¡Excelente decisión! Aprenderás mucho desarrollando esta app desde cero. Aquí tienes un \*\*plan paso a paso\*\* con herramientas y tecnologías recomendadas (enfocadas en aprendizaje y escalabilidad):

---

### \*\*1. Definir el stack tecnológico (recomendado para principiantes/intermedios)\*\*

- \*\*Frontend (Interfaz de usuario)\*\*:

- \*\*Web\*\*: React.js (con Vite.js para configuración rápida) o Next.js (si quieres SSR + SEO).

- \*\*Móvil\*\*: React Native (reutilizarás lógica de React.js) + Expo (para simplificar el desarrollo).

- \*\*Backend (Lógica y base de datos)\*\*:

- \*\*API REST\*\*: Node.js + Express (fácil para empezar) o Python + Flask (si prefieres Python).

- \*\*Base de datos\*\*: Firebase (si quieres algo rápido) o PostgreSQL/MongoDB (para bases de datos relacionales o NoSQL).

- \*\*Diseño\*\*:

- \*\*UI/UX\*\*: Figma (para prototipar antes de codear).

- \*\*Estilos\*\*: Tailwind CSS (rápido y moderno) o CSS Modules.

---

### \*\*2. Plan de desarrollo paso a paso\*\*

#### \*\*Fase 1: Prototipo y diseño (1-2 semanas)\*\*

- Diseña en Figma las pantallas principales:

- Home (resultados recientes, próximos partidos).

- Calendario de partidos.

- Noticias.

- Perfil de cada equipo (estadísticas, jugadores).

- Define la estructura de datos (ej.: cómo almacenar partidos, equipos, noticias).

#### \*\*Fase 2: Backend (2-3 semanas)\*\*

1. \*\*Configura el servidor\*\*:

- Crea una API con Node.js + Express (ejemplo básico):

```javascript

const express = require('express');

const app = express();

app.get('/partidos', (req, res) => {

res.json([{ equipo1: "Palacios CF", equipo2: "UD Mosqueo", resultado: "2-1" }]);

});

app.listen(3000, () => console.log('API corriendo en http://localhost:3000'));

```

2. \*\*Base de datos\*\*:

- Usa Firebase Firestore (NoSQL) o PostgreSQL (SQL) para guardar:

- Partidos (fecha, equipos, resultado).

- Noticias (título, contenido, imagen).

- Tabla de posiciones.

#### \*\*Fase 3: Frontend Web (3-4 semanas)\*\*

1. \*\*Estructura con React.js\*\*:

- Crea un proyecto con `npm create vite@latest` (elige React).

- Organiza componentes:

- `Partidos.jsx`: Muestra lista de partidos.

- `Noticias.jsx`: Cards de noticias.

- `Navbar.jsx`: Navegación entre equipos.

2. \*\*Consume tu API\*\*:

- Usa `fetch` o Axios para obtener datos del backend:

```javascript

const [partidos, setPartidos] = useState([]);

useEffect(() => {

fetch('http://localhost:3000/partidos')

.then(res => res.json())

.then(data => setPartidos(data));

}, []);

```

#### \*\*Fase 4: App Móvil (2-3 semanas)\*\*

1. \*\*Configura React Native + Expo\*\*:

- Instala Expo CLI: `npm install -g expo-cli`.

- Crea proyecto: `expo init nombre-app`.

- Reutiliza lógica de React.js (adaptando estilos).

2. \*\*Navegación\*\*:

- Usa `react-navigation` para moverte entre pantallas.

#### \*\*Fase 5: Features avanzados (opcionales)\*\*

- \*\*Autenticación\*\*: Firebase Auth (para que usuarios comenten noticias).

- \*\*Notificaciones push\*\*: Firebase Cloud Messaging (FCM).

- \*\*Web scraping\*\*: Si no hay API, usa Python (BeautifulSoup) para extraer datos de webs de federaciones.

---

### \*\*3. Herramientas de apoyo\*\*

- \*\*Control de versiones\*\*: Git + GitHub (obligatorio para colaborar o guardar tu progreso).

- \*\*Deploy\*\*:

- \*\*Web\*\*: Vercel o Netlify (gratis para frontend).

- \*\*Backend\*\*: Render.com o Railway (gratis para Node.js/Python).

- \*\*Móvil\*\*: Expo App Store (publica tu APK/IPA).

---

### \*\*4. Recursos para aprender\*\*

- \*\*React.js\*\*: [Documentación oficial](https://react.dev/).

- \*\*React Native\*\*: [Tutorial en freeCodeCamp](https://www.youtube.com/watch?v=obH0Po\_RdWk).

- \*\*Node.js/Express\*\*: [The Net Ninja (YouTube)](https://www.youtube.com/playlist?list=PL4cUxeGkcC9jsz4LDYc6kv3ymONOKxwBU).

- \*\*Base de datos\*\*: [Firebase Firestore Tutorial](https://firebase.google.com/docs/firestore).

---

### \*\*5. Consejos clave\*\*

1. \*\*Empieza pequeño\*\*: Primero haz que muestre un partido estático, luego conecta el backend.

2. \*\*Divide el proyecto en issues\*\*: Usa GitHub Issues para organizar tareas (ej.: "Crear componente de resultados").

3. \*\*No reinventes la rueda\*\*: Usa librerías como:

- `date-fns` (para manejar fechas).

- `react-icons` (íconos gratuitos).

4. \*\*Pide feedback\*\*: Comparte avances en foros como Reddit (r/learnprogramming) o con la comunidad de tu pueblo.

---

Si en algún paso necesitas ayuda con código específico (ej.: cómo conectar Firebase a React), dime y te doy ejemplos detallados. ¡Es un proyecto fantástico para tu portafolio! 💻⚽