

React Native: Introducción y Setup

Diego Muñoz

09 de noviembre de 2025

Introducción al entorno móvil moderno

- React Native es un **framework de Meta** para desarrollar aplicaciones móviles nativas usando **JavaScript y React**.
- Permite construir para **Android y iOS** desde un mismo código base.
- Ofrece rendimiento nativo, acceso a APIs del sistema y una curva de aprendizaje rápida para quienes ya dominan React.

Es el puente entre el desarrollo web y el desarrollo móvil profesional.

Rol e importancia de React Native

Contexto y propósito

- Antes, Android y iOS requerían proyectos separados: Java/Kotlin y Swift/Objective-C.
- React Native unifica ambos mundos bajo una misma lógica React.
- Usa **componentes nativos reales**, no un navegador embebido.

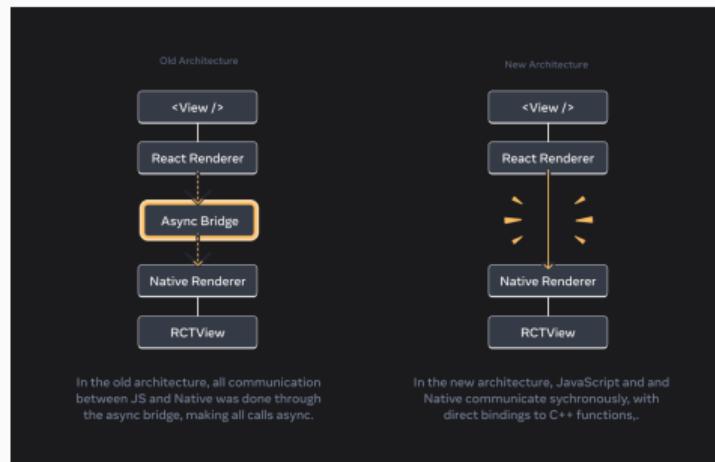


Figure 1: Comunicación JavaScript - Nativo

Cómo se diferencia del desarrollo web

- En la web, React manipula el **DOM del navegador**.
- En móvil, React Native comunica su estructura al **motor nativo** de Android/iOS.
- No existe HTML ni CSS; se usan **componentes nativos y estilos en objetos JavaScript**.
- React Native combina la experiencia del desarrollador web con el rendimiento nativo.

Requisitos técnicos

- Node.js ≥ 24.x
- Yarn (el gestor de dependencias del ramo)
- Teléfono Android o iPhone con la app **Expo Go** instalada
- Conexión estable en la misma red Wi-Fi que el computador
Descarga Expo Go:

[Play Store](#)

[App Store](#)

Ventajas y funcionamiento

- **Expo** simplifica la creación y ejecución de apps sin instalar SDKs nativos.
- Permite crear, ejecutar y depurar en tiempo real.
- El comando `yarn start` abre un servidor con un **código QR** para probar la app en el teléfono.
- Basado en la documentación oficial: docs.expo.dev/get-started/set-up-your-environment

Comandos iniciales

```
yarn create expo <nombre-del-proyecto>  
cd <nombre-del-proyecto>  
yarn start
```

- Abre el panel de desarrollo de Expo en el navegador.
- Escanea el QR con la app **Expo Go**.

Ajuste necesario para React Native

- React Native y Expo no funcionan correctamente con **Plug'n'Play (PnP)** de Yarn 2+.
- Se debe forzar el linker tradicional agregando un archivo **.yarnrc.yml** con el siguiente contenido:

nodeLinker: node-modules

- En **Windows**:
 1. Abre PowerShell o CMD en la carpeta del proyecto.
 2. Crea (si no existe) el archivo **.yarnrc.yml**.
 3. Agrega la línea **nodeLinker: node-modules**.
 4. Ejecuta **yarn install**.

Esto evita que Expo o React Native “no encuentren” las dependencias, un error común con Yarn moderno.

Estructura base del proyecto

Archivos generados

- `App.js` – punto de entrada de la aplicación.
- `/assets` – imágenes, fuentes e íconos.
- `package.json` – scripts y dependencias.
- `.expo` – metadatos locales.

Esta estructura se crea automáticamente al iniciar el proyecto.

Primer renderizado en React Native

Estructura básica del componente

```
import { Text, View, StyleSheet } from "react-native";

export default function App() {
  return (
    <View style={styles.container}>
      <Text style={styles.title}>Hello React Native</Text>
      <Text style={styles.subtitle}>Running on my phone</Text>
    </View>
  );
}
```

- `View` reemplaza a `<div>` y `Text` a `<p>`.
- Todo el layout se basa en **componentes nativos**.

Definición con StyleSheet

```
const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: "center",
    alignItems: "center",
  },
  title: { fontSize: 28 },
  subtitle: { fontSize: 18, opacity: 0.8, marginTop: 8 },
});
```

- Estilos escritos como **objetos JS**, no como CSS.
- No se usan unidades (px, %), solo números o strings válidos.
- **StyleSheet** mejora rendimiento y validación.

Elementos fundamentales

1. Render nativo en lugar de DOM.
2. Estilos declarativos en JS.
3. Layout con **Flexbox**.
4. Carga de imágenes con `require()` o URL.
5. Librerías dedicadas para navegación y sensores.

React Native mantiene la filosofía de React, pero en el entorno móvil.

1. React Native combina React y desarrollo móvil nativo.
2. Expo ofrece un entorno rápido y multiplataforma para iniciar.
3. Con Yarn y la configuración correcta, el flujo es estable y simple.
4. Próxima clase: **estilos, Flexbox y NativeWind**.

Recursos adicionales

- Configuración oficial del entorno (Expo)
- Documentación React Native
- Documentación Expo
- Expo Go en Play Store
- Expo Go en App Store

Tip: yarn start --tunnel evita bloqueos de red o firewall institucional.