EcomLabs

GUÍA PASO A PASO

Expo Router en el SDK 56: migrar sin React Navigation

La migración de una app con Expo Router del SDK 55 al 56, en orden: inventario de imports, subir el SDK, el codemod, lo que se arregla a mano, las librerías de terceros y las pruebas en iOS y Android. Con todos los comandos y la checklist.

PARA	Apps de Expo con Expo Router en el SDK 55 que suben al SDK 56
PROBADO CON	Node 22 · expo-codemod 58.0.1 sobre archivos de prueba · tsc --strict contra los tipos de expo-router 56.2.22
ACTUALIZADA	19 de noviembre de 2026
DOCUMENTACIÓN	docs.expo.dev/router/migrate/sdk-55-to-56
ARTÍCULO	ecomlabs.dev/g/router
AUTOR	Luis Morales · @luismoralesdev

Índice

Antes de empezar	Qué cambia
Paso 01	Inventario de imports
Paso 02	Subir el SDK
Paso 03	Correr el codemod
Paso 04	Lo que se arregla a mano
Paso 05	Librerías de terceros
Paso 06	Probar en iOS y Android
Después	SDK 57 y SDK 58
Al terminar	Checklist

ANTES DE EMPEZAR

Qué cambia

Desde el SDK 56, Expo Router no depende de React Navigation: Expo hizo un fork de las partes que usaba y las metió en el paquete. Según la guía oficial:

- Tu app ya no puede importar de `@react-navigation/*`; si lo hace, el bundler de Expo CLI 56 se detiene con un error.
- La API no cambia: solo se mueve la ruta de los imports.
- Las librerías de `node_modules` que importan `@react-navigation/core` se reescriben solas, de forma temporal.

Trabaja en una rama nueva con el árbol limpio: el codemod reescribe archivos y vas a revisar el diff completo.

PASO 01

Inventario de imports

El primer comando lista cada import de React Navigation en tu código; el segundo, qué dependencias traen React Navigation (con yarn o pnpm, `yarn why` o `pnpm why`):

```
grep -rnE "from ['\"]@react-navigation/" app src components
npm ls @react-navigation/core @react-navigation/native
```

Anota lo que el codemod no resuelve: imports de `native-stack` o `drawer`, imports con `import * as` o por defecto, y `createStackNavigator` o `TransitionPresets` de `@react-navigation/stack`.

PASO 02

Subir el SDK

Expo recomienda subir un SDK a la vez. Desde el 55:

```
npm install expo@^56.0.0
npx expo install --fix
npx expo-doctor
```

El primero instala el SDK 56; el segundo pone el resto de paquetes, incluido `expo-router` (ahora 56.x, la versión sigue al SDK), en las versiones del SDK; el tercero revisa problemas comunes y en el SDK 56 avisa si `react-navigation` sigue instalado junto a `expo-router`. Ese aviso lo resuelves en el paso 05.

PASO 03

Correr el codemod

Desde la raíz del proyecto. Pasa todas las carpetas con código, o un glob entre comillas. Con otros gestores, cambia `npx` por `yarn dlx`, `pnpm dlx` o `bunx`:

```
npx expo-codemod sdk-56-expo-router-react-navigation-replace app components
```

Al final imprime `ok`, `skipped` y `errors`. Lo que vimos con `expo-codemod` 58.0.1:

- **ok:** archivo reescrito. Los imports con el mismo destino se juntan en uno y conservan los `type`.
- **skipped:** el archivo tenía `import * as`, un import por defecto, o un import de `native-stack` o `drawer`. No se tocó ninguno de sus imports.

> **CUIDADO**
> El codemod escribe las rutas con comillas dobles, y en Windows dejó saltos de línea CRLF. Corre tu formateador y revisa `git diff`.

