29: Caso exitoso.

La misma mecánica se repite en los demás tipos de prácticas:



Figura 30: Práctica translate-video.

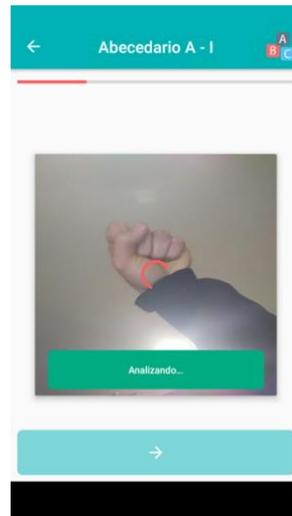


Figura 31: Práctica take-sign.



Figura 32: Práctica discover-image.

Al finalizar todas las prácticas de una lección se da retroalimentación del logro obtenido



Figura 33: logro obtenido al finalizar práctica.

2.2. Servidor (back-end)

2.2.1. Despliegue en AWS

AWS LSC

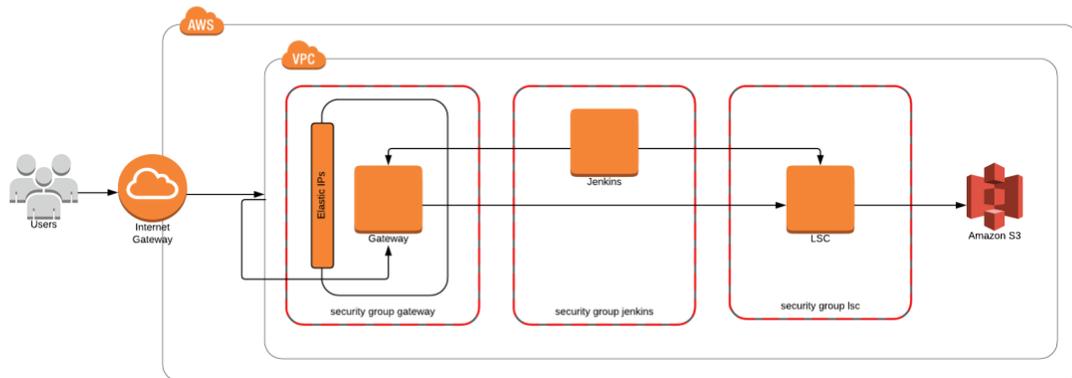


Figura 34: Despliegue en AWS.

La figura 49 representa la secuencia de una petición realizada por un usuario, al momento de formar parte de la red de conexiones Internet, pasando a través de los servicios *Amazon Web Services (AWS) Cloud*, dentro de una *Virtual Private Cloud (VPC)*, hasta llegar a los nodos EC2 desplegados como *Gateway* y LSC.

Teniendo en cuenta la información especificada en [33] los componentes evidenciados en la figura 49 hacen referencia a:

- **Internet gateway:** permite la comunicación entre las instancias alojadas en la VPC y la internet.
- **VPC:** red privada virtual que permite alojar recursos y servicios AWS con beneficios de escalabilidad.
- **Security group:** controla el tráfico de una o más instancias, asociando uno o más nodos a un grupo, además de definir reglas para restricciones de entrada y salida de peticiones por grupo.
- **EC2:** servicio web que provee capacidad de cómputo y almacenamiento en la nube, ofreciendo control completo sobre los recursos.
- **Elastic IP:** IPv4 estática, alcanzable desde internet, asociada a una instancia EC2.
- **Amazon S3:** proporciona servicios de recopilación, almacenamiento y análisis de datos en la nube.

Para el diseño se propuso relacionar:

- 1 *Internet Gateway*: ofrecido y configurado por AWS.
- 1 VPC: ofrecida y configurada por AWS.
- 3 instancias EC2 correspondientes a:
 - nodo del servidor *Gateway* con 1 IP elástica (18.208.61.61),
 - nodo del servidor LSC,
 - y nodo asociado a Jenkins.
- 3 *security groups* permitiendo el acceso a las instancias por los puertos descritos en la tabla 15, además del acceso por el puerto 22 relacionado a SSH (*Secure Shell*).

- 1 instancia de Amazon S3: almacena el contenido multimedia de la aplicación y cuenta con un espacio de 5GB, del que actualmente se utiliza menos de 50MB.

Todos los servicios se encuentran en la región US East, con zona de disponibilidad North Virginia, debido a su costo a comparación de los demás.

Instancia	Componente	Puerto	Fuente
Gateway	Gateway	12345	0.0.0.0 Cualquiera
	Users	12346	Gateway
	MongoDBUsers	12356	No aplica
LSC	Common	12347	Gateway
	MongoDBCommon	12357	No aplica
	Dictionary	12348	Gateway
	MongoDBDictionary	12358	No aplica
Jenkins	Jenkins	8080	0.0.0.0 Cualquiera

Tabla 15: Mapeo de puertos por componente.

