



Énfasis en Programación Móvil

Entrega 1 – Semana 3

Documento Diseño de Aplicación

Tutor

Víctor Fabián Castro López

Presentado por

Jacobo Marín Gómez

María Lizarazo Ríos

Andrés Joan Mantilla Rodríguez

Sergio Alejandro Manrique

Facultad de Ingeniería, diseño e innovación

Índice

Proyecto Recordari.....	3
Justificación	4
Objetivos.....	4
Objetivo General.....	4
Objetivos específicos.....	4
Requerimientos	5
Funcionales	5
No funcionales	7
Flujo de Navegación	9
Mockups / Wireframes	10
Referencias	16

Proyecto Recordari

Justificación

Crearemos una aplicación llamada Recordari la cual surge como una herramienta práctica para estudiantes y personas que deseen memorizar información de forma sencilla por medio de la repetición. Muchas veces, las técnicas de estudio tradicionales no son suficientes para retener conceptos o ideas a largo plazo; es necesaria la repetición para poder asimilar y grabar en el cerebro lo que queremos aprender. Para eso, Recordari utiliza el método de flashcards (tarjeta de memoria), el cual ha demostrado ser una de las estrategias más efectivas para el aprendizaje progresivo.

Esta aplicación se desarrollará en Flutter, lo que permite crear una experiencia multiplataforma (Android - iOS) con un solo código base, garantizando accesibilidad y rendimiento sin requerir infraestructura de backend. Su implementación como app híbrida facilita el acceso desde diferentes dispositivos, manteniendo una interfaz amigable, moderna y totalmente funcional sin conexión a internet.

Objetivos

Objetivo General

Desarrollar una aplicación móvil multiplataforma que permita a los usuarios crear, organizar, y estudiar tarjetas de memoria de manera interactiva y sin conexión a internet, para mejorar su capacidad de retención y aprendizaje autónomo.

Objetivos específicos

1. Diseñar una interfaz intuitiva y agradable para la creación y gestión de tarjetas de memoria.
2. Implementar una base de datos local (SQLite o Hive) para el almacenamiento de información sin necesidad de conexión a internet.
3. Desarrollar un flujo de estudio basado en repeticiones y autoevaluación.
4. Permitir la organización de tarjetas por temas o categorías.
5. Incorporar métricas simples de progreso (porcentaje de aciertos, números de tarjetas repasadas).

Requerimientos

Funcionales

RF1: Crear, editar y eliminar tarjetas de memoria.

1. Crear tarjetas de memoria

Descripción: Como usuario, quiero crear tarjetas de memoria, para guardar información qué deseo estudiar.

Criterios de aceptación:

- a. El usuario puede ingresar texto en ambos lados de la tarjeta (pregunta y respuesta).
- b. La tarjeta se guarda correctamente en la base de datos local.
- c. El sistema confirma la creación exitosa.

2. Editar tarjetas de memoria

Descripción: Como usuario, quiero editar mis tarjetas de memoria, para actualizar o corregir la información guardada.

Criterios de aceptación:

- a. El usuario puede modificar el texto de una tarjeta existente.
 - b. Los cambios se guardan correctamente.
 - c. La tarjeta actualizada reemplaza la versión anterior.
3. Eliminar tarjetas de memoria

Descripción: Como usuario, quiero eliminar tarjetas de memoria, para mantener mi colección organizada y solo con la información que necesito.

Criterios de aceptación:

- a. El usuario puede seleccionar una tarjeta y eliminarla.
- b. La aplicación solicita confirmación antes de eliminar.
- c. La tarjeta desaparece de la lista después de ser eliminada.

RF2: Organizar tarjetas en categorías o conjuntos.

Descripción: Como usuario, quiero agrupar mis tarjetas en categorías o conjuntos, para poder estudiar por temas.

Criterios de aceptación:

- a. El usuario puede crear categorías o conjuntos con un nombre personalizado.
- b. Las tarjetas pueden asignarse o moverse entre conjuntos.
- c. Al seleccionar un conjunto, solo se muestran las tarjetas pertenecientes a él.

RF3: Mostrar tarjetas aleatoriamente o en orden.

Descripción: Como usuario, quiero elegir entre mostrar las tarjetas en orden o al azar, para variar mi método de estudio.

