****

**Documento de requerimientos de software**

**Proyecto:** APP LENGUA DE SEÑAS

**Autores:** Sebastián Peralta, Jesús Morales, Byron Román

**Versión del documento**: 1.3

25 de septiembre del 2024

**Índice de contenidos**

[**Versiones de documento 3**](#_4bbq91ubu0k5)

[Introducción 3](#_4zlobvf54ps1)

[**1. Propósito 3**](#_nrvscdbpl1np)

[**2. Alcance del producto / Software 3**](#_qwlryitrkqii)

[2.1. Objetivo General 3](#_l208dz8q6rca)

[2.2. Beneficios 4](#_o5pflrwj5net)

[**3. Funcionalidades del producto 4**](#_p4dik8jzc96n)

[3.1. Registro y autenticación de usuarios 4](#_apjeuwep9ux0)

[3.2. Validación de señas en tiempo real 4](#_z04rykbrnqeo)

[3.3. Desafíos educativos 4](#_nd6mqkhyuz8g)

[3.4. Envío de notificaciones 4](#_v1stmu8ccvx4)

[**4. Clases y características de usuarios 4**](#_egd3njonistj)

[4.1. Usuarios Generales 4](#_f2t78ik1d5oj)

[4.2. Educadores 5](#_i1druweo6jlz)

[4.3. Personas con Discapacidad Auditiva 5](#_sxjxa5207w)

[**5. Entorno operativo 5**](#_uym44rxx8y5r)

[5.1. Frontend 5](#_axryj3zfsig)

[5.2. Backend 5](#_ve5uow3cr0cn)

[5.3. Base de datos 5](#_k6iqfx10njv5)

[5.4. Servicios de autenticación 5](#_8p4of2f2fl3e)

[5.5. Inteligencia artificial 5](#_txzjxv21qd7u)

[5.6. Herramientas de desarrollo 5](#_hkrkb520121t)

[**6. Requerimientos funcionales 6**](#_iscr4efdwj1q)

[6.1. Tabla de características de la Norma ISO/IEC 25010 6](#_eglsodliavte)

[6.2. Tabla de requerimientos funcionales 8](#_3jxx0a7jk3e8)

[**7. Requerimientos no funcionales 10**](#_g8fay8dgb5kw)

[7.1. Tabla de requerimientos no funcionales 10](#_vth1nu1k5md8)

[**8. Reglas de Negocio 13**](#_8togtomxx02m)

[**9. Glosario de términos 13**](#_hknmr1f8dimo)

[**10. Bibliografía 13**](#_ejtj4cz6ojjc)

#### 

#### 

#### 

# Versiones de documento

| **Versión** | **Fecha** | **Descripción** | **Autor** |
| --- | --- | --- | --- |
| 1.0 | 22/09/2024 | Documento inicial | Sebastián Peralta |
| 1.1 | 22/09/2024 | Actualización de requerimientos | Byron Román |
| 1.2 | 22/09/2024 | Finalización de documento | Jesús Morales |
| 1.3 | 25/09/2024 | Implementación de criterios de la Norma ISO/IEC 25010 | Byron Román |

# Introducción

# Propósito

El presente documento describe los requerimientos funcionales y no funcionales de la **App Lengua de Señas**, una aplicación educativa destinada a enseñar lengua de señas mediante desafíos interactivos y retroalimentación en tiempo real a través de inteligencia artificial. El objetivo principal es fomentar la inclusión social y ofrecer una herramienta accesible para el aprendizaje de la lengua de señas.

Este documento sigue las directrices de la norma ISO/IEC 25000 para la gestión y evaluación de la calidad del software, asegurando que los requerimientos funcionales y no funcionales cumplan con los estándares internacionales de calidad.

# Alcance del producto / Software

El desarrollo del software sigue las guías establecidas por la norma ISO/IEC 25000, con especial énfasis en la norma ISO/IEC 25010, que define el modelo de calidad para la evaluación de los sistemas y software. Esto garantiza que el producto cumpla con los requisitos de calidad en cuanto a funcionalidad, usabilidad, rendimiento, seguridad y mantenibilidad.

## Objetivo General

Los objetivos del software son los siguientes:

* Desarrollar una aplicación accesible que permita a los usuarios aprender lengua de señas en distintos niveles (básico, intermedio y avanzado).
* Integrar inteligencia artificial para validar las señas en tiempo real, proporcionando retroalimentación inmediata.
* Ofrecer recursos educativos gratuitos y complementarios, como videos y artículos, que mejoren la experiencia de aprendizaje.
* Crear una plataforma inclusiva que funcione en dispositivos móviles Android.

