Archivo de Contexto - Aplicación Lector Global

Fecha de actualización: 22/04/2025 - 23:30

Versión de la App: 1.5

# 1. Descripción General del Proyecto

Lector Global es una aplicación educativa diseñada para mejorar la comprensión lectora en todos los idiomas. Inspirada en la filosofía de Duolingo, la app ofrece una experiencia progresiva de aprendizaje desde el nivel A1 hasta C2, de acuerdo con el Marco Común Europeo de Referencia para las Lenguas (MCER).

# 2. Estructura General del Sistema

El sistema está organizado en carpetas para separar la lógica de la aplicación, pantallas, modelos y servicios. Las pantallas están ubicadas en la carpeta `lib/screens`, que incluye las interfaces de autenticación (login y registro) y bienvenida. Los servicios (como `auth\_service.dart`) gestionan operaciones relacionadas con la autenticación.  
  
Cada pantalla está implementada en un archivo separado con clases estructuradas y comentarios detallados en español.

# 3. Autenticación

La autenticación actual permite iniciar sesión o registrarse mediante correo y contraseña. Las pantallas de login y registro están desarrolladas con validaciones básicas. Se han implementado mensajes de error específicos para indicar campos vacíos o credenciales inválidas. Firebase Authentication es el backend elegido para gestionar el acceso.

# 4. Pantallas Desarrolladas

## 4.1. WelcomeScreen

Pantalla de introducción. Muestra un mensaje de bienvenida con un ícono representativo y un botón 'Comenzar' que redirige al login. Se mejoró el diseño usando `SingleChildScrollView` y `SafeArea` para evitar problemas de visualización en distintos tamaños de pantalla.

## 4.2. LoginScreen

Incluye campos para correo electrónico y contraseña. Valida entradas vacías y muestra mensajes adecuados si no coinciden con los usuarios existentes. El botón '¿No tienes cuenta? Regístrate aquí' permite acceder a la pantalla de registro.

## 4.3. RegisterScreen

Permite el registro mediante nombre, correo y contraseña. Valida cada campo y muestra errores personalizados. Al completar el registro, muestra un `SnackBar` y vuelve automáticamente al login. Incluye un botón de carga durante el proceso.

# 5. Servicios Implementados

Se implementó `auth\_service.dart` como archivo base para la gestión de autenticación con Firebase. Este servicio se expandirá más adelante para incluir autenticación con Google y Facebook, manejo de sesiones, y recuperación de contraseña.

# 6. Control de Versiones

Se utilizan commits organizados por versión (ej. 'Versión 1.5 - Validación de login con mensajes de error implementada') y el proyecto está gestionado en GitHub. Cada versión refleja una funcionalidad clave implementada o mejorada, como validación visual, estructura de navegación y mensajes informativos.

# 7. Tareas Pendientes

- Reparar error de compilación en Windows causado por dependencias conflictivas de C++ (firebase\_auth)  
- Implementar autenticación con Google y Facebook  
- Agregar navegación condicional según sesión activa  
- Crear pantalla principal (dashboard)  
- Validación de formularios con más profundidad (correo duplicado, errores específicos de Firebase)

**🟩 Restauración del sistema de internacionalización (Versión 2.3)** o similar según tu estructura:

📌 **Versión 2.3 - Eliminación del idioma coreano (ko) por incompatibilidad de codificación - 29/04/2025 - 00:06**

Después de detectar fallos de codificación en el archivo de localización correspondiente al idioma coreano (ko), se tomó la decisión técnica de eliminar temporalmente dicho idioma del sistema de internacionalización. Este cambio se realizó en el commit aaad2e4ff771cf09e35707ce2549cbcf10f3e245, el cual fue seleccionado como punto estable de restauración para mantener la funcionalidad fluida del selector dinámico de idiomas en la app.

A partir de este commit:

* El sistema de internacionalización vuelve a ser completamente funcional.
* El selector de idiomas muestra correctamente los 8 idiomas restantes.
* Se verificó manualmente la ejecución de la app en navegador (flutter run -d chrome) confirmando la correcta visualización y traducción.
* La rama restauracion-internacionalizacion-sin-coreano fue creada y luego fusionada exitosamente en main.
* El repositorio remoto fue forzado a este nuevo estado con git push origin main --force.

🟢 Este estado queda registrado como versión estable funcional **sin el idioma coreano**, y sirve de base para nuevas adiciones multilingües o ajustes finos posteriores.

**🟩 Validación del Formulario de Registro**

📌 **Fecha:** 29 de abril de 2025  
**Commit:** 9a0b5ab

**Descripción técnica del avance:**

Se mejoró la pantalla de registro con una validación profesional y visualmente elegante, adecuada para ser utilizada como plantilla base. Se aplicaron los siguientes ajustes y mejoras:

* Validación completa de campos obligatorios: nombre, correo electrónico y contraseña.
* Verificación del formato del correo mediante expresión regular.
* Requisitos de seguridad para contraseña (mínimo 6 caracteres).
* Retroalimentación visual clara y moderna (errores suaves y comprensibles).
* Botón de registro desactivado si hay errores en el formulario.
* Íconos de visibilidad de contraseña e interacción en tiempo real.
* Código documentado en español y estructurado de forma modular y mantenible.

El formulario cumple estándares de calidad profesional, listo para producción y escalabilidad, y deja la base sólida para futuras integraciones como validación en tiempo real, campos adicionales o autenticación extendida.

**🟩 Navegación Condicional basada en Autenticación**

📌 **Fecha:** 29 de abril de 2025  
**Archivo principal:** main.dart  
**Pantalla controladora:** RootScreen  
**Commit relacionado:** *(estructura ya incluida previamente)*

**Descripción del avance:**

Se implementó una arquitectura profesional que redirige automáticamente a los usuarios según su estado de sesión en Firebase:

* Si el usuario **ya inició sesión**, la app lo redirige automáticamente al DashboardScreen tras cargar la aplicación.
* Si **no hay sesión activa**, se muestra la pantalla de bienvenida (WelcomeScreen).
* Se utiliza un StreamBuilder que escucha FirebaseAuth.instance.authStateChanges(), garantizando una reacción en tiempo real a cualquier cambio de autenticación.
* Esta lógica está encapsulada en una clase especial llamada RootScreen, asignada como valor del parámetro home del MaterialApp.
* El diseño es altamente escalable, ideal para producción y para ser usado como plantilla base.

Este enfoque mejora drásticamente la experiencia de usuario, profesionaliza la navegación y deja la base lista para futuras funcionalidades como recuperación de sesión, tokens persistentes y usuarios recordados.

**🟩 Corrección de carga de imágenes en botones sociales**

📌 **Fecha:** 29 de abril de 2025  
**Commit:** b0f7f14  
**Archivos:** assets/images/google.png, assets/images/facebook.png

**Descripción del avance:**

Se solucionó el error visual en la pantalla de inicio de sesión causado por la falta de imágenes de los botones de autenticación con Google y Facebook. Para ello se realizaron los siguientes pasos:

* Se creó (o verificó) la carpeta assets/images/.
* Se descargaron y renombraron correctamente los íconos oficiales de Google y Facebook en formato .png.
* Se almacenaron en la ruta esperada por Flutter.
* Se verificó que las imágenes cargaran correctamente desde el código sin errores.
* Se eliminó un archivo mal nombrado (google\_logo.png .png) que interfería con la carga de recursos.
* Se volvió a compilar con éxito y se visualizó la interfaz funcionando de forma elegante y profesional.

Esta corrección mejora la estética del login y completa la estructura visual necesaria para un sistema de autenticación social.