2.2.2. Secuencia de despliegue

Para realizar el despliegue del servidor en los servicios de cloud ofrecidos por AWS, se sigue el proceso ilustrado en la figura 50. Se tienen en cuenta los componentes de un sistema de repositorios centralizado, un sistema de integración y despliegue continuo, Jenkins, además de las instancias donde será desplegado el servidor.

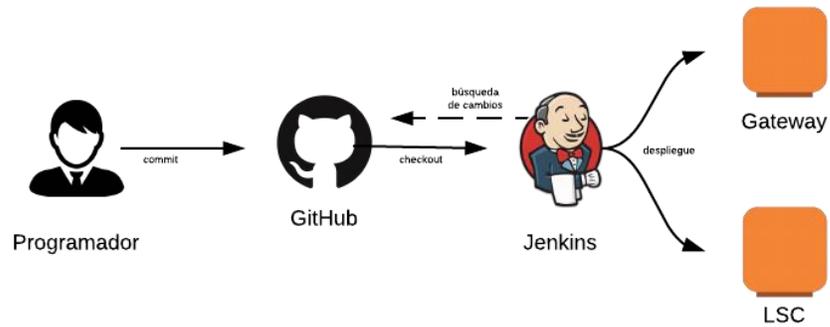


Figura 35: Secuencia de despliegue.

La figura 50 ilustra el proceso desde que el programador realiza cambios sobre un repositorio local, sube los cambios a un repositorio centralizado, GitHub para el presente proyecto; y al mismo tiempo, Jenkins escucha cualquier cambio realizado sobre la rama Master de git, y de haber algún cambio obtiene todos los datos actualizados.

Jenkins compila todos los proyectos java utilizando Maven, genera todas las unidades de despliegue .jar y las copia en las instancias Gateway y LSC dependiendo de las instancias descritas en la tabla 15. Finalmente, Jenkins finaliza todos los procesos Java y Mongo de las instancias y ejecuta las nuevas unidades de despliegue.

2.2.3. Peticiones

Los servicios fueron expuestos para ser accedidos por medio de peticiones REST, razón por la cual, desde la aplicación, un ambiente de desarrollo para API (*Application Programming Interface*) o un navegador, es posible acceder a los mismos.

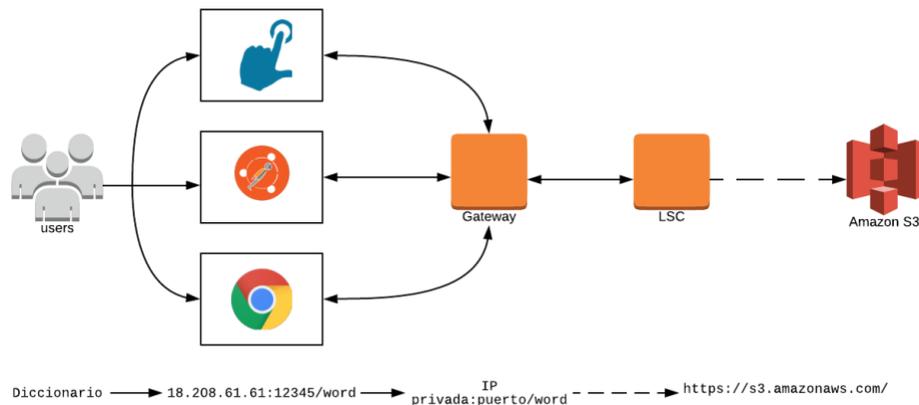


Figura 36: Ejemplo de petición REST.

En la figura 51 se muestra el flujo desde que un usuario o administrador realiza una petición para obtener el diccionario de señas hacia el *Gateway*, encargado de redirigir la petición al servidor LSC, que contiene el microservicio que gestiona las palabras. Finalmente, la petición es respondida con una lista de palabras que contienen videos y palabras asociados por palabra, almacenados en una instancia S3.

2.3. Administrador web (front-end)

El administrador web es una funcionalidad agregada no prevista en la planeación del proyecto pero que surgió luego de descubrir que al hacer extensible la aplicación, puede permitir cambios y promover su crecimiento a futuro [42]. En esta primera versión del administrador, se implementó un CRUD sencillo de los videos y palabras que se utilizan en la aplicación, por lo que tiene funciones de actualizar, eliminar o añadir. La página del administrador se conecta al servidor de la aplicación por medio de servicios RESTful usando métodos CRUD tales como POST, GET, PUT y DELETE [40].