## **Beneficios**

* Aumentar la inclusión de las personas con discapacidad auditiva en entornos sociales y laborales.
* Proporcionar una herramienta educativa útil para escuelas, universidades y centros de trabajo que buscan ser más inclusivos.
* Apoyar a los profesionales que trabajan con personas con discapacidad auditiva, como terapeutas del lenguaje, psicólogos, maestros y fonoaudiólogos.

# Funcionalidades del producto

## Registro y autenticación de usuarios

* El usuario debe poder registrarse e iniciar sesión utilizando **Firebase Authentication** con sus cuentas de Google, Facebook, Apple, Twitter o mediante correo electrónico.

## Validación de señas en tiempo real

* Implementar **TensorFlow Lite** para analizar y validar los gestos de los usuarios a través de la cámara del dispositivo.
* Retroalimentación visual en tiempo real sobre la precisión de las señas.

## Desafíos educativos

* El usuario puede participar en desafíos y juegos educativos que refuercen el aprendizaje de la lengua de señas, con niveles que aumentan en dificultad progresivamente.

## Envío de notificaciones

* Envío de notificaciones push a los usuarios a través de **Firebase Cloud Messaging**, recordando desafíos diarios, nuevas lecciones o retroalimentación de sus progresos.

# Clases y características de usuarios

## Usuarios Generales

Personas interesadas en aprender lengua de señas de todas las edades, desde estudiantes hasta adultos.

## **Educadores**

Usuarios con un perfil docente que podrán monitorizar el progreso de los estudiantes, crear retos personalizados y generar informes.

## Personas con Discapacidad Auditiva

Usuarios que desean mejorar sus habilidades en lengua de señas para facilitar su comunicación en diferentes entornos.

# Entorno operativo

## Frontend

* **Flutter (Dart)**: Permite desarrollar la aplicación móvil con interfaces intuitivas y eficientes.

## Backend

* **Firebase Functions (JavaScript)**: Para manejar la lógica del servidor, integraciones con APIs y autenticación.

## Base de datos

* **Firebase Firestore**: Para almacenar datos de los usuarios y su progreso.
* **Firebase Cloud Storage**: Para guardar archivos multimedia relacionados con los juegos y ejercicios.

## Servicios de autenticación

* **Firebase Authentication**: Para autenticar usuarios con correo electrónico o redes sociales.

## Inteligencia artificial

* **TensorFlow Lite**: Para la validación de señas en tiempo real.

## Herramientas de desarrollo

* **Figma**: Diseño y prototipado de la interfaz de usuario.
* **Firebase Analytics**: Para monitorear el comportamiento de los usuarios.
* **Firebase Crashlytics**: Detección y reporte de fallos.
* **Firebase Cloud Messaging**: Envío de notificaciones push.
* **Firebase Test Lab**: Para realizar pruebas automatizadas y manuales de la aplicación en una variedad de dispositivos y configuraciones.
* **Firebase App Distribution**: Es una herramienta que permite distribuir versiones de prueba de aplicaciones móviles a testers antes de su lanzamiento oficial en las tiendas de aplicaciones.
* **GIT**: Control de versiones y repositorio del proyecto.

# Requerimientos funcionales

## Tabla de características de la Norma ISO/IEC 25010

| **Características** | **Subcaracterísticas** | **Definición** |
| --- | --- | --- |
| **Adecuación funcional** | Completitud funcional | El sistema cubre todas las funciones requeridas. |
| Corrección funcional | Las funciones proporcionan resultados correctos. |
| Pertinencia funcional | Las funciones son apropiadas para las tareas. |
| **Eficiencia de desempeño** | Comportamiento temporal | El tiempo de respuesta y procesamiento es adecuado. |
| Utilización de recursos | El uso de los recursos es óptimo. |
| Capacidad | El sistema maneja la carga esperada de usuarios y datos. |
| **Compatibilidad** | Coexistencia | El sistema puede operar con otros productos de software. |
| Interoperabilidad | El sistema puede intercambiar información con otros sistemas. |
| **Usabilidad** | Reconocimiento de facilidad de uso | Es fácil aprender y operar el sistema. |
| Aprendizaje | El sistema es fácil de aprender. |
| Operabilidad | El sistema es fácil de usar en la práctica. |
| Protección contra errores de usuario | Ayuda a los usuarios a prevenir errores. |
| Estética de la interfaz | La interfaz es atractiva y clara. |
| Accesibilidad | El sistema es accesible para todos los usuarios, incluidos aquellos con discapacidades. |
| **Fiabilidad** | Madurez | El sistema tiene pocas fallas. |
| Disponibilidad | El sistema está disponible cuando se necesita. |
| Tolerancia a fallos | El sistema sigue funcionando bajo condiciones adversas. |
| Recuperabilidad | El sistema puede recuperarse tras una falla. |
| **Seguridad** | Confidencialidad | Protege los datos contra accesos no autorizados. |
| Integridad | Previene modificaciones no autorizadas a los datos. |
| No repudio | Los usuarios no pueden negar sus acciones. |
| Autenticidad | Verifica la identidad de los usuarios. |
| Responsabilidad | Permite rastrear las acciones de los usuarios. |
| **Mantenibilidad** | Modularidad | El sistema puede ser dividido en componentes independientes. |
| Reusabilidad | Los componentes del sistema pueden reutilizarse en otros sistemas. |
| Analizabilidad | Es fácil identificar las causas de los fallos. |
| Modificabilidad | El sistema puede modificarse fácilmente. |
| Probabilidad de prueba | Es fácil probar el sistema para asegurar su funcionamiento. |
| **Portabilidad** | Adaptabilidad | El sistema puede ser adaptado para diferentes entornos. |
| Instalabilidad | El sistema puede ser instalado fácilmente. |
| Sustituibilidad | Es fácil reemplazar el sistema por otro. |

