

# Diccionario LSC- Aprende Lenguaje de señas

1

Medina Gabriel Andrés, Solano Yirlan Andrey, Urrutia Juan Pablo, Santamaría Juan Nicolás

No. de Equipo Trabajo: X

## I. INTRODUCCIÓN

Según datos del Instituto Nacional para Sordos (INSOR), en Colombia para el año 2018 había 543.000 personas sordas en el territorio Nacional, de los cuales cerca de 64.482 personas estaban en Bogotá, esto basado en los datos proyectados de la Secretaría de Planeación Distrital para el mismo año [1].

La sordera representa un reto en la comunicación de las personas que la padecen, pues al no contar con una competencia lingüística tienen que recurrir a factores extralingüísticos, como mímicas o gestos, para darse a entender de una manera elemental. Es por esto que, en 1996, el Estado Colombiano reconoce La Lengua de Señas Colombiana (LSC) como lengua propia de la población sorda, para que junto con otras competencias como la escritura se llegue a un nivel más elaborado de comunicación

## II. DESCRIPCIÓN DEL PROBLEMA A RESOLVER

La gran mayoría de las personas sin ninguna discapacidad auditiva desconocen el lenguaje de señas y su importancia, de hecho, un gran porcentaje de la misma población sorda no sabe LSC y son analfabetas. La educación es uno de los grandes retos que tienen las personas sordas, y cada año se ve más excluidos según datos de del Sistema Integrado de Matricula (SIMAT) para el 2013 habían cerca de 10464 estudiantes sordos matriculados en establecimientos educativos, sin embargo, para el año 2018 esta cifra disminuyó a 6160 estudiantes.[2]

Es por lo nuestro objetivo es crear un aplicativo móvil, que consta de un diccionario de Lenguaje de Señas Colombiano que permita a las personas no sordas tener una herramienta que dado el caso les permita tener una conversación con una persona con discapacidad auditiva. Asimismo, se plantea que dicha aplicación cuente con una herramienta practica en donde se fortalezca el aprendizaje de las palabras más comunes, utilizando la técnica de la repetición espaciada. Y la posibilidad de que usuario agregue o elimine las palabras que tenga la necesidad o intención de aprender en la herramienta de práctica.

## III. USUARIOS DEL PRODUCTO DE SOFTWARE

Con el fin de promover la aplicación en distintos ámbitos educativos y que esta sirva como una herramienta tanto para estudiantes autodidactas, como para las instituciones o profesores, se establecieron tres roles dentro de la aplicación.

- Visitante: Usuario que puede usar la aplicación sin la necesidad de registrarse, pero no hacer un seguimiento de su avance ni estadísticas.
- Usuario autenticado: Luego de hacer el registro dentro de una de las pantallas de la aplicación, esta puede acceder a estadísticas y graficas que le den retroalimentación sobre el proceso de aprendizaje espaciado.
- Administrador: Los usuarios con permisos de administrador pueden añadir, editar y eliminar palabras del diccionario. Esto con el fin de que si un profesor quiere que sus estudiantes aprendan palabras específicas del LSC y que no estén añadidas puedan incluirlas.

## IV. REQUERIMIENTOS FUNCIONALES DEL SOFTWARE

### • Autenticarse en la aplicación

En la primera pantalla, cuando se ejecuta la aplicación. El usuario podrá iniciar sesión, crear en una cuenta en caso de que no tenga una o entrar como un invitado sin la necesidad de autenticarse.

Al hacer login en la aplicación, se leerán los datos ingresados en los dos espacios correspondientes. El primero para el correo del usuario y el Segundo para la contraseña. Estos datos se comparan con los de los usuarios registrados en la nube con Google Authentication. Si el usuario ya tenía una cuenta creada, puede ingresar a la aplicación como administrador o como usuario común; de lo contrario, se le pide que vuelva a ingresar sus datos.

Como requerimiento no funcional, la comparación de los datos ingresados con los de la nube se realiza con tiempo constante dependiendo de la conexión a internet del usuario.

### • Registrar Usuarios nuevos

Si un usuario no está registrado puede crear una cuenta por medio del botón de registro que lo llevará directamente a otra pantalla donde tiene que ingresar el correo con el que desea registrarse y una contraseña válida dos veces para confirmar su correcta escritura, y definir si es administrador o no.