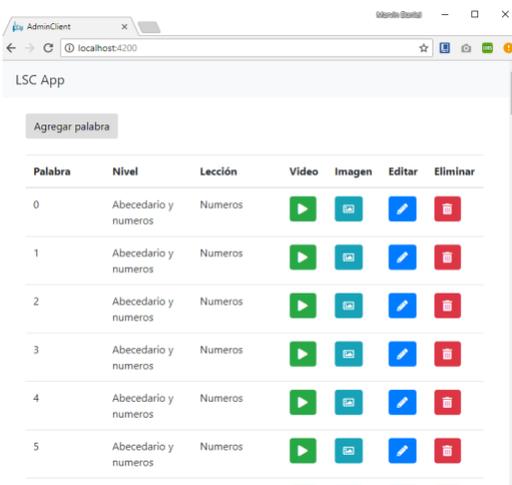


Figura 37: Administrador web recibir datos.

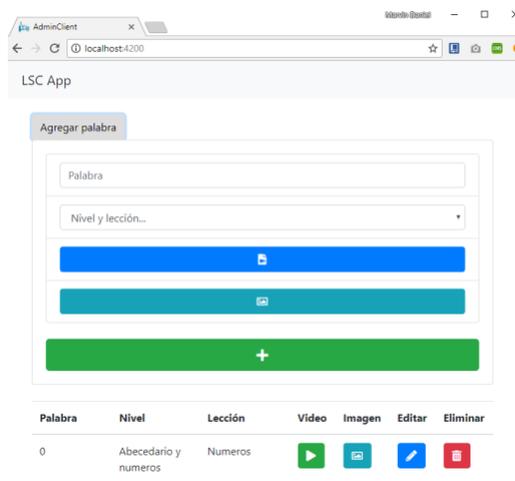


Figura 38: Administrador web agregar video.

VII- RESULTADOS

1. Control de calidad

1.1. PRUEBAS DE LA APLICACIÓN

1.1.1.1. Pruebas de instrumentación

En la aplicación móvil, para probar la interfaz de usuario, se implementaron pruebas instrumentadas utilizando la librería Espresso. Este *framework* permitió programar de manera automática, simulaciones de usuario en un dispositivo real para validar el correcto funcionamiento de la aplicación. Las pruebas se organizaron según las entidades principales de la aplicación y las acciones que el usuario realiza sobre ellas. Cada vez que se ejecutan las pruebas Espresso genera un reporte que detalla el estado final de las pruebas, el cual puede ser consultado en el siguiente enlace ([Reporte pruebas instrumentación](#)). Para ver la descripción de las pruebas generadas ir a [Documento de Control de Calidad](#).

1.1.1.2. Pruebas de usuario

La prueba fue diseñada con el propósito de evaluar el producto y conocer tendencia de aprobación de los usuarios. La encuesta se dividió en 2 etapas que median la aceptabilidad y el nivel de satisfacción al momento de usar la aplicación.

El proceso de la prueba consistió en mostrarle el celular con la aplicación instalada al usuario, después que el usuario detallará la aplicación se le pidió que llenara una encuesta. Con el fin de alcanzar el mayor número de individuos se les envió un video demostrativo presentando el funcionamiento de la aplicación, de igual modo se les pedía que llenaran la encuesta.

Esta prueba fue realizada a 38 usuarios al azar los cuales fueron divididos por 3 criterios que determinaban el conocimiento del LSC. Se encontró que 2 personas lo conocían muy bien, mientras 17 personas tenían cierto conocimiento de la lengua y 19 personas no sabían nada. Entonces se puede afirmar que pocas personas conocen del todo el LSC.

¿Que tantas señas del LSC conocía antes de ver la aplicación?

38 responses

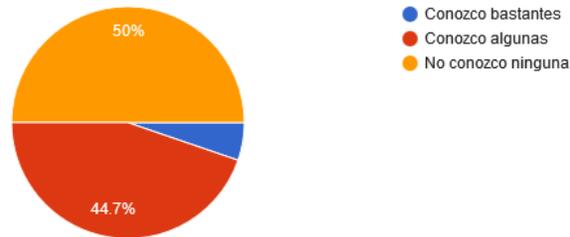


Figura 39: Distribución de personas que conocen el LSC.

1.1.1.2.1. Pruebas de usabilidad

La sección de usabilidad basada en QUIS 7.0 [43] consistió en adaptar 14 preguntas divididas en 5 partes que sirvieron para determinar la satisfacción del cliente al momento de usar o visualizar la aplicación. Las partes seleccionadas fueron:

Interfaz de la Aplicación		1	2	3	4	5	
La interfaz en general le pareció	Nada atractiva	0	0	0	16	16	Muy atractiva
La interacción con la aplicación le pareció	Nada predecible	0	1	6	16	15	Predecible

Tabla 16: Resultados interfaz de la aplicación.