## Tabla de requerimientos funcionales

| **ID** | **Descripción del requerimiento** | **Dependencia** | **Características** | **Subcaracterísticas** | **Gestión según ISO** |
| --- | --- | --- | --- | --- | --- |
| **RF1** | Los usuarios deben poder registrarse e iniciar sesión utilizando su correo electrónico o autenticación a través de redes sociales. | Firebase Authentication | Adecuación funcional | Completitud funcional, Corrección funcional | Asegurar que el sistema cumpla con las funciones de autenticación utilizando pruebas funcionales. |
| **RF2** | Los usuarios deben practicar señas en tiempo real y recibir validación a través de IA, calificando la precisión de las señas. | TensorFlow Lite | Adecuación funcional | Pertinencia funcional | Evaluar la precisión de la IA en validar correctamente las señas según especificaciones funcionales. |
| **RF3** | El sistema debe enviar notificaciones personalizadas, recordatorios y alertas sobre nuevos contenidos o actualizaciones. | Firebase Cloud Messaging | Adecuación funcional | Completitud funcional | Validar que las notificaciones se envían adecuadamente a los usuarios mediante pruebas funcionales. |
| **RF4** | Los usuarios deben poder acceder y visualizar su historial de progreso en el aprendizaje de señas, incluyendo estadísticas detalladas. | Firebase Firestore | Adecuación funcional | Pertinencia funcional | Verificar la precisión de los datos históricos y estadísticas mediante pruebas de funcionalidad. |
| **RF5** | La aplicación debe permitir a los usuarios seleccionar su nivel de aprendizaje (básico, intermedio, avanzado) y acceder a contenido ajustado a su nivel. | Sistema de niveles | Adecuación funcional | Completitud funcional | Probar que el sistema presenta contenido adecuado según el nivel seleccionado por el usuario. |
| **RF6** | Los usuarios deben poder reportar errores o problemas con los ejercicios o juegos de aprendizaje a través de un sistema de retroalimentación dentro de la app. | Sistema de reporte, Firebase Cloud Messaging | Adecuación funcional | Completitud funcional | Verificar que los reportes se registran y gestionan adecuadamente dentro del sistema de feedback. |
| **RF7** | Debe haber una sección de tutoriales en video, clasificados por niveles de aprendizaje, disponibles para todos los usuarios. | Sistema de Feedback | Adecuación funcional | Completitud funcional | Verificar que los tutoriales se cargan correctamente y están bien clasificados. |
| **RF8** | Los usuarios deben poder ver su progreso en tiempo real mediante un dashboard de estadísticas interactivo y fácil de entender. | Firebase Cloud Storage | Usabilidad | Operabilidad, Facilidad de uso | Asegurar que el dashboard es intuitivo y que las estadísticas se actualizan en tiempo real. |
| **RF9** | Los usuarios deben poder personalizar su avatar y perfil dentro de la app para mejorar la experiencia de usuario y reflejar su identidad personal. | Firebase Firestore | Usabilidad | Operabilidad, Facilidad de uso | Evaluar la facilidad de personalización y experiencia de usuario mediante pruebas de usabilidad. |
| **RF10** | El sistema debe corregir automáticamente señas incorrectas y ofrecer sugerencias de mejora detalladas. | Sistema de Feedback | Adecuación funcional, Usabilidad | Pertinencia funcional, Operabilidad | Verificar que el sistema detecta errores y proporciona retroalimentación útil para mejorar las señas. |

