

Distribución de builds iOS con EAS — VeriX

Dos formas de instalar un build de prueba en un iPhone físico sin pasar por el simulador.

“ Para generar el build en sí (`.ipa` / `.tar.gz`) con versión y nombre automáticos, ver `BUILD.md` (`scripts\build.ps1`). Esta guía asume que ya tienes ese archivo listo.

Opción A: Ad Hoc (directo al dispositivo, sin TestFlight)

Rápido, pero limitado a dispositivos registrados manualmente (máx. 100 UDIDs/año en tu cuenta Apple Developer).

1. Ajustar el perfil en `eas.json`

```
"test": {
  "distribution": "internal",
  "ios": {
    "simulator": false
  }
}
```

2. Registrar el iPhone (obtener su UDID)

```
npx eas-cli device:create
```

- Genera un link/QR.
- Ábrelo **desde Safari en el iPhone**.
- Instala el perfil de configuración que se descarga — esto registra el UDID automáticamente en Expo/Apple.

3. Correr el build

```
.\scripts\build.ps1 -Env test -Platform ios
```

- EAS genera (o reutiliza) el certificado de distribución y un provisioning profile ad hoc que incluye ese dispositivo.
- El resultado es un `.ipa` (no `.tar.gz`).

4. Instalar en el iPhone

- El build aparece en expo.dev o en el link que EAS imprime en consola al terminar.
- Abre ese link **desde el navegador del iPhone**.
- Toca instalar — no requiere Mac, App Store ni TestFlight.

Notas

- Solo dispositivos registrados *antes* de generar el provisioning profile pueden instalar el build.
- Si el dispositivo se agregó después de un build viejo, puede que necesites regenerar el profile:

```
eas build --clear-provisioning-profile
```

- El provisioning profile expira (normalmente ~1 año).

Opción B: TestFlight

Mejor para testers externos o cuando no quieres registrar UDIDs uno por uno. Pasa por App Store Connect, así que toma un poco más de tiempo (revisión automática, no manual, suele ser minutos-horas).

1. Requisitos previos

- Cuenta de Apple Developer Program (paga, \$99/año) vinculada al proyecto en EAS.
- App creada en [App Store Connect](#).

2. Ajustar el perfil de build en `eas.json`

Para TestFlight normalmente se usa el perfil de `production` (o uno dedicado), con distribución **store**:

```
"production": {
  "distribution": "store",
  "ios": {
    "simulator": false
  }
}
```

3. Generar el build

```
.\scripts\build.ps1 -Env prod -Platform ios
```

(o el ambiente que corresponda al perfil que uses para TestFlight)

4. Subir el build a App Store Connect

```
npx eas-cli submit -p ios --latest
```

- Sube el `.ipa` más reciente automáticamente a App Store Connect.
- Espera a que Apple procese el build (aparece en la sección TestFlight de App Store Connect, usualmente en minutos).

5. Agregar testers

- En App Store Connect → tu app → pestaña **TestFlight**.
- Agrega testers por email (grupo interno) o crea un **link público** de invitación si quieres que cualquiera con el link pueda unirse.
- Los testers instalan la app **TestFlight** desde el App Store, y desde ahí instalan tu build usando el link/invitación.

Notas

- No requiere registrar UDIDs — cualquier tester invitado puede instalar sin configuración previa.
 - Los builds de TestFlight expiran a los 90 días.
 - Testers externos (fuera de tu equipo de Apple Developer) requieren que Apple apruebe el build para "external testing" (revisión ligera, normalmente rápida).
-

Opción C: Producción (publicación en App Store)

El build es técnicamente el mismo que para TestFlight (`distribution: "store"`), pero además de subirlo hay que enviarlo a **revisión de Apple** y publicarlo.

1. Build y submit (igual que TestFlight, pasos 3-4)

```
.\scripts\build.ps1 -Env prod -Platform ios  
npx eas-cli submit -p ios --latest
```

Esto sube el `.ipa` a App Store Connect — mismo build que usarías para TestFlight, solo que ahora lo vas a enviar a revisión en vez de (o además de) dejarlo en testing interno.

2. Completar la ficha en App Store Connect

Antes de enviar a revisión, en App Store Connect → tu app → **App Store** tab necesitas tener listo:

- Screenshots (por tamaño de dispositivo)
- Descripción, keywords, categoría
- Política de privacidad (URL)
- Clasificación de contenido (age rating)
- Precio/disponibilidad por país

3. Enviar a revisión

- Selecciona el build que subiste (paso 1) en la sección **Build**.
- Click en **Add for Review** → **Submit for Review**.
- Apple revisa (normalmente 24-48h, puede variar).

4. Publicación

- Si aprueban, puedes elegir publicación **manual** (tú decides cuándo sale al público) o **automática** (sale apenas Apple aprueba) — se configura antes de enviar a revisión.
- También puedes usar **phased release** (lanzamiento gradual a % de usuarios en 7 días) para reducir riesgo si algo sale mal.

Notas

- El número de versión (`app.json` → `version`) y el `buildNumber` deben ser mayores a la última versión publicada — tu script ya incrementa el patch automáticamente, pero revisa que no choque con una versión ya en revisión.
- Una vez enviado a revisión, no puedes subir un build nuevo sin cancelar el envío anterior.
- Si tu app usa capacidades especiales (push notifications, in-app purchases, etc.), Apple puede pedir información adicional o rechazar por incumplimiento de guidelines — vale la pena revisar el build en TestFlight primero antes de enviarlo a producción.

¿Cuál usar?

	Ad Hoc	TestFlight	Producción
Setup inicial	Rápido	Más pasos (App Store Connect)	Igual que TestFlight + ficha completa
Requiere registrar UDID	Sí	No	No
Límite de dispositivos	100/año	Sin límite práctico	Sin límite (público)
Expiración del build	~1 año	90 días	No expira (versión publicada)
Pasa por revisión de Apple	No	Ligera (external testing)	Sí (revisión completa, 24-48h)
Ideal para	Prueba rápida en 1-2 dispositivos del equipo	Testers externos, distribución más amplia	Lanzamiento público

Revision #4
Created 2026-08-04 12:28:08 UTC by Admin
Updated 2026-08-04 12:41:47 UTC by Admin