Criterios de aceptación:

- a. La aplicación ofrece la opción de modo “ordenado” o “aleatorio”.
- b. Las tarjetas se muestran según el modo seleccionado.

RF4: Registrar progreso básico del usuario.

Descripción: Como usuario, quiero que la aplicación registre mi progreso, para saber cuántas tarjetas he aprendido.

Criterios de aceptación:

- a. La app guarda el número de tarjetas revisadas y correctas.
- b. Se muestra un resumen del progreso (porcentaje o conteo).
- c. El progreso se mantiene guardado al cerrar la aplicación.

RF5: Funcionar completamente sin conexión a internet.

Descripción: Como usuario, quiero usar la aplicación sin conexión, para estudiar en cualquier momento y lugar.

Criterios de aceptación:

- a. Todas las funciones principales (crear, editar, eliminar, estudiar) funcionan sin internet.
- b. Los datos se guardan localmente.

No funcionales

RNF1: La aplicación debe ser ligera (menos de 50 MB).

Descripción: Como usuario, quiero que la aplicación sea ligera, para que no ocupe mucho espacio en mi dispositivo.

Criterios de aceptación:

- a. El tamaño total de la aplicación no supera los 50 MB.

RNF2: Debe funcionar en Android y iOS.

Descripción: Como usuario, quiero poder usar la app en Android y iOS, para acceder a mis tarjetas desde distintos dispositivos

Criterios de aceptación:

- a. La aplicación se instala y ejecuta correctamente en ambos sistemas.

RNF3: Debe mantener una interfaz fluida y responsive.

Descripción: Como usuario, quiero que la interfaz sea fluida y responsive, para tener una experiencia agradable al usar la app.

Criterios de aceptación:

- a. Las animaciones y transiciones son suaves.
- b. No hay retrasos o bloqueos al interactuar.

RNF4: Los datos deben permanecer guardados incluso tras cerrar la aplicación.

Descripción: Como usuario, quiero que mis datos se guarden incluso al cerrar la aplicación, para no perder mi progreso.

Criterios de aceptación:

- a. Los datos se guardan de forma local y segura.
- b. Al abrir la aplicación, las tarjetas y el progreso aparecen intactos.

RNF5: Tiempos de carga inferior a 3 segundos.

Descripción: Como usuario, quiero que la app cargue rápidamente, para empezar a estudiar sin demoras.

Criterios de aceptación:

- a. El tiempo de inicio de la app es inferior a 3 segundos.

Flujo de Navegación

1. Pantalla de inicio: listas de conjuntos (temas) creados.
2. Pantalla de tarjetas: muestra las tarjetas del conjunto elegido.
3. Pantalla de estudio: permite repasar tarjetas con acción de “mostrar respuestas” o “marcar como aprendida”
4. Pantalla de progreso: estadísticas simples del usuario.

