

# Flujo de desarrollo GitHub - Asana

**Vigente desde:** 29 de julio de 2026 **Aplica a:** los 12 repositorios activos bajo DevelopmentNexus33  
**Motivo:** cumplimiento SOC2 — trazabilidad de cambios a tickets autorizados + segregación de funciones (nadie aprueba su propio código)

## 1. Repositorios cubiertos

| Repo                          | Rama principal | Rama de desarrollo |
|-------------------------------|----------------|--------------------|
| nexus33-microservice-nxs      | master         | development        |
| nexus33-config-file           | master         | development        |
| nexus33-frontend              | master         | development        |
| nexus33-app-verix             | main           | development        |
| nexus33-web-site-page         | main           | development        |
| nexus33-microservice-security | master         | development        |
| nexus33-api-gateway           | master         | development        |
| nexus33-authentication-server | master         | development        |
| nexus33-common                | master         | development        |
| nexus33-config-server         | master         | development        |
| nexus33-eureka-server         | master         | development        |
| compax-frontend               | main           | development        |

No incluidos en este flujo (quedan intactos, fuera de alcance por ahora): nexus33-microservice-nexuspoint, nexus33-zipkin-server, nexus33-config-file-cityofhighpoint, nexus33-config-file-safesky.

## 2. Qué cambió

A partir de hoy, en **todos** los repos de la tabla:

1. **No hay push directo a `master/main` ni a `development`**. Todo cambio entra por Pull Request, sin excepción — ni siquiera el dueño de la cuenta puede saltárselo (`enforce_admins` activo).
2. **Se necesita mínimo 1 aprobación** de otra persona para poder mergear. Nadie puede aprobar su propio PR.
3. **La aprobación tiene que venir de un Code Owner** (archivo `.github/CODEOWNERS` en cada repo — hoy incluye a todos los colaboradores del repo, revisar/ajustar si en el futuro se quiere dividir por área).
4. **Todo PR debe incluir el ticket de Asana en el título**, formato `NEXDEV-123` (número real del campo personalizado de la tarea). Un chequeo automático bloquea el botón de Merge si falta.
5. **No se permite force-push ni borrar `master/main/development`**.
6. Si se suben commits nuevos después de una aprobación, esa aprobación se descarta automáticamente y hay que volver a aprobar.

---

## 3. Los 2 campos que importan en un PR

| Campo                                        | ¿Qué es?                                                                                                                                                                                                                                                                    | ¿Quién lo pone?                                                       | ¿Bloquea el merge?                                        |
|----------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------|-----------------------------------------------------------|
| <b>Asignado</b> ( <i>Assignee</i> )          | Quién está trabajando en el cambio. Puramente organizativo, no tiene efecto técnico.                                                                                                                                                                                        | Cualquiera, manual (normalmente el autor se auto-assigna)             | No                                                        |
| <b>Revisor de código</b> ( <i>Reviewer</i> ) | Quién revisa <b>y aprueba</b> el PR. GitHub lo asigna automáticamente según el archivo <code>.github/CODEOWNERS</code> de cada repo (hoy: cualquier colaborador del repo). Cuando ese revisor le da clic a "Approve", esa aprobación es la que se exige para poder mergear. | Automático (CODEOWNERS), o manual si hace falta agregar a alguien más | <b>Sí</b> — se necesita mínimo 1 aprobación de un revisor |

No hay más roles que estos 2 dentro de un PR. (Aparte existe el nivel de acceso de cada persona al repo — Read/Write/Admin, configurado una sola vez en `Settings → Collaborators` — pero eso no cambia de un PR a otro.)

---

## 4. Cómo referenciar el ticket de Asana

Va en el **título del PR** (la caja de una sola línea arriba de la descripción, no en el cuerpo), en cualquier parte del texto:

Recuperar contraseña NEXDEV-123

NEXDEV-123: recuperar contraseña

Fix login bug (nexdev-123)

El chequeo no distingue mayúsculas/minúsculas. Si falta, el PR se puede crear igual, pero el botón de **Merge queda bloqueado** — se soluciona editando el título del PR ya creado (el check se vuelve a correr solo, no hace falta abrir uno nuevo).