Los datos ingresados son comparados con los almacenados en Google Authentication para verificar que no exista otro usuario con el mismo correo. Si el registro es exitoso el correo y contraseña ingresados son guardados y el usuario ingresa a la aplicación como administrador o usuario dependiendo de su elección. Si el registro no es exitoso se le pide al usuario volver a ingresar sus datos. Nuevamente como requerimiento no funcional, el registro se realiza en tiempo constante dependiendo de la conexión a internet del usuario y la interfaz tiene mensajes emergentes que guían al usuario de cualquier error.

- *Buscar Palabras*

En la pantalla de diccionario se podrá consultar las palabras dentro de la base de datos por medio de una barra de búsqueda. Esta debe ser lo mas intuitiva posible, que tenga la opción de clasificar en diferentes temas (requerimiento no funcional) y asimismo completar la búsqueda en el menor tiempo posible. Esta opción la tendrá disponible los 3 roles de usuario (Invitado, usuario común o administrador).

- *Agregar una palabra nueva*

Los usuarios con el rol de administrador contarán con una opción adicional en la pantalla de diccionario que no será visible para un usuario común. Esta le permitirá agregar una palabra en la base de datos del diccionario que no exista ya y que cumpla con los requerimientos mínimos para crear una palabra, que por el momento es nombre y definición.

- *Eliminar una palabra existente*

Al igual que en el punto anterior, el usuario con rol de administrador podrá eliminar palabras de la base de datos que no considere necesarias o apropiadas considerando el enfoque pedagógico de la aplicación con el lenguaje LCS. Esto le permite a los educadores que quieren enfocar la aplicación para la enseñanza de ciertos temas en particular o consideren que tal vez ciertas palabras no son adecuadas para sus estudiantes.

- *Obtener un registro de la actividad del usuario*

Los usuarios que son autenticados podrán tener información útil sobre las palabras que más han buscado y la cantidad de veces que las han buscado. Esto con el fin de que los usuarios pueden tener un registro de las palabras aprendidas y un acceso rápido a estas.

- *Juegos para aprender LSC*

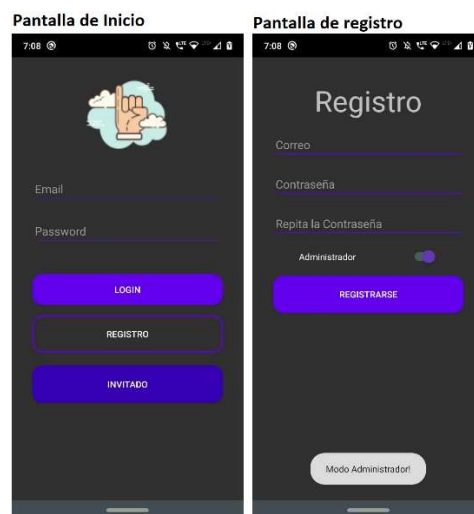
En la pantalla de “Aprender” se contarán con varios minijuegos que sirvan para que el usuario aprenda las palabras en lenguaje de señas, estas palabras estarán conectadas con la base de datos y actualizaran su estado según los errores y aciertos del usuario. Es decir, una palabra en la cual siempre se equivoque se ubicará en el nivel mas bajo de la repetición espaciada contrario a una

donde siempre se acierte. Este requerimiento se busca que sea rápido e implemente tanto la búsqueda como la actualización de los datos.

## V. DESCRIPCIÓN DE LA INTERFAZ DE USUARIO PRELIMINAR

Se plantean varias pantallas básicas para la aplicación.

Primero La pantalla de inicio de sesión, donde se definen los 3 roles del usuario. Le botón de registro lo lleva a la pantalla para registrar un usuario y agregarlo a la base de datos.



*Ilustración 1 Pantalla Inicio- Pantalla Crear Usuario*

Ya con el usuario registrado y habiendo iniciado sesión se accede a las siguientes pantallas conectadas entre las tres por medio de la barra inferior.



*Ilustración 2 Pantallas Diccionario, Aprender y Perfil*

En la primera pantalla, la de la izquierda, es el diccionario. Acá se podrá buscar las palabras que están en la base de datos, y en

caso de ser administrador, eliminar o agregar palabras a la base de datos. De no ser administrador

En la pantalla del medio “Aprender-Jugar”, encontraremos las actividades basadas en aprendizaje ramificado y aprendizaje por medio de repetición espaciada. Dado el objetivo académico de este proyecto, esta pantalla será una de las ultimas de implementar.

En la pestaña de Usuario encontraremos graficas que muestran el progreso del perfil, un botón para salir de la sesión y una lista de trofeos que se obtendrán de la pantalla de aprendizaje.