# Requerimientos no funcionales

## Tabla de requerimientos no funcionales

| **ID** | **Descripción del requerimiento** | **Dependencia** | **Características** | **Subcaracterísticas** | **Gestión según ISO** |
| --- | --- | --- | --- | --- | --- |
| **RNF1** | La app debe cargarse en menos de 3 segundos en dispositivos móviles modernos. | Sistema completo | Eficiencia de desempeño | Comportamiento temporal | Probar la eficiencia de la app en diferentes dispositivos para asegurar tiempos de carga óptimos. |
| **RNF2** | El sistema debe soportar hasta 20 usuarios concurrentes sin degradación del rendimiento, asegurando la capacidad dentro de los límites de la versión gratuita de Firebase. | Backend de la app | Eficiencia de desempeño, Escalabilidad | Capacidad, Utilización de recursos | Realizar pruebas de carga para asegurar que el sistema es escalable y mantiene el rendimiento esperado dentro de los límites de Firebase Spark Plan. |
| **RNF3** | Cumplir con las normativas de seguridad como la Ley 19.628 (Artículos 3, 4 y 10), asegurando la privacidad y protección de datos, y realizar pruebas de vulnerabilidad. | Seguridad del sistema | Seguridad | Confidencialidad, Integridad | Realizar auditorías de seguridad y pruebas de penetración para verificar el cumplimiento de la ley y proteger los datos personales. |
| **RNF4** | Las actualizaciones de la app deben realizarse con un mantenimiento programado en la madrugada (Chilena) con un máximo de 4 horas. | Sistema completo | Mantenibilidad | Modificabilidad, Disponibilidad | Asegurar procesos de actualización continua mediante pruebas de mantenimiento y disponibilidad del sistema. |
| **RNF5** | El sistema debe realizar un respaldo automático de la base de datos cada 24 horas. | Base de datos | Fiabilidad | Disponibilidad, Recuperabilidad | Verificar que los respaldos automáticos se realizan de forma correcta y segura. |
| **RNF6** | La aplicación debe cargar todos los recursos multimedia (videos, imágenes, audios) en menos de 5 segundos. | Sistema multimedia | Eficiencia de desempeño | Comportamiento temporal | Realizar pruebas de rendimiento para garantizar la carga rápida de los recursos multimedia. |
| **RNF7** | La solución debe ser escalable para soportar un número creciente de usuarios conforme la aplicación se expanda. El sistema debe mantener un rendimiento óptimo incluso bajo condiciones de carga alta, asegurando tiempos de respuesta rápidos, una alta tasa de éxito en las operaciones, y un uso eficiente de los recursos. | Backend de la app | Escalabilidad | Capacidad, Utilización de recursos | Probar el sistema bajo diferentes condiciones de uso para verificar su escalabilidad. |
| **RNF8** | Las métricas de uso de la aplicación deben actualizarse en tiempo real en el dashboard de Firebase Analytics para reflejar con precisión las actividades del usuario. Las actualizaciones deben producirse con una latencia mínima, asegurando que los datos relevantes estén disponibles para análisis casi instantáneamente. | Firebase Analytics | Eficiencia de desempeño | Comportamiento temporal | Verificar que las métricas se actualizan sin retrasos mediante pruebas de monitoreo en tiempo real. |
| **RNF9** | El sistema debe ser accesible para personas con discapacidades, compatible con tecnologías asistivas (lectores de pantalla, etc.). | Frontend de la app | Usabilidad | Accesibilidad | Realizar pruebas de accesibilidad y asegurar el cumplimiento de las guías WCAG 2.1. |

# Reglas de Negocio

| **ID** | **Descripción del requerimiento** |
| --- | --- |
| **RB1** | Solo usuarios autenticados mediante correo electrónico o redes sociales pueden acceder a desafíos y validación de señas. |
| **RB2** | Toda la información personal de los usuarios, como contraseñas, datos de perfil y progreso de aprendizaje, será cifrada de extremo a extremo tanto en tránsito como en reposo. |
| **RB3** | Se debe cumplir con la política de privacidad descrita en el documento respectivo, asegurando el cumplimiento de normativas locales e internacionales, como la Ley 19.628. |

# 

# Glosario de términos

* **Firebase**: Plataforma de desarrollo de aplicaciones web y móviles creada por Google.
* **TensorFlow Lite**: Herramienta para ejecutar modelos de aprendizaje automático en dispositivos móviles.
* **Ley 19.628**: Ley chilena sobre protección de la vida privada.

# Bibliografía

PMOInformatica. (2018, abril 4). Documento de requerimientos de software. <https://www.pmoinformatica.com/2018/04/documento-de-requerimientos-de-software_37.html>