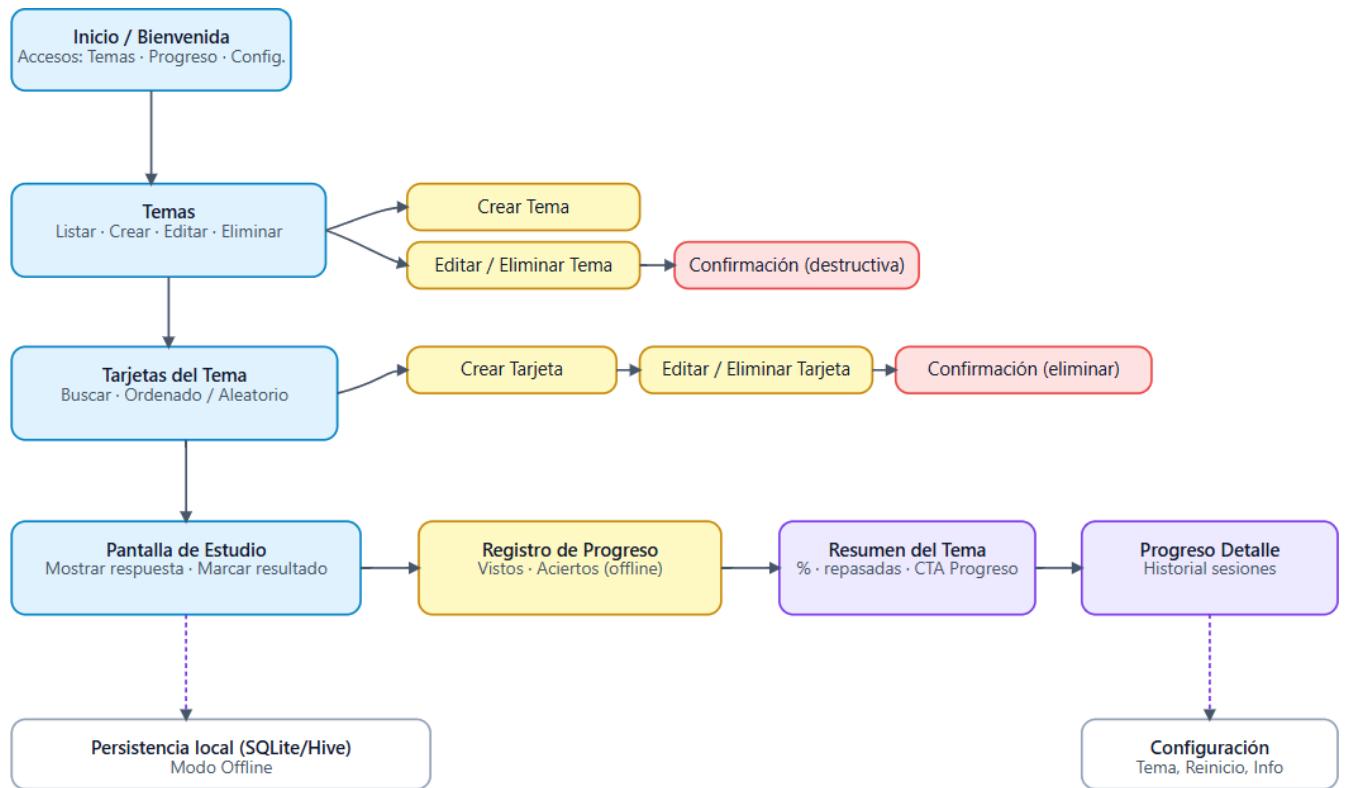


Imagen 1 – Diagrama (Flujo de navegación)

Mockups / Wireframes

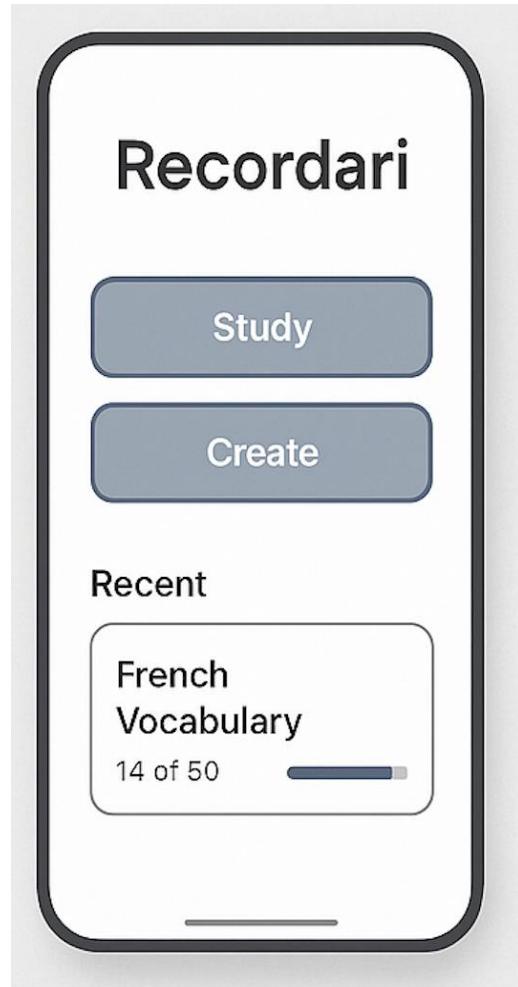


Imagen 2 – Pantalla de inicio:

Vista principal de *Recordari* con opciones para estudiar o crear conjuntos de tarjetas, y una sección de progreso reciente.

