

Submit a TestFlight — VeriX (iOS)

Guía completa, ya probada de punta a punta, para subir un build de VeriX a TestFlight y que un tester lo instale en su iPhone. Incluye el método de autenticación que **sí funciona** en esta cuenta de Apple Developer (Nexus 33 Group LLC) y por qué el método "estándar" (API Key) no sirve aquí.

“ Requiere tener ya un build de iOS generado. Ver `BUILD.md` para eso (`scripts\build.ps1 -Env test -Platform ios`).

?? Importante: usar Apple ID + App-Specific Password, NO API Key

La forma "recomendada" por Expo para `eas submit` es una **API Key de App Store Connect** (`.p8` + Key ID + Issuer ID). **En esta cuenta esa forma no funciona** — probamos 5 variantes distintas (key manual, `.bat` con variables de entorno, generación automática por EAS, servicio de credenciales cifrado `eas credentials`, y una segunda key nueva generada desde cero) y las 5 fallaron con el mismo error:

```
Error: Apple 403 detected - Access forbidden.
This request is forbidden for security reasons - The API key in use does not allow this request
× Failed to fetch App Store Connect API Key.
```

Se descartaron como causa: rol de la key (era Admin), tipo de key (Team Keys, que es el correcto), Términos y Condiciones (aceptados), membresía de Apple Developer (activa y pagada), y corrupción del archivo `.p8` (se confirmó que nunca se abrió/editó). El patrón (2 keys distintas, mismo error exacto) apunta a una restricción a nivel de cuenta que Apple no expone con un mensaje claro — posiblemente ligada a que la membresía de la organización se aprobó recientemente.

La solución que sí funcionó: autenticar con Apple ID + una **App-Specific Password**, en vez de una API Key. Es un mecanismo distinto dentro de `eas-cli` que no pasa por la llamada que estaba fallando.

Paso 1 — Generar una App-Specific Password

Esto se hace en la cuenta personal de Apple, **no en App Store Connect** (son sitios distintos).

1. Ve a **https://appleid.apple.com** (no `appstoreconnect.apple.com`).
2. Inicia sesión con el Apple ID que tiene acceso al equipo Nexus 33 Group LLC (ej. `hectoregarcia@hotmail.com`).
3. Click en **Sign-In and Security**.
4. Busca la tarjeta **App-Specific Passwords** → click en ella.
5. Click en + para generar una nueva.
6. Ponle un nombre descriptivo, ej. `EAS Submit`.
7. Apple muestra la contraseña en formato `abcd-efgh-ijkl-mnop`. **Cópiala en ese momento** — Apple no la vuelve a mostrar después.

“ Cada dev del equipo que vaya a hacer `eas submit` necesita generar la suya propia con su propio Apple ID (si tiene acceso al equipo).

Paso 2 — Configurar `eas.json`

Agrega el `appleId` (esto no es secreto, se puede dejar en el repo) dentro del perfil de submit que uses:

```
"submit": {
  "test": {
    "ios": {
      "appleId": "hectoregarcia@hotmail.com"
    }
  }
}
```

Paso 3 — Setear la contraseña como variable de entorno (por sesión, no se guarda en ningún archivo)

En PowerShell, antes de correr el submit:

```
$env:EXPO_APPLE_APP_SPECIFIC_PASSWORD = "abcd-efgh-ijkl-mnop"
```

No hay confirmación visual — es normal. Esta variable solo vive en esa ventana de terminal; se pierde al cerrarla (por diseño, para no dejarla en disco).

Para no repetir esto cada sesión (opcional, persiste solo en tu máquina, nunca en el repo):

```
[System.Environment]::SetEnvironmentVariable("EXPO_APPLE_APP_SPECIFIC_PASSWORD", "abcd-efgh-ijkl-mnop", "User")
```

(abre una terminal nueva para que tome efecto)

Paso 4 — Correr el submit

En la misma terminal donde seteaste la variable:

```
npx eas-cli submit -p ios --profile test
```

Selecciona el build que quieres subir cuando te pregunte. Con la contraseña específica ya en el entorno, no debería pedirte ninguna API Key — autentica directo con el Apple ID.

Al terminar deberías ver:

```
✓ Submitted your app to Apple App Store Connect!  
Your binary has been successfully uploaded to App Store Connect!
```

Apple tarda **5-10 minutos** en procesar el build (a veces más). Llega un correo cuando termina.

Paso 5 — Agregar testers en App Store Connect

1. Ve a **<https://appstoreconnect.apple.com>** → selecciona la app **VerixApp**.
2. Pestaña **TestFlight** → sección **iOS Builds**.
3. Una vez el build aparece con estado "**Ready to Submit**" o "**Ready to Test**", entra al build (click en el número/ícono).
4. Si pide **Export Compliance** (uso de encriptación), respóndelo según corresponda a la app (la mayoría de apps que solo usan HTTPS estándar responden que no usan encriptación propia / usan solo la exenta).
5. Vuelve a **TestFlight** → **Internal Testing** → el grupo ya existente (ej. "**Team (Expo)**", creado automáticamente por Expo).
6. Click en el grupo → pestaña **Testers** → + para agregar un email.

“ **Requisito:** para poder agregar a alguien como tester interno, esa persona debe existir primero como usuario del equipo en **Users and Access** → **People**. Si su email no aparece, hay que invitarla ahí primero.

Paso 6 — El tester instala VeriX

Desde el **iPhone** de la persona invitada:

1. Revisa el correo — llega una invitación de Apple/TestFlight con el nombre de la app y un código de 8 caracteres (ej. `BHHDZRWF`).
2. Instala la app **TestFlight** desde el App Store, si no la tiene.
3. Abre TestFlight → toca **Redeem**.
4. Escribe el código del correo.
5. VeriX aparece en la lista → toca **Install**.

“ El código de canje es único por invitación/persona — cada tester tiene el suyo, no se puede compartir entre dos personas.

Resumen rápido (una vez ya configurado todo)

```
# 1. Generar build (ver BUILD.md)
.\scripts\build.ps1 -Env test -Platform ios

# 2. Setear la contraseña (si no está persistida ya)
$env:EXPO_APPLE_APP_SPECIFIC_PASSWORD = "abcd-efgh-ijkl-mnop"

# 3. Submit
npx eas-cli submit -p ios --profile test

# 4. Esperar correo de Apple (~5-10 min)

# 5. Agregar tester en App Store Connect (si es alguien nuevo)

# 6. El tester canjea el código en TestFlight desde su iPhone
```

Ver también

- `BUILD.md` — cómo generar el build (`.ipa`) con versión y nombre automáticos.
- `distribucion-ios-eas.md` — alternativa de distribución ad hoc (sin pasar por TestFlight, instalación directa en dispositivos registrados).
- `distribucion-android-eas.md` — equivalente para Android.

Revision #1

Created 2026-08-05 02:10:39 UTC by Admin

Updated 2026-08-05 02:11:05 UTC by Admin