**Adicional (recomendado, no forzado por el sistema):** pegar también el link de la tarea de Asana en la descripción del PR, para que la integración Asana↔GitHub la muestre en ambos lados.

## 5. Flujo completo — ejemplo real

Caso: ticket `NEXDEV-123`, "Recuperar contraseña", toca `nexus33-authentication-server` y `nexus33-frontend`.

1. **Ticket en Asana** — ya existe con su número `NEXDEV-123`.
2. **Crear rama** desde `development`:

```
git checkout development && git pull
git checkout -b feature/recuperar-password
```

3. **Desarrollar y hacer push** de la rama (esto sí está permitido, no es la rama protegida):

```
git push origin feature/recuperar-password
```

4. **Abrir PR** hacia `development`, título: `Recuperar contraseña NEXDEV-123`. Pegar el link de la tarea de Asana en la descripción.
5. **Revisor aprueba** el PR (Approve). Con eso se cumple el mínimo de 1 aprobación + code owner.
6. **Check de ticket pasa** automáticamente (ya tiene `NEXDEV-123` en el título).
7. **Merge a `development`**. Se prueba en el ambiente correspondiente (`nexus33-config-file` tiene configs `test`/`prod` separadas por cliente: `bistatedev`, `cityofhighpoint`, `sempa`, `verix`, `demo`).
8. **Cuando está validado:** PR de release, `development` → `master`, mismas reglas.
9. **Merge a `master`**.
10. **Build/deploy a producción** — disparado por un webhook externo a GitHub (*pendiente documentar el detalle exacto de este último tramo*).

---

## 6. Integración Asana ? GitHub

- Conectada desde Asana: **perfil** → **Ajustes** → **Aplicaciones** → **Ver todas las aplicaciones** → **GitHub** → **Conectar**, autorizado contra la cuenta `DevelopmentNexus33`.
  - Se conecta proyecto por proyecto en Asana (no hay gestión centralizada de apps en el plan Starter — eso es exclusivo de Enterprise+).
  - Requiere ser Admin del workspace de Asana + Admin/dueño de la organización en GitHub.
  - Con la tarea vinculada (link pegado en la descripción del PR), Asana refleja el estado del PR como comentario/adjunto en la tarea.
- 

## 7. Preguntas frecuentes

**¿Qué pasa si necesito hacer un cambio urgente y no tengo ticket?** Crea el ticket en Asana primero (toma segundos) y referencia su número. No hay excepción técnica para saltarse esto — es intencional.

**¿El chequeo revisa cada commit?** No. Revisa únicamente el **título del PR**, una sola vez por PR (se re-evalúa si editas el título). No es necesario poner el ticket en cada commit individual.

**¿Puedo aprobar mi propio PR?** No, GitHub no lo permite estructuralmente.

**¿Qué pasa si edito el título después de que el check falló?** Se vuelve a correr automáticamente. No hace falta cerrar ni recrear el PR.

---

## 8. Pendiente / roadmap

- ☐ Documentar qué dispara exactamente el build/deploy a producción tras el merge a `master`.
  - ☐ `generate_changelog.js` actualizado para traer evidencia real de PR (aprobador, fecha, ticket) vía API de GitHub, como reporte de evidencia SOC2.
  - ☐ Evaluar dividir `CODEOWNERS` por carpeta/servicio en vez de "todos aprueban todo".
  - ☐ Secret scanning y 2FA obligatorio — no disponibles en el plan actual (GitHub Pro personal); requeriría migrar a organización con GitHub Advanced Security o Enterprise.
- 

Revision #5

Created 2026-07-29 06:48:12 UTC by Admin

Updated 2026-07-29 21:54:15 UTC by Admin