## VI. ENTORNOS DE DESARROLLO Y DE OPERACIÓN

La aplicación se desarrollará en la última versión Android Studio, en su configuración con Java, el manejo de las versiones se hará en Git, en preferencia se usará Github Desktop o SourceTree.

El software operará en dispositivos Android a partir de la versión 5.0 “Lollipop”, en cuanto a requerimientos de hardware, no se hará un uso exigente de los mismos por tanto no existe ningún requerimiento mínimo más allá de la versión de Android.

## VII. PROTOTIPO DE SOFTWARE INICIAL

Como fue mencionado anteriormente, el control de versiones del proyecto se hace en Git, y el repositorio del proyecto es:

<https://github.com/andreysolano/Diccionario-LSC>

```

root
├── README.md
├── src
│   ├── my.packages // Carpeta donde se almacena el código fuente
│   │   ├── Main.(*) // Paquete principal del proyecto (kernel)
│   │   ├── ... // Clase principal que ejecuta la aplicación. Ejemplo. Java: cl
│   │   └── ... // Otros archivos java con el código fuente de la app
├── docs
│   ├── Entrega_(Plantilla).pdf //Documento plantilla con la información de la er
│   ├── .....
│   └── OtrosDocumentos_X.pdf //Otros documento según se requiera
├── data //Carpeta con los datos de prueba del proyecto
├── dist //Carpeta en donde se almacenan los archivos que ejecutan la apl
│   ├── proyectoEntrega1.jar
│   ├── proyectoEntrega2.jar
│   ├── proyectoEntrega3.jar
│   └── .....
├── lib // Carpeta con las librerías requeridas en la aplicación
└── Otros archivos... // Otros archivos de ejecución y configuración necesarios.
    // Ejemplo: project.xml, build.xml (ant), run.sh, run.bat (ejec

```

Las palabras del diccionario se encuentran en un archivo XML que está guardado en la memoria interna del dispositivo. Al iniciar la aplicación se comprueba si el dispositivo cuenta con conexión a internet, de ser positiva, se descargan los datos de Fire Base - Real Time Data Base para reemplazar las palabras del XML. De esta forma, si se agregó o eliminó alguna palabra en algún dispositivo, los cambios se vuelven efectivos para todos los demás usuarios.

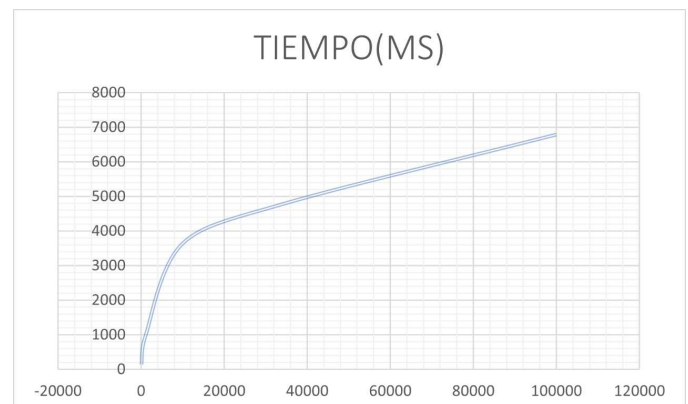
## VIII. IMPLEMENTACIÓN Y APLICACIÓN DE LAS ESTRUCTURAS DE DATOS

En este prototipo se cambió el manejo de los datos de una lista doblemente enlazada a un árbol binario que se organiza acorde al orden lexicográfico de las palabras. Además, se dejó de guardar la información de los usuarios registrados de manera interna y se empezó a manejar de manera externa en Google Authentication.

El árbol que contiene las palabras, almacena los objetos palabra (una clase creada que contiene el ID, la palabra como String, el GIF o imagen de su forma LCS y el significado da cada palabra) y las organiza dejando en la rama derecha las de mayor valor y a la izquierda de menor valor. A este, se le implementaron métodos de inserción, eliminación y búsqueda de palabras que funcionan recorriendo el árbol desde la raíz hasta la palabra deseada de manera recursiva. En el caso de la eliminación, al retirar la palabra, si esta no corresponde a una hoja del árbol, este se reorganiza para que siga cumpliendo el mismo orden pre establecido.