Como se muestra en la tabla 16 indica que el diseño de la aplicación fue agradable a la vista de las personas e intuitiva para usar.

Elementos de la aplicación		1	2	3	4	5	
La fuente (texto) en la pantalla le pareció	Difícil de leer	0	0	2	16	20	Fácil de leer
La cantidad de colores disponibles le pareció	Inadecuada	0	1	3	17	17	Adecuada

Tabla 17: Resultados elementos de la aplicación.

Además, en la tabla 17 las personas no presentaron molestias en la lectura del texto y a la cantidad de colores ya que el diseño de la aplicación desde los *mockups* quiso cumplir con un diseño estético y sencillo.

Terminología e información del sistema		1	2	3	4	5	
La cantidad de información por pantalla le pareció	Muy confusa	0	0	4	17	17	Muy adecuada
Los términos y palabras usadas en la aplicación le parecieron	Nada entendible	0	1	1	16	20	Muy entendible

Tabla 18: Resultado terminología e información del sistema.

Aprendizaje		1	2	3	4	5	
Aprender a utilizar la aplicación le pareció	Difícil	0	0	1	13	24	Fácil
Explorar la aplicación por completo le pareció	Difícil	0	1	1	13	23	Fácil
Recordar los nombres y las instrucciones de la aplicación le pareció	Difícil	0	1	2	13	22	Fácil

Tabla 19: Aprendizaje.

Según la tabla 18 y tabla 19 las personas logran adaptarse fácilmente al uso de la aplicación.

Contenido		1	2	3	4	5	
La calidad de ilustraciones y fotografías le pareció	Mala	0	0	0	11	27	Buena
Los juegos de ludificación le parecieron	Nada entretenidos	0	0	3	14	20	Entretenidos
La calidad del vídeo de la aplicación le pareció	Mala	0	0	3	11	24	Buena

Tabla 20: Resultado contenido.

Para el desarrollo del entorno lúdico se buscó que los usuarios se mantuvieran entretenidos, según la tabla 20 se puede ver resultados favorables al momento de la elección de contenido.

1.1.1.2.2. Pruebas de aceptación

En esta sección se encontró el impacto en general que los usuarios tuvieron después de usar a aplicación, que arroja resultados favorables como se observa en la tabla

Pruebas de aceptación		1	2	3	4	5	
La idea de una aplicación para conocer el LSC le parece	Muy mala	0	0	1	11	26	Muy buena
El contenido de la aplicación le pareció	Nada estimulante	0	0	3	16	19	Muy estimulante
El propósito de la aplicación le pareció	Poco claro	0	0	2	5	31	Claro
¿Qué tanto impacto cree que la aplicación tenga en Colombia?	Poco impacto	1	0	10	15	12	Mucho impacto
¿Qué tanto impacto cree que tuvo la aplicación en usted?	Poco impacto	0	0	8	14	16	Mucho impacto

Tabla 21: Resultados de pruebas de aceptación.

Adicionalmente se encontró que la mayoría de los usuarios están satisfechos con el contenido de la aplicación, con una excepción de un 21.1% que recomienda: Usar más tipos de juegos, Poner los verbos conjugados e involucrar otro tipo de vocabulario cotidiano, como WhatsApp, comidas y otros. Estas recomendaciones se tomaron en cuenta en la (sección Trabajo futuro).

También se utilizó el NPS (*Net Promotor Score*) que es un indicador que sirve para medir la lealtad de los clientes respecto a un producto [44].

- Promotores: quienes dan entre 9 y 10 puntos.
- Pasivos: los que dan entre 7 y 8 puntos.
- Detractores: los que dan puntuaciones entre 0 y 6 puntos.

¿Recomendaría esta aplicación a un familiar o amigo?

38 responses

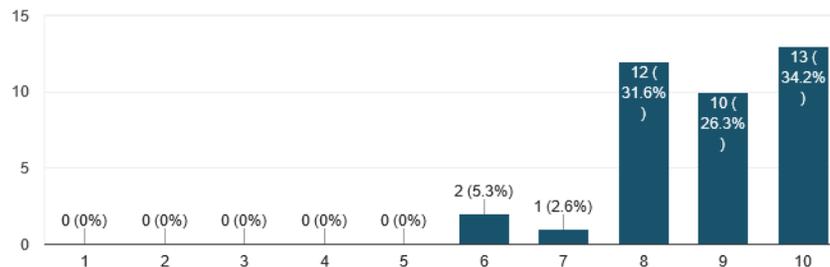


Tabla 40: Resultado de encuesta NPS.

