¡De una! Te dejo un flujo claro para dejar VS Code listo (React, Next.js y React Native) y luego crear y correr tu primer “Hello Word” en tu celular 👇

**1) Arma un workspace en VS Code con todo lo necesario**

**A. Requisitos básicos**

* Instala **VS Code** y **Node.js LTS**.
* (Opcional pero útil) Instala **Git** y un manejador de versiones de Node (nvm).

**B. Crea el workspace**

1. En VS Code: *File → Add Folder to Workspace…*
   * Agrega/crea dos carpetas (puedes cambiarlas luego):  
     web/ (Next.js) y mobile/ (React Native/Expo).
2. Guarda el workspace: *File → Save Workspace As…* y nómbralo, por ej: fullstack-mobile.code-workspace.

**C. Extensiones recomendadas (instálalas desde la pestaña Extensions o con CLI)**

# Ejecuta en terminal (si tienes el comando `code` disponible)

code --install-extension dbaeumer.vscode-eslint # ESLint

code --install-extension esbenp.prettier-vscode # Prettier

code --install-extension dsznajder.es7-react-js-snippets # Snippets React/Redux/RN

code --install-extension msjsdiag.vscode-react-native # React Native Tools

code --install-extension expo.vscode-expo-tools # Expo Tools

code --install-extension bradlc.vscode-tailwindcss # (opcional) Tailwind IntelliSense

* **React Native Tools**: depura y ejecuta comandos RN dentro de VS Code. ([Marketplace de Visual Studio](https://marketplace.visualstudio.com/items?itemName=msjsdiag.vscode-react-native&utm_source=chatgpt.com), [Microsoft en GitHub](https://microsoft.github.io/vscode-react-native/?utm_source=chatgpt.com))
* **Snippets ES7+**: atajos para componentes y hooks de React. ([Marketplace de Visual Studio](https://marketplace.visualstudio.com/items?itemName=dsznajder.es7-react-js-snippets&utm_source=chatgpt.com))
* **Tailwind IntelliSense** (si usarás Tailwind en Next.js): autocompletado y resaltado. ([Marketplace de Visual Studio](https://marketplace.visualstudio.com/items?itemName=bradlc.vscode-tailwindcss&utm_source=chatgpt.com), [tailwindcss.com](https://tailwindcss.com/docs/editor-setup?utm_source=chatgpt.com))
* Cómo administrar extensiones en VS Code: ([code.visualstudio.com](https://code.visualstudio.com/docs/configure/extensions/extension-marketplace?utm_source=chatgpt.com))

**D. Ajustes útiles del workspace**  
Crea/edita tu .code-workspace y pega algo así (ajústalo a tu gusto):

{

"folders": [

{ "path": "web" },

{ "path": "mobile" }

],

"settings": {

"editor.formatOnSave": true,

"eslint.validate": ["javascript","javascriptreact","typescript","typescriptreact"],

"[javascript]": { "editor.defaultFormatter": "esbenp.prettier-vscode" },

"[typescript]": { "editor.defaultFormatter": "esbenp.prettier-vscode" },

"[javascriptreact]": { "editor.defaultFormatter": "esbenp.prettier-vscode" },

"[typescriptreact]": { "editor.defaultFormatter": "esbenp.prettier-vscode" }

},

"extensions": {

"recommendations": [

"dbaeumer.vscode-eslint",

"esbenp.prettier-vscode",

"dsznajder.es7-react-js-snippets",

"msjsdiag.vscode-react-native",

"expo.vscode-expo-tools",

"bradlc.vscode-tailwindcss"

]

}

}

**E. Inicializa tus proyectos dentro del workspace (opcional ahora, útil luego)**

* **Web (Next.js)**:
* cd web
* npx create-next-app@latest
* # ESLint viene integrado con next; puedes ejecutar:
* npx next lint

Detalles de ESLint en Next: ([nextjs.org](https://nextjs.org/docs/app/api-reference/config/eslint?utm_source=chatgpt.com))

* **Móvil (Expo/React Native)**:
* cd ../mobile
* npx create-expo-app@latest

create-expo-app te genera un proyecto RN listo para Android/iOS/web. ([docs.expo.dev](https://docs.expo.dev/more/create-expo?utm_source=chatgpt.com" \o "create-expo-app))

**2) Tu primera app móvil “Hello Word” con Expo**

**Paso 1. Crea el proyecto**

# Dentro de la carpeta mobile o donde prefieras:

npx create-expo-app@latest hello-word

cd hello-word

Guía oficial de “crea tu primera app”: ([docs.expo.dev](https://docs.expo.dev/tutorial/create-your-first-app/?utm_source=chatgpt.com" \o "Create your first app - Expo Documentation))

**Paso 2. Ejecuta en modo desarrollo**

npx expo start

Esto abre el servidor de desarrollo y muestra un **QR**. ([docs.expo.dev](https://docs.expo.dev/get-started/start-developing/?utm_source=chatgpt.com" \o "Start developing))