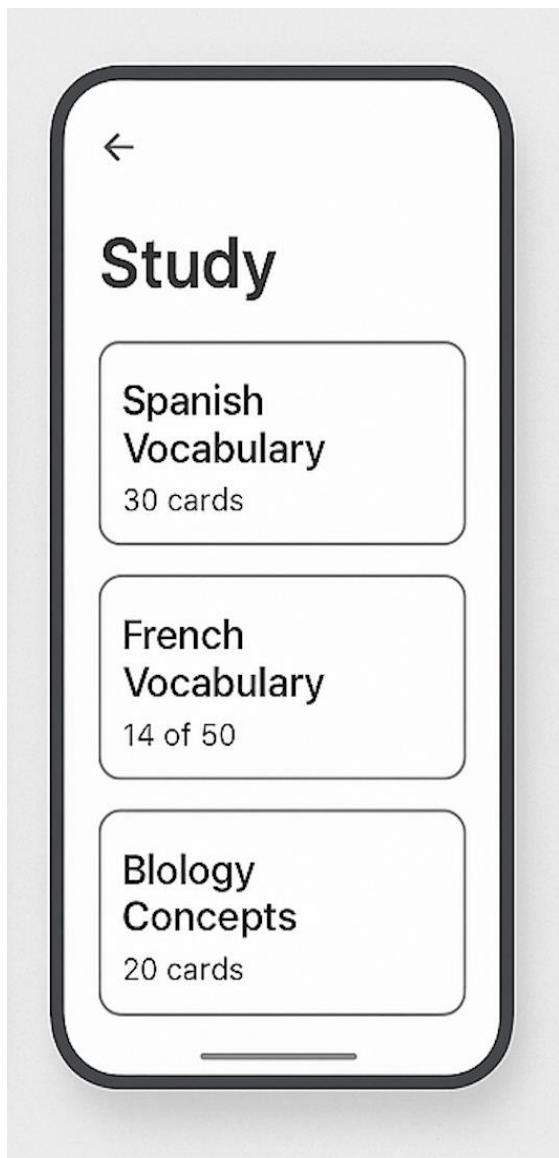


Imagen 3 – Lista de conjuntos:

Pantalla donde el usuario visualiza los temas o categorías disponibles, junto con la cantidad de tarjetas por conjunto.

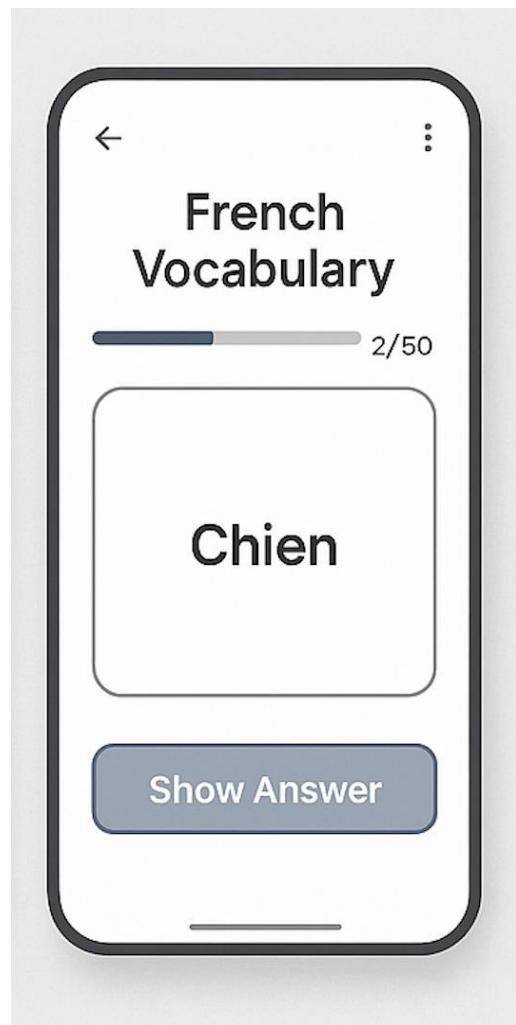


Imagen 4 – Modo de estudio (frontal):

Interfaz del modo de repaso mostrando la palabra o pregunta del lado frontal de la tarjeta y un botón para revelar la respuesta.