De los resultados obtenidos se puede concluir que los usuarios potenciales de la aplicación serían los usuarios promotores y pasivos quienes están satisfechos con el producto usado. Muchos de los encuestados comentaron:

- “Muy novedosa, quizá más adelante incorporar otras funciones, pero por el momento me parece una grandiosa idea, teniendo en cuenta la falta de información que se tiene sobre LSC.”
- “Me gustó la aplicación. Obvio el uso se reduce a las allegadas de personas con problemas de audición, sin embargo, la idea me pareció muy buena y yo las descargaría.”

1.2. PRUEBAS DEL SERVIDOR

1.2.1. Pruebas de API

Teniendo en cuenta los conceptos básicos planteados en [45] sobre BDD (*Behavior Driven Development*), el servidor es desarrollado a partir de las historias de usuario creadas, además de los escenarios de pruebas planteados por cada una. Se aseguró que los criterios de aceptación y cada resultado esperado por prueba, fuera congruente con las peticiones y respuestas desarrolladas.

BDD surge como evolución de TDD, basa su lenguaje en el comportamiento, y describe en un lenguaje de fácil entendimiento tanto para desarrolladores, como para cualquier persona interesada, los criterios de aceptación que se tienen por cada característica previamente planteada.

La estructura de un escenario de prueba se desarrolló bajo el siguiente esquema, teniendo en cuenta el lenguaje Gherkin, comprensible por humanos y ordenadores, y aceptado en un plan de desarrollo guiado por comportamiento:

Característica: historia de usuario.

Escenario: breve descripción de la validación a realizar. Se incluyó si el escenario de prueba validaba un caso positivo o negativo.

Dado: condiciones previas que un actor debe cumplir antes de llevar a cabo una funcionalidad.

Cuando: función a ejecutar.

Entonces: resultado esperado de la función.

Dentro del esquema previamente planteado se pueden agregar dos aserciones más, **Y**, que denota una agregación al contexto, o por su contrario, **Pero**, que como en lenguaje natural, enlaza dos oraciones que se contraponen.

En el marco del proceso de validación y verificación del servidor, se decide enfocarse en las pruebas de integración, más exactamente las pruebas de API. Se siguen los escenarios de prueba planteados para validar una petición al servidor y la respuesta obtenida en diferentes contextos. Dichos escenarios fueron aprobados por el equipo de trabajo, previo al desarrollo de la solución.

Para llevar a cabo las pruebas se propusieron dos alternativas, ambas ejecutadas. La primera, obligatoria, ejecutar cada escenario de prueba de forma manual, realizando una petición REST por medio del ambiente de desarrollo para API, Postman.

Posterior a la ejecución de una prueba manual, si el resultado es el esperado y la prueba pasa, se procedía a automatizar la prueba, es decir, codificarla para correrla en múltiples ocasiones con menor costo en cuanto a tiempo se refiere. Como ventaja, en caso de ser necesario, al implementar o corregir funcionalidades, las pruebas automatizadas fueron ejecutadas nuevamente, con el fin de validar si una nueva integración pudo afectar de forma negativa la versión previa del software.

Para la automatización de las pruebas se hizo uso de Cucumber y Serenity, software enfocado en correr escenarios de prueba de forma automática, leyendo los casos de prueba escritos en Gherkin; y, conjunto de librerías utilizadas para la generación de reportes automatizados, respectivamente.

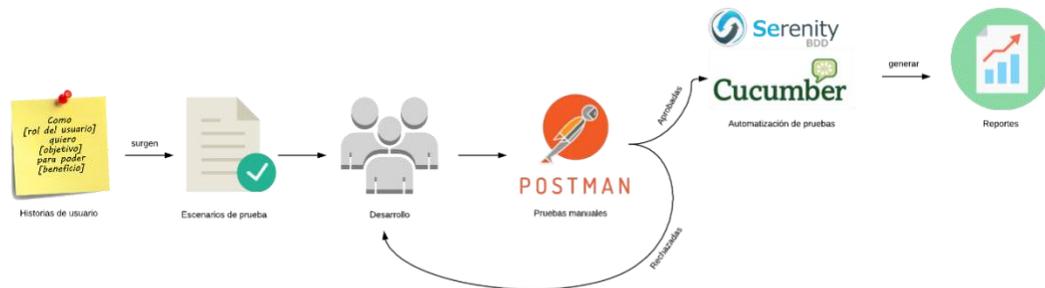


Figura 41: Proceso de ejecución de pruebas.

La figura 57 responde al proceso general seguido para la validación y verificación de las pruebas. Incluye la definición de las historias de usuario y de los escenarios de pruebas, seguido por el desarrollo de la solución, hasta la ejecución de pruebas manuales y automatizadas por documento y microservicio.

En la figura 58 se muestra un ejemplo de un reporte generado al ejecutar pruebas automatizadas sobre un documento de pruebas.

