GUÍA NODESTUDIO

Tu app en la App Store con Claude Code + Expo

Creala con Claude Code, probala en tu celular y publicala en la App Store.

⏱️ Unas horas de trabajo + los tiempos de Apple

1 Antes de empezar

Qué necesitás	Para qué
Claude Code instalado	Escribe el código y corre los comandos por vos.
Node.js LTS (`nodejs.org`)	Lo usa Expo para crear y levantar el proyecto.
Cuenta gratis de Expo (`expo.dev/signup`)	Para usar Expo Go y compilar con EAS.
App Expo Go en el celular	Para probar la app escaneando un QR. Está en la App Store y en Google Play.
Apple Developer Program	Solo para publicar. Cuesta 99 USD por año (paso 3).

Documentación oficial: `docs.expo.dev`

> **NO NECESITÁS UNA MAC**
> EAS compila la app de iOS en la nube y la sube a App Store Connect desde Windows, Linux o macOS.

Configurá Expo

PROMPT PARA PEGAR EN CLAUDE CODE

```
Verificá que tenga Node.js LTS instalado con node -v. Después instalá EAS CLI con npm install --global eas-cli y confirmá con eas --version que quedó instalado. Para los comandos de Expo usá siempre npx expo (no instales el paquete viejo expo-cli).
```

LO QUE CORRE CLAUDE CODE

```
node -v
npm install --global eas-cli
eas --version
```

El inicio de sesión hacelo vos, en tu terminal: te pide usuario y contraseña de Expo, y esos datos no se los pases a Claude Code.

EN TU TERMINAL

```
npx expo login
```

EAS usa esa misma sesión (lo confirmás con `eas whoami`). Después abrí Expo Go en el celular e iniciá sesión **con la misma cuenta** (ícono de perfil, arriba a la derecha). En iPhone es obligatorio: sin sesión en los dos lados, Expo Go no abre el proyecto.

2 Creá la app y probala en tu celular

1. **Abrí Claude Code** en la carpeta donde guardás tus proyectos: entrá a esa carpeta en la terminal y escribí `claude`.
2. **Pedile la app.** Pegale el prompt de la página 7. Claude Code crea el proyecto con `npx create-expo-app@latest` y escribe las pantallas. El proyecto que genera Expo ya trae un `CLAUDE.md` con instrucciones de Expo para Claude Code.
3. **Levantá el servidor de desarrollo** en otra terminal, dentro de la carpeta del proyecto (usá el nombre de carpeta que haya elegido Claude Code). Queda corriendo y muestra un QR:

   EN TU TERMINAL

   ```
   cd habitos
   npx expo start
   ```

4. **Escaneá el QR.** En iPhone, con la app Cámara: el link se abre en Expo Go. En Android, abrí Expo Go y tocá *Scan QR code*.
5. **Si no conecta:** el celular y la compu tienen que estar en la misma red Wi-Fi. Si igual no carga, cortá el servidor con `Ctrl+C` y levantalo con túnel:

   ```
   npx expo start --tunnel
   ```

6. **Iterá.** Pedile cambios a Claude Code: cuando guarda los archivos, la app se actualiza sola en el celular. Si no se actualiza, apretá `r` en la terminal del servidor para recargar.

> **¿EXPO GO DICE QUE EL SDK NO ES COMPATIBLE?**
> La versión de Expo Go de las tiendas corre una sola versión del SDK de Expo (la ves en `expo.dev/go`; hoy es la 57). Si el proyecto usa otra, pedile a Claude Code que lo cree o lo pase a esa versión.

3 Configurá Apple Developer y App Store Connect

Para publicar en la App Store necesitás una cuenta paga de Apple. Estos pasos los hacés vos en el navegador: son tu cuenta y tus datos legales.

1. **Activá la verificación en dos pasos** en tu Apple Account. Es requisito para inscribirte.
2. **Inscribite en el Apple Developer Program** en `developer.apple.com/programs/enroll`. Cuesta 99 USD por año (el precio puede variar según la región). Podés anotarte como persona o como organización; como organización te piden además un número D-U-N-S. Esperá a que Apple active tu membresía antes de seguir.
3. **Aceptá los acuerdos.** Entrá a `appstoreconnect.apple.com` → *Business* y aceptá el acuerdo pendiente. Hasta que no lo firmes, Apple no te deja agregar apps.
4. **Elegí el bundle identifier:** el ID único de tu app, con formato de dominio invertido, por ejemplo `com.tunombre.habitos`. Pedile a Claude Code que lo configure:

   PROMPT PARA PEGAR EN CLAUDE CODE

   ```
   En app.json agregá ios.bundleIdentifier con el valor com.tunombre.habitos y
   dejá la versión en 1.0.0. Mostrame cómo quedó el archivo.
   ```

5. **La ficha de la app se crea sola.** No hace falta crearla a mano en App Store Connect: la primera vez que corras `eas submit` (paso 4), si la app todavía no existe, EAS la crea y te pide el nombre. Ese nombre tiene que estar libre en la App Store.