## IX. PRUEBAS DEL PROTOTIPO Y ANÁLISIS COMPARATIVO

Para probar el tiempo de ejecución de los diferentes métodos se realizaron arboles con la organización explicada anteriormente en el documento y se midió el tiempo en el que comenzaban y terminaban las operaciones sobre este para saber el tiempo que demoraban en terminar. Se tomaron medidas desde los 100 datos hasta el millón para cada método. El tiempo de ejecución de los tres métodos (Búsqueda, inserción y eliminación) tiende a ser logarítmico ( $O(\log n)$ ) como se muestra en la siguiente gráfica, lo que es una mejora considerando que anteriormente el tiempo de ejecución fue lineal ( $O(n)$ ) con listas doblemente enlazadas.



## X. ROLES Y ACTIVIDADES

Se debe describir de manera concisa los roles asignados y las respectivas actividades realizadas por cada uno de los integrantes del equipo durante el desarrollo de esta entrega. Para esto, tenga en cuenta los siguientes roles positivos, que puede aplicar durante el desarrollo del proyecto; los mismos deberán ser ‘rotados’ entre todos los integrantes del equipo en las diferentes entregas.

	Actividades fundamentales
Líder/esa	Consultar a los otros miembros del equipo, atento que la información sea constante para todos. Aportar con la organización y plan de trabajo.
Coordinador/a:	Mantener el contacto entre todos, Programar y agendar y reuniones, Ser facilitador con el acceso a los recursos.
Experto:	Líder técnico que propende por coordinar las funciones y actividades operativas.
Investigador	Consultar otras fuentes. Propender por resolver inquietudes comunes para todo el equipo.
Observador	Siempre está atento en el desarrollo del proyecto y aporta en el momento apropiado cuando se requiera apoyo adicional por parte del equipo.
Animador/a	Energía positiva, motivador en el grupo.
Secretario	Se convierte en un facilitador de la comunicación en el grupo. Documenta (actas) de los acuerdos/compromisos realizados en las reuniones del equipo.
Técnico	Aporta técnicamente en el desarrollo del proyecto.

Puede utilizar la siguiente tabla para definir los integrantes del grupo, los roles asignados y el listado de actividades durante el desarrollo de la entrega:

INTEGRANTE	ROL(ES)	ACTIVIDADES REALIZADAS (Listado)
Gabriel Medina	Investigador	Lectura y escritura de documentos descargados de la base de datos.
	Técnico	
Andrey Solano	Líder	Vinculación de la parte gráfica en Android Studio con la estructuración lógica de los datos.
	Técnico	
Juan Urrutia	Experto	Implementar el árbol que organiza los datos junto con su métodos de adición, eliminación y búsqueda.
	Técnico	
Juan Nicolás Santamaría	Investigador	Uso de Fire Base para como base de datos y de login y registro de usuarios.
	Técnico	

## XI. DIFICULTADES Y LECCIONES APRENDIDAS

Inicialmente se pensó en guardar y descargar un archivo XML de Fire Base que contuviera las palabras del diccionario y así solo reemplazar el que se encuentra en la memoria interna cada vez que el usuario abriera la aplicación con conexión internet. Pero rápidamente tocó descartar esta opción ya que al descargar el archivo de Fire Base - Storage, este no quedaba con los permisos necesarios para ingresar y modificar su contenido para poder construir el árbol a partir de este. Como solución, se decidió guardar las palabras una por una en Fire Base – Real Time Data Base y descargarlas para construir el árbol a partir de estas cuando el usuario inicia la aplicación y tiene conexión a internet.

Por otro lado, al crear los usuarios, en Fire Base – Authentication, no quedaba registrado si estos eran

administradores o no, por lo que fue necesario crear una rama adicional en Fire Base – Real Time Data Base que guarda el ID de cada usuario registrado y guarda si es un administrador o no.

Por último, dado que las pruebas de rendimiento requieren de grandes cantidades de datos, fue necesario aprender sobre el manejo e integración de grandes bases de datos, lo que resultó un poco problemático en un comienzo.

## XII. REFERENCIAS BIBLIOGRÁFICAS

[1] Insor.gov.co. 2020. Preguntas Frecuentes – INSOR – INSOR Instituto Nacional Para Sordos. [online] Available at: <<http://www.insor.gov.co/home/servicio-alciudadano/preguntas-frecuentes/>> [Accessed 8 September 2020].

[2]2019. Plan Institucional 2019-2022. online] Available at:  
<[http://www.insor.gov.co/home/descargar/plan\\_estrategico\\_NSOR\\_2019\\_2022V1.pdf](http://www.insor.gov.co/home/descargar/plan_estrategico_NSOR_2019_2022V1.pdf) [Accessed 8 September 2020].