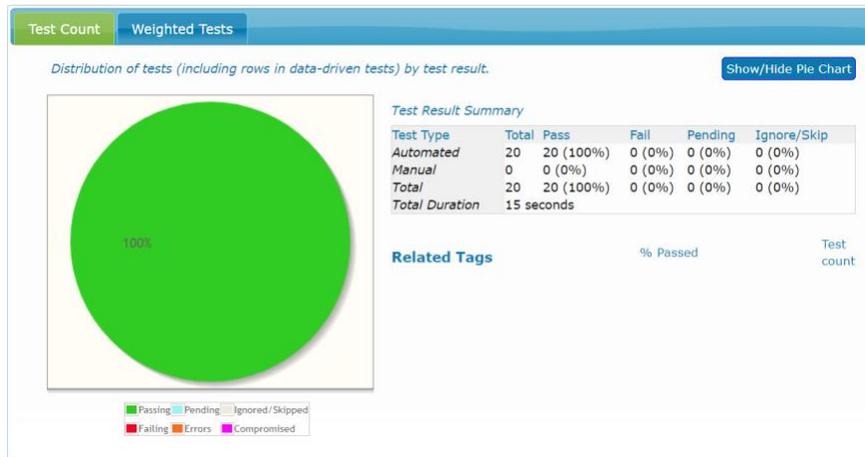


Figura 42: Ejemplo de reporte para pruebas automatizadas de integración.

Como resultado de lo desarrollado se obtuvieron 8 documentos de pruebas, uno por cada historia de usuario, con aproximadamente 25 escenarios de pruebas. Al ser el diseño una arquitectura orientada a microservicios, se propuso agrupar los documentos de pruebas acorde a las funcionalidades expuestas de la siguiente forma:

- **Users:** crear cuenta, iniciar sesión, cerrar sesión, ver perfil y editar perfil.
- **Common:** desarrollar nivel.
- **Dictionary:** gestionar diccionario.

Al finalizar el proyecto, todas las pruebas fueron automatizadas, además de tener un 100% como resultado de pruebas exitosas. Cabe anotar, que este porcentaje es únicamente enfocado en pruebas de API desarrolladas para el servidor.

VIII- CONCLUSIONES

1. Análisis del impacto del proyecto

La solución desarrollada que surge del proyecto de grado sirve como una herramienta útil para disminuir la barrera de comunicación con la población sorda debido a que invita a la población sin discapacidad auditiva a incursionar en el LSC. Esto se ve reflejado en las pruebas con los usuarios finales, de usabilidad y de aceptación que marcaron resultados positivos.

Como se ha dicho, el proyecto tiene la finalidad de divulgar el LSC. Esto lo conocerán primeramente muchos colombianos que conocen a personas sordas, a su vez, para ellos despertará el interés de practicar gestos básicos. En consecuencia, esto provocará que más personas pertenecientes a este círculo social la conozcan. Para citar un buen ejemplo se puede imaginar a una persona sorda laborando donde una interacción continua con sus compañeros de trabajo o también cuando una persona sorda hace salidas donde requiere buscar información ya sea para solicitar una cita médica o ir al cine. Si la comunicación a las personas sordas mejora, el entorno en la que ellas conviven mejoraría al ser incluyente en actividades comunes ya mencionadas como en la educación, la salud, el trabajo, el deporte, entre otros.

Dicho hasta aquí las personas tendrán mayor acceso a la información con el fin de conocer y practicar el LSC. De manera que, se podrá contribuir positivamente al apoyo de la población con discapacidad auditiva y su acceso en términos de oportunidades en igualdad de condiciones, así como la población con capacidad para oír [7].

2. Conclusiones

Al ser un proyecto desarrollado por tres personas, se incursiono en el trabajo en equipo definiendo tareas específicas asignadas a cada uno de los integrantes según las habilidades para desarrollarlas. Se logró conocer de manera general la estructura y sintaxis de la LSC al definir la cobertura del léxico, lo cual permitió crear un método para mostrar las señas.

Se realizó una planificación del proyecto, tanto en la definición de conceptos del LSC, como en la apariencia visual de la aplicación y su interacción a través de mockups y el instructivo de ludificación, que sirvieron como guía en el proceso de prácticas de la lengua de señas.

Tener como objetivo la inclusión de las personas con discapacidad auditiva, alentó a que el proyecto contará con la asesoría de personas relacionadas a la comunidad de sordos, por lo que la tarea desarrollada de filmar 87 videos de señas, conto con la colaboración de Miguel Mejia, que como interprete oficial del LSC, fue ideal para esta tarea..

