**Cómo funcionan los roles en tu app**

**1) Alta de usuario (primera vez que inicia sesión)**

* Cuando alguien se autentica por primera vez, ProfileScreen.\_loadUser() revisa si existe su doc en /users/{uid}.
* Si **no existe**, lo **crea automáticamente** con:
  + role: 'operador' ✅ (este es el **rol por defecto**)
  + active: true
  + displayName, email, photoURL, timestamps…

👉 Resultado: **todo nuevo usuario entra como OPERADOR** hasta que un admin lo cambie.

**2) Quién puede cambiar el rol**

* En la UI de ProfileScreen, el **dropdown de Rol** solo es editable si el usuario **ya es administrador**:
* final canEditRole = \_roleIsAdminInMemory; // true solo si es administrador
* onChanged: canEditRole ? (...) : null
* Es decir, **cada quien solo ve su propio perfil** y **solo un admin puede cambiar su propio rol** (o sea, bajar o quedarse igual).
* ¿Y entonces quién asciende a otros? Con tus **reglas de Firestore** (las que hicimos), un **administrador** puede modificar **cualquier** documento de /users (no solo el suyo). La pantalla actual edita **solo el propio**; para editar a **otros** tienes dos opciones:

**Opción A (rápida hoy)**

Cambiar el campo role y active del nuevo usuario **desde la consola de Firestore** (un clic).  
Valores válidos:

* 'operador'
* 'supervisor'
* 'diseñador' (con acento, como en tu enum)
* 'administrador'

**Opción B (mejor UX)**

Crear una pantalla **“Administrar usuarios”** (solo visible para administradores) que:

* Liste /users (nombre, email, rol, activo)
* Permita cambiar role y active con un DropdownButton y Switch
* Guarde directo al doc de cada usuario (tus reglas ya lo permiten para admin)

Si te late, la armamos después (es corta).

**¿Por qué ahora no ves operadores en el alta?**

El AddProductionEntryScreen consulta:

collection('users')

.where('role', isEqualTo: 'operador')

.where('active', isEqualTo: true)

.orderBy('displayName')

Así que hasta que tu compañero se registre (se crea su doc) y **tenga role: 'operador' + active: true**, empezará a aparecer en ese dropdown.  
Mientras pruebas, puedes:

* Cambiar el rol a operador en consola, o
* Usar el “fallback” opcional que te dejé (agrega al usuario actual si no hay operadores) — útil solo en desarrollo.

**Detalles finos del profile\_screen.dart que acabas de pegar**

* **Enum + mapeo**  
  Tu enum usa 'diseñador' con acento y 'administrador'. Eso está bien y es **consistente** con las reglas que te pasé (usábamos 'administrador' / 'supervisor' / 'operador'). Mantén **exactamente** esos strings en toda la app y en Firestore.
* **Creación automática** en /users/{uid}: rol operador por defecto (ya