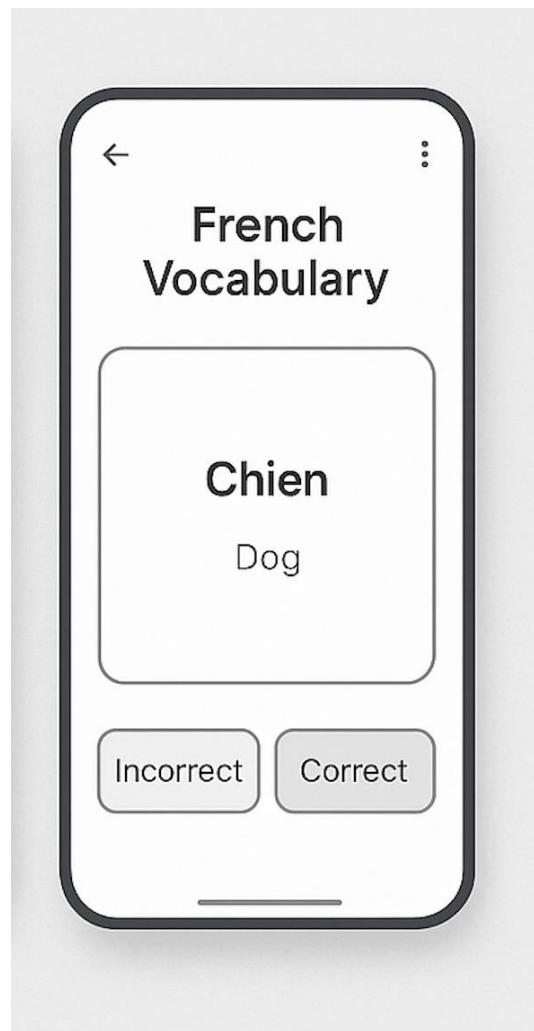


Imagen 5 – Modo de estudio (reverso):

Tarjeta volteada que muestra la respuesta, con botones para marcar si se recordó o no, y una barra de progreso en la parte superior.



Imagen 6 - Introducción App (comercial)



Imagen 7 – Tarjetas y funcionalidad (comercial)

Agrupa tus tarjetas y crea nuevas preguntas.

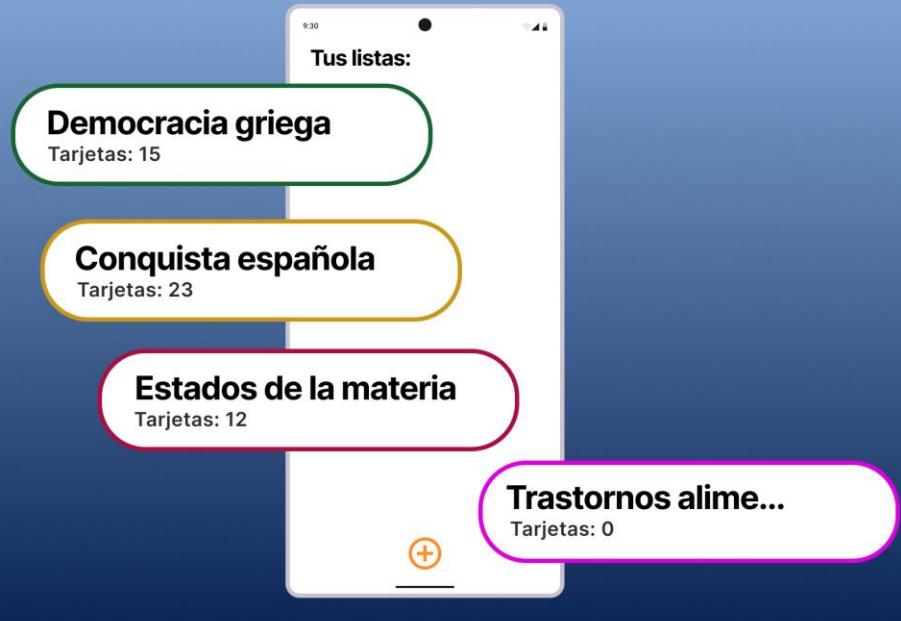


Imagen 8 - Agrupación de temas (comercial)

Referencias

Anki. (n.d.). *Anki – powerful, intelligent flashcards*. Recuperado de <https://apps.ankiweb.net/>

Flutter. (n.d.). *Build apps for any screen*. Google Developers. Recuperado de <https://flutter.dev/>