Cuando se definió el algoritmo de reconocimiento de imágenes se encontró que además de algoritmos a implementar, también existía servicios novedosos que ofrecían reconocer conceptos de imágenes que el mismo desarrollador podía poner a entrenar. Con ayuda de la documentación ofrecida por Microsoft CustomVision, que fue el servicio elegido, se logró implementar el algoritmo de reconocimiento de gestos, que por medio de *machine learning*, identifica señas estáticas realizadas, más exactamente correspondientes a la práctica del abecedario y los números. Esta tecnología logró satisfacer las necesidades de la aplicación.

Adicionalmente, por medio de la adaptación de metodologías ágiles se logró diseñar e implementar una aplicación móvil, un servidor de aplicaciones, y una página web funcionales, que ofrecen servicios a los usuarios y administradores, permitiendo mostrar y gestionar el diccionario de señas, además de los ejercicios y componentes establecidos en el documento de técnicas de ludificación. Esta arquitectura definida, donde se separan responsabilidades, permite que la aplicación sea robusta en el sentido que su contenido puede seguir aumentando y es posible adaptarse a nuevos requerimientos que surjan en el futuro.

Por último, el proceso de validación y verificación en donde se realizaron además de pruebas funcionales, pruebas con los usuarios finales, de usabilidad y de aceptación, marcaron resultados positivos relacionados al diseño, funcionamiento y uso de la aplicación móvi.

3. Trabajo futuro

Los objetivos planteados en este proyecto fueron cumplidos satisfactoriamente, pero durante el desarrollo de estos se descubrieron nuevas funcionalidades que podrían implementarse a mediano o largo plazo, las cuales identificamos y detallamos a continuación:

- Realizar un servicio para que el administrador agregue imágenes de entrenamiento a partir de las predicciones realizadas.
- Agregar otro tipo de actividades lúdicas.
- Añadir en todas las secciones lúdicas después del nivel de verbos que tengan frases conjugadas.
- Agregar niveles que involucre vocabulario cotidiano.
- Agregar funcionalidades en el administrador que involucre crear nuevos niveles y logros.
- Agregar una sección de desafío de agilidad donde el usuario practique las señas en contra-reloj y pueda realizar la mejor puntuación, como se ve en la 59.
- Ampliar funcionalidad de el algoritmo de reconocimiento de imágenes para que pueda también reconocer videos.

IX- REFERENCIAS

- [1] Departamento Nacional de Estadística, "Censo General," DANE.
- [2] Instituto Nacional para Sordos, "Boletín No.3 Observatorio Social Población Sorda Colombiana," INSOR.
- [3] Instituto Nacional Para Sordos, "«Estudiantes Sordos en la Educación Superior. Equiparación de Oportunidades»,," INSOR.
- [4] A. G. Barreto, "La Incidencia de la Interpretación de Lengua de Señas en la Educación de los Sordos: Dos Hipótesis a Partir de la Obra de Harlan Lane," Universidad Nacional Abierta y a Distancia UNAD.
- [5] D. L. Sánchez et al., «MINISTERIO DE TECNOLOGÍAS DE LA INFORMACIÓN Y LAS COMUNICACIONES REPÚBLICA DE COLOMBIA», p. 52, 2017.
- [6] C.E.E. Tiempo, "Colombia: ¿un territorio difícil para los móviles Apple?", vol. 2017, no. Nov 5.
- [7] DICCIONARIO BÁSICO, DE LA and LENGUA DE SEÑAS COLOMBIANA, "DICCIONARIO BÁSICO DE LA LENGUA DE SEÑAS COLOMBIANA,".
- [8] I. L. Chacón, "Modelo de Identificación y clasificación del lenguaje de señas colombiano,".
- [9] R. Hartanto and A. Kartikasari, "Android based real-time static Indonesian sign language recognition system prototype," 2016 8th International Conference on Information Technology and Electrical Engineering (ICITEE), 10, pp. 1-6.
- [10] Z. Xiaolong and W. Kwan-Yee, "Single-Frame Hand Gesture Recognition Using Color and Depth Kernel Descriptors,".
- [11] G. A. Realpe, "Reconocimiento del Lenguaje de Señas Manuales con el Kinect," Universidad de los Andes.
- [12] S. Deterding, M. Sicart, L. Nacke, K. O'Hara and D. Dixon, "Gamification. Using Game-design Elements in Non-gaming Contexts," pp. 2425-2428.
- [13] D. Dicheva, C. Dichev, G. Agre and G. Angelova, "Gamification in Education: A Systematic Mapping Study," , vol. 18, 2015.
- [14] K. M. Kapp, «The Gamification of Learning and Instruction», p. 5, 2013.
- [15] B. Wake, «INVEST in Good Stories, and SMART Tasks», XP123. 2018.
- [16] «Flaticon, the largest database of free vector icons», Flaticon. [En línea]. Disponible en: <https://www.flaticon.com/>. [Accedido: 16-jun-2018].
- [17] VasuAgrawal, GestureDetection: 15-112 Term Project. 2018.