> **TENÉ A MANO PARA LOS PASOS 4 Y 5**
>
> Tu Apple ID y contraseña, el dispositivo donde recibís el código de verificación de Apple, la URL de tu política de privacidad y una URL de soporte.

4 Generá el build con EAS y subilo

EAS compila la app en la nube y la sube a App Store Connect. Estos comandos corrélos vos en tu terminal, dentro de la carpeta del proyecto: son interactivos y te piden tu Apple ID y el código de verificación.

1. **Configurá EAS para iOS.** Crea el archivo `eas.json` y vincula el proyecto con tu cuenta de Expo. Si te pregunta si querés crear el proyecto en EAS, aceptá.

```
eas build:configure --platform ios
```

2. **Generá el build de producción.** Te pide iniciar sesión en tu cuenta de Apple; dejá que EAS genere el certificado de distribución y el provisioning profile. Si te pregunta si la app usa solo cifrado estándar/exento y tu app no usa cifrado propio, respondé que sí. El avance lo seguís en `expo.dev` → Builds. En el plan gratis, el build puede esperar un rato en cola.

```
eas build --platform ios --profile production
```

3. **Subilo a App Store Connect.** `--latest` elige el último build. Te pide tu Apple ID y crea la app en App Store Connect si todavía no existe.

```
eas submit --platform ios --latest
```

4. **Esperá a que Apple lo procese.** El build aparece en la pestaña *TestFlight* de tu app en App Store Connect. Según Expo, el procesamiento suele tardar de 10 a 15 minutos.

Para las próximas versiones podés hacer el build y la subida en un solo comando:

```
eas build --platform ios --auto-submit
```

> **SUBIR NO ES MANDAR A REVISIÓN**
> `eas submit` y `--auto-submit` dejan el build en App Store Connect / TestFlight, pero **no** mandan la app a revisión de Apple. Ese paso es manual y lo hacés vos (página siguiente).

5 Mandala a revisión desde App Store Connect

Esto no lo hacen Expo ni Claude Code: lo hacés vos en `appstoreconnect.apple.com` → *Apps* → tu app → la versión 1.0 (estado *Prepare for Submission*). Los nombres de las secciones pueden aparecerte en español.

1. **Capturas de pantalla:** entre 1 y 10 de iPhone, en PNG o JPG sin transparencias. App Store Connect te indica los tamaños aceptados. Si la app corre en iPad (`supportsTablet: true` en `app.json`), también te pide capturas de iPad.
2. **Textos de la ficha:** descripción, palabras clave, URL de soporte y copyright.
3. **Build:** en la sección *Build*, elegí el que subiste con EAS.
4. **App Information:** categoría, clasificación por edad (es un cuestionario) y derechos de contenido.
5. **App Privacy:** la URL de tu política de privacidad y el cuestionario sobre qué datos recolecta la app.
6. **Pricing and Availability:** precio (o gratis) y países donde se publica.
7. **App Review Information:** tus datos de contacto. Si la app tiene login, dejá un usuario de prueba para que Apple pueda entrar.
8. **Enviala:** arriba a la derecha tocá *Add for Review* y después *Submit for Review* en el panel del envío. Cuando Apple empieza a revisarla, el estado pasa a *In Review*.

> **SI APPLE LA RECHAZA**
>
> El motivo te queda en App Store Connect. Corregí lo que te piden (si hace falta, generá y subí un build nuevo con el paso 4) y volvé a enviarla.

Prompts listos para copiar

Pegalos tal cual en Claude Code, dentro de la carpeta donde querés trabajar.

PARA CREAR LA APP · PASO 2

```
Creame una app para trackear hábitos con Expo que funcione en iPhone y Android.
```

Cuando termine, levantá el servidor con `npx expo start` y escaneá el QR.

PARA PUBLICARLA · PASOS 3 Y 4

```
Prepará esta app para publicarla en la App Store. Configurá Expo/EAS, verificá todo lo necesario y guiame paso a paso hasta subirla a App Store Connect.
```

Pegalo con la carpeta del proyecto abierta en Claude Code.

DOS COSAS QUE SIGUEN SIENDO TUYAS

- Los comandos que piden credenciales (`npx expo login` , `eas build` , `eas submit`) corrélos en tu terminal. No le pases tus contraseñas a Claude Code.
- El envío a revisión: después de subir el build, entrás a App Store Connect y tocás *Add for Review* → *Submit for Review* (paso 5).



NodeStudio

Automatizamos procesos de negocio con IA y sistemas a medida.

nodestudioai.com