**Paso 3. Abre en tu teléfono**

* Instala **Expo Go**:
  + **Android**: desde Google Play o la página de Expo Go. ([Expo](https://expo.dev/go?utm_source=chatgpt.com))
  + **iOS**: desde App Store (Expo Go). ([Apple](https://apps.apple.com/us/app/expo-go/id982107779?utm_source=chatgpt.com))
* Con tu móvil y tu compu en la **misma Wi-Fi**, abre Expo Go y **escanea el QR** que te muestra la terminal/Metro. (También puedes abrir por enlace). ([docs.expo.dev](https://docs.expo.dev/get-started/start-developing/?utm_source=chatgpt.com" \o "Start developing))
* Alternativamente, sigue las guías “Run on device” de RN (si usas CLI nativa). ([reactnative.dev](https://reactnative.dev/docs/running-on-device?utm_source=chatgpt.com" \o "Running On Device))

**Paso 4. Cambia el texto a “Hello Word”**  
Los proyectos nuevos de Expo traen **Expo Router** con app/. Abre app/index.tsx y deja algo mínimo:

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

export default function Index() {

return (

<View style={{flex:1, alignItems:"center", justifyContent:"center"}}>

<Text style={{ fontSize: 24 }}>Hello Word 👋</Text>

</View>

);

}

(En templates antiguos, el archivo podría ser App.js. El tutorial explica ambas rutas.) ([docs.expo.dev](https://docs.expo.dev/tutorial/create-your-first-app/?utm_source=chatgpt.com" \o "Create your first app - Expo Documentation))

**3) ¿Cómo “instalarla” en tu celular (sin depender de Expo Go)?**

Para **probar rápido**, Expo Go basta. Si quieres **instalar** un ícono de app:

**A) Android – genera un APK para instalar**

1. Instala EAS CLI y autentícate:
2. npm install -g eas-cli
3. eas login

([docs.expo.dev](https://docs.expo.dev/deploy/submit-to-app-stores/?utm_source=chatgpt.com" \o "Submit to app stores))

1. Crea eas.json con un perfil que genere **APK**:
2. {
3. "build": {
4. "preview": {
5. "android": { "buildType": "apk" }
6. }
7. }
8. }

([docs.expo.dev](https://docs.expo.dev/build-reference/apk/?utm_source=chatgpt.com" \o "Build APKs for Android Emulators and devices))

1. Construye y obtén el enlace de descarga:
2. eas build -p android --profile preview

Te dará un **link** para instalar o descargar el .apk. Para compartir con testers, mira **Internal distribution**. ([docs.expo.dev](https://docs.expo.dev/build/setup/?utm_source=chatgpt.com" \o "Create your first build))

**B) iPhone – usa TestFlight**

* Necesitas **cuenta de Apple Developer** y hacer **EAS Build** para iOS:
* npm install -g eas-cli && eas login
* eas build --platform ios --profile production
* # luego sube a TestFlight (EAS Submit) o usa `npx testflight`:

Pasos oficiales (bundle id, credenciales, envío a App Store/TestFlight).

Tip: si tu app usa librerías nativas no soportadas por Expo Go, crea un **development build** (tu propio “Expo Go”) y pruébala en tu dispositivo/emulador.

**Mini checklist rápido**

* **VS Code + extensiones** (RN Tools, ES7 snippets, ESLint/Prettier, Expo Tools, Tailwind si usas Next).
* **Crear app con Expo**: npx create-expo-app@latest → npx expo start.
* **Abrir en el celular** con **Expo Go** escaneando el QR.
* **Instalar en Android (APK)** o **publicar en iOS (TestFlight/App Store)** con **EAS Build**.

Si quieres, te genero el archivo .code-workspace listo para copiar/pegar y los comandos exactos para Windows/macOS según tu equipo. ¿En qué SO estás trabajando?

<https://www.google.com/search?sca_esv=4e30fb7402af9a52&rlz=1C1CHBF_enCR1134CR1134&sxsrf=AE3TifORLBgSisnxu-5M24b3gV_ZcJFYwg:1754866731642&udm=7&fbs=AIIjpHx4nJjfGojPVHhEACUHPiMQ33KVMWSawhpRB5XP_HUyXOzcS6Ni12gUJ1mwL4TV-hjAAWlEcsrYf79aptwTsEDuHz0evL0YdIdFlwnUPjw0d5iqpe5qlwEJgG6f0ezCOxcGHUQC1HrTZgMZPOT6BMyhUWoNUbBxqNYSnTKkzNCI8YOWFZqkCBVVOrHxnCna58xUDu6lBhqVBkQxj41FKfnxqMA7Dw&q=mi+primer+app+android+con+expo+y+react+native+espa%C3%B1ol&sa=X&ved=2ahUKEwi5wKuBrIGPAxVmmIQIHSnsAJYQtKgLegQIFBAB&biw=1745&bih=866&dpr=1.1#fpstate=ive&vld=cid:c374fc47,vid:U23lNFm_J70,st:0>