- [18] The One, Hand Gesture Detection AI With Convolutional Neural Networks. 2018.
- [19] «Custom Vision». [En línea]. Disponible en: <https://www.customvision.ai/>. [Accedido: 16-jun-2018].
- [20] «API Vision: análisis del contenido de las imágenes», Google Cloud. [En línea]. Disponible en: <https://cloud.google.com/vision/?hl=es>. [Accedido: 16-jun-2018].
- [21] «Amazon Rekognition – Videos e imágenes – AWS». [En línea]. Disponible en: <https://aws.amazon.com/es/rekognition/>. [Accedido: 16-jun-2018].
- [22] anrothMSFT, «Overview of Custom Vision Service machine learning - Azure Cognitive Services». [En línea]. Disponible en: <https://docs.microsoft.com/en-us/azure/cognitive-services/custom-vision-service/home>. [Accedido: 16-jun-2018].
- [23] «Paneles de control», Android Developers. [En línea]. Disponible en: <https://developer.android.com/about/dashboards/?hl=es-419>. [Accedido: 16-jun-2018].
- [22] «Android Architecture Components», Android Developers. [En línea]. Disponible en: <https://developer.android.com/topic/libraries/architecture/>. [Accedido: 16-jun-2018].
- [23] «Android From Scratch: Using REST APIs». [En línea]. Disponible en: <https://code.tutsplus.com/tutorials/android-from-scratch-using-rest-apis--cms-27117>. [Accedido: 16-jun-2018].
- [24] P. Costa, «Android Networking II: OkHttp, Retrofit, Moshi and Picasso», Medium, 04-jun-2017.
- [25] «Introduction to Dagger 2, Using Dependency Injection in Android: Part 1», Mindorks, 26-dic-2016.
- [28] «Dagger». [En línea]. Disponible en: <https://google.github.io/dagger/>. [Accedido: 16-jun-2018].
- [29] Google Developers, ExoPlayer: Adaptive video streaming on Android. 2018.
- [31] «Glide vs. Picasso – Multidots – Medium». [En línea]. Disponible en: <https://medium.com/@multidots/glide-vs-picasso-930eed42b81d>. [Accedido: 16-jun-2018].
- [32] «What are microservices?» [En línea]. Disponible en: <https://www.redhat.com/es/topics/microservices/what-are-microservices>. [Accedido: 16-jun-2018].

- [33] «AWS | Cloud Computing - Servicios de informática en la nube», Amazon Web Services, Inc. [En línea]. Disponible en: <https://aws.amazon.com/es/>. [Accedido: 16-jun-2018].
- [34] «MongoDB for GIANT Ideas | MongoDB». [En línea]. Disponible en: <https://www.mongodb.com/>. [Accedido: 16-jun-2018].
- [35] Lydia Leong, Raj Bala, Craig Lowery and Dennis Smith, "Magic Quadrant for Cloud Infrastructure as a Service, Worldwide," Jun 15, pp. 9.
- [36] TechMagic, «ReactJS vs Angular5 vs Vue.js — What to choose in 2018? », Medium, 16-mar-2018.
- [37] «Comparison with Other Frameworks — Vue.js». [En línea]. Disponible en: <https://vuejs.org/v2/guide/comparison.html#Angular-Formerly-known-as-Angular-2>. [Accedido: 16-jun-2018].
- [38] «Android Jetpack | Android Developers». [En línea]. Disponible en: <https://developer.android.com/jetpack/>. [Accedido: 16-jun-2018].
- [39] «Angular CLI». [En línea]. Disponible en: <https://cli.angular.io/>. [Accedido: 16-jun-2018].
- [40] «HTTP Methods for RESTful Services». [En línea]. Disponible en: <http://www.restapitutorial.com/lessons/httpmethods.html>. [Accedido: 16-jun-2018].
- [41] «Angular - QuickStart». [En línea]. Disponible en: <https://angular.io/guide/quickstart>. [Accedido: 16-jun-2018].
- [42] «ACCU: The Philosophy of Extensible Software». [En línea]. Disponible en: <https://accu.org/index.php/journals/391>. [Accedido: 16-jun-2018].
- [43] «Questionnaire For User Interaction Satisfaction». [En línea]. Disponible en: <http://lap.umd.edu/quis/>. [Accedido: 16-jun-2018].
- [44] «Encuesta NPS | e-encuesta.com». [En línea]. Disponible en: <https://www.e-encuesta.com/nps/>. [Accedido: 16-jun-2018].
- [45] «Introduction to BDD». [En línea]. Disponible en: <http://jbehave.org/introduction.html>. [Accedido: 16-jun-2018].