

Aplicación Comunicador para Niño con Autismo

1. Stack Tecnológico Recomendado

- 1 Frontend: React (Web) + React Native (Móvil)
- 2 Backend: Node.js + Express
- 3 Base de datos: PostgreSQL
- 4 Contenedores: Docker para despliegue uniforme
- 5 Empaquetado multiplataforma: Electron (Windows, Linux, Mac) y Capacitor/Expo (Android/iOS)

2. Estructura del Proyecto

comunicador-app/	Proyecto raíz
backend/	Node.js + Express
■■■■ src/models/	Esquemas DB (acciones, usuarios)
■■■■ src/routes/	Rutas API
■■■■ src/controllers/	Lógica de negocio
■■■■ src/index.js	Punto de entrada
frontend/	React (web) + Capacitor/Electron
■■■■ public/	Pictogramas estáticos
■■■■ src/components/	Botones, tarjetas, tablero
■■■■ src/pages/	Pantallas principales
■■■■ src/hooks/	Lógica de estado (ej: frases)
■■■■ src/App.jsx	Componente principal
mobile/	React Native (Expo)
■■■■ App.js	Punto de entrada móvil
■■■■ components/	Componentes móviles
database/	Migraciones o seeds
docs/	Documentación
docker-compose.yml	Levanta backend + frontend + DB fácil

3. Funcionalidades Clave

- Tablero de pictogramas inicial con acciones básicas (comer, beber, baño, plaza, dormir, jugar).
- Construcción de frases mediante la selección de pictogramas.
- Reproducción de frases completas mediante Text-to-Speech (TTS).
- Modo edición para familiares: agregar nuevas acciones, imágenes o fotos reales.
- Modo offline: la aplicación debe poder funcionar sin conexión a internet.

4. Funcionalidades Extra

- Historial de frases más usadas.
- Colores personalizables para preferencias sensoriales.

- Botón SOS para enviar notificación a un familiar.
- Multiusuario para distintos niños o necesidades.

5. Plan de Desarrollo Propuesto

- 1 Definir conjunto inicial de pictogramas y frases (mínimo 6).
- 2 Implementar backend en Node.js con conexión a PostgreSQL.
- 3 Desarrollar frontend en React para tablero de pictogramas.
- 4 Integrar Text-to-Speech para reproducción de frases.
- 5 Agregar soporte multiplataforma con Electron y React Native.