PASO 04

Lo que se arregla a mano

ANTES	DESPUÉS
`@react-navigation/native`, `core`, `elements`, `routers`	`expo-router/react-navigation`
`@react-navigation/stack`	`expo-router/js-stack`
`@react-navigation/bottom-tabs`	`expo-router/js-tabs`
`@react-navigation/material-top-tabs`	`expo-router/js-top-tabs`
`@react-navigation/native-stack`	Layout `Stack` de `expo-router`
`@react-navigation/drawer`	Layout `Drawer` de `expo-router/drawer`

Import de espacio de nombres. Cambia `import * as Nav from '@react-navigation/native'` por imports con nombre (`import { useNavigation } from ...`) y vuelve a correr el codemod sobre ese archivo.

native-stack. En Expo Router el stack es un layout por archivos: un `_layout.tsx` en la carpeta de las pantallas reemplaza a `createNativeStackNavigator`, y cada archivo de la carpeta es una pantalla. Para `drawer`, usa `Drawer` de `expo-router/drawer`.

```
// app/_layout.tsx
import { Stack } from 'expo-router';

export default function Layout() {
  return <Stack />;
}
```

> **CUIDADO**
> En el SDK 56, `expo-router/js-stack` solo exporta `Stack` (lo comprobamos en los tipos de 56.0.0 y 56.2.22). `createStackNavigator` y `TransitionPresets` no están y el import falla en TypeScript.

Repite el `grep` del paso 01. Si no imprime nada, tu código ya está migrado.

PASO 05

Librerías de terceros

Si una librería importa `@react-navigation/core`, Expo CLI 56 y 57 reescriben el import a `expo-router/react-navigation`. Si importa `@react-navigation/native` u otro paquete, no: busca una versión nueva, prueba la pantalla donde la usas y, si falla, abre un issue en la librería. Para aislar un problema:

```
EXPO_ROUTER_DISABLE_RN_NAVIGATION_CHECK=1 npx expo start
```

Esa variable apaga la reescritura y también el error para imports viejos en tu código. No la dejes en `.env` ni en CI. Al terminar, quita los paquetes `@react-navigation/*` de tu `package.json` y revisa otra vez:

```
npm uninstall @react-navigation/native @react-navigation/elements
npm ls @react-navigation/core @react-navigation/native
npx expo-doctor
```

Pon en `npm uninstall` los que tengas. Si `npm ls` sigue mostrando alguno, lo trae una librería.

PASO 06

Probar en iOS y Android

```
npx expo start --clear
npx expo run:ios
npx expo run:android
```

`--clear` arranca Metro con la caché vacía; `run:ios` compila en local y necesita una Mac con Xcode; `run:android` compila la app de Android. En los dos sistemas revisa el tema claro y oscuro, los encabezados y el botón de volver, las pestañas, los enlaces profundos, las pantallas con `useIsFocused` o `useFocusEffect`, y cada pantalla con una librería del paso 05.

DESPUÉS

SDK 57 y SDK 58

El SDK 57 (30 de junio de 2026) no trae cambios incompatibles en Expo Router. El SDK 58 (15 de octubre de 2026) rehízo el núcleo de navegación y quitó la mayor parte de la API copiada de React Navigation; tiene su guía en docs.expo.dev/router/migrate/sdk-57-to-58.

> **CUIDADO**
> Expo CLI 58.0.2 quitó las comprobaciones de `@react-navigation`: en el SDK 58 no hay error para imports viejos ni reescritura para librerías. Deja el `grep` del paso 01 en tu CI.

AL TERMINAR

Checklist

- [ ] Rama nueva e inventario de imports y librerías hecho.
- [ ] `expo` en `^56.0.0` y `npx expo install --fix` corrido.
- [ ] Codemod sobre todas las carpetas; archivos saltados arreglados a mano.
- [ ] `native-stack` y `drawer` pasaron a los layouts `Stack` y `Drawer`.
- [ ] El `grep` de imports viejos no imprime nada.
- [ ] `npx expo-doctor` sin avisos y librerías probadas.
- [ ] La app compila y funciona en iOS y en Android.
- [ ] `EXPO_ROUTER_DISABLE_RN_NAVIGATION_CHECK` no quedó puesta.