

Basándome en cada detalle que hemos construido y refinado juntos, aquí tienes un prompt completo y detallado que daría origen a la aplicación que tienes ahora mismo.

---

## Prompt para Generar una Aplicación Web Educativa de Quizzes

### ## Objetivo Principal

Crear una aplicación web completa para practicar matemáticas, diseñada como una herramienta de aprendizaje interactiva. La aplicación debe ser moderna, modular y fácilmente actualizable. Se desarrollará utilizando **React con Vite**, JavaScript (ES6+), HTML5 y CSS3, siguiendo las mejores prácticas de desarrollo.

### ## Pila Tecnológica (Stack)

- **Framework:** React 18+
- **Entorno de Desarrollo:** Vite
- **Lenguaje:** JavaScript (JSX)
- **Estilos:** CSS3 (sin frameworks de UI, estilos puros en un archivo index.css)

### ## Estructura de Datos (archivo public/reactivos.json)

El contenido de la aplicación debe estar completamente separado de la lógica. Se creará un archivo reactivos.json con la siguiente estructura:

- Un objeto principal donde cada **clave** es el nombre de un tema (ej. "Fracciones y Decimales").
- El **valor** de cada clave será un array de 10 objetos, donde cada objeto representa un reactivo.
- Cada reactivo debe tener los siguientes campos:
  - id: Un número único.
  - pregunta: El texto de la pregunta.
  - imagen (opcional): La ruta a una imagen guardada en la carpeta public.
  - opciones: Un array de 4 objetos, cada uno con id (ej. "a") y texto.
  - respuestaCorrecta: El id de la opción correcta (ej. "b").
  - explicacion (opcional): Un texto que explique la solución, para ser usado en la revisión.

### ## Flujo de la Aplicación y Componentes (4 Vistas Principales)

La aplicación funcionará como una "Single Page Application" gestionando cuatro vistas principales: Menú, Quiz, Resultados y Revisión.

## 1. Pantalla de Inicio (Menú Principal)

- **Branding:** En la parte superior, debe mostrar un logotipo circular, el título "I learn math" y un subtítulo "Creado por: Jairo Farrera". Todos los elementos deben estar centrados.
- **Entrada de Usuario:** Justo antes de la lista de temas, debe haber un campo de texto que pida al alumno su nombre.
- **Selección de Temas:**
  - Los botones para cada tema deben generarse dinámicamente leyendo las claves del archivo reactivos.json.
  - Estos botones deben estar **deshabilitados** hasta que el usuario haya escrito algo en el campo de nombre.
  - Al hacer clic en un tema, la aplicación debe pasar a la vista del Quiz.

## 2. Pantalla del Quiz

- **Navegación:** Un botón de "← Regresar al Menú" debe estar siempre visible en la parte superior.
- **Aleatoriedad:** Las 10 preguntas del tema seleccionado deben presentarse en un **orden aleatorio** cada vez que se inicia el quiz.
- **Feedback Mínimo:** Al seleccionar una opción, **no se debe mostrar si es correcta o incorrecta**. La aplicación simplemente avanzará automáticamente a la siguiente pregunta para evitar que se compartan las respuestas.
- **Contador de Aciertos:** Debe llevar un conteo visible de los aciertos del usuario.
- **Sonido:** Al hacer clic en cualquier opción de respuesta, se debe reproducir un sonido (sonidito.mp3).
- **Mejoras de Experiencia (Nivel 1):**
  - **Barra de Progreso:** Incluir una barra visual que se llene a medida que el usuario avanza en las preguntas.
  - **Animaciones Suaves:** Implementar una transición de fundido (fade-out/fade-in) entre cada pregunta para que la experiencia sea más fluida.
- **Registro de Datos:** La aplicación debe guardar un historial de las respuestas del usuario para usarlo en la pantalla de revisión.

## 3. Pantalla de Resultados

- Aparece automáticamente al responder la última pregunta.
- **Personalización:** Mostrará un saludo personalizado usando el nombre ingresado (ej. "¡Felicidades, Jairo!").

- **Puntuación:** Mostrará la puntuación final (ej. "Tu puntuación final es: 9 de 10").
- **Link para Compartir:** Generará un link único y compartible que contenga los siguientes parámetros en la URL: ?alumno=[nombre]&tema=[nombreTema]&aciertos=[numero]&total=10. Un botón permitirá copiar este link al portapapeles.
- **Navegación Post-Quiz:** Ofrecerá dos botones claros:
  - Un botón "Volver al Menú Principal".
  - Un botón "Evaluación Formativa" para pasar a la pantalla de revisión.

#### 4. Pantalla de Revisión ("Evaluación Formativa")

- **Funcionalidad (Nivel 2):**
  - Mostrará la lista completa de las 10 preguntas del quiz recién terminado.
  - Para cada pregunta, debe indicar visualmente:
    - La **respuesta correcta** (marcada permanentemente en verde).
    - La **respuesta que dio el usuario** (si fue incorrecta, debe marcarse en rojo y tacharse).
  - Si una pregunta tiene una "explicacion" en el JSON, se mostrará un botón "¿Por qué?" que, al ser presionado, revele el texto de la explicación. El botón cambiará a "Ocultar Explicación".
- **Navegación:** También incluirá un botón para "← Regresar al Menú".

#### ## Requisitos Adicionales

- **Código Modular:** El código debe estar organizado en componentes de React reutilizables (ej. App.jsx, Quiz.jsx, PantallaFinal.jsx, PantallaRevision.jsx, OpcionTexto.jsx).
- **Responsividad:** El diseño debe ser adaptable y funcionar correctamente tanto en dispositivos móviles como en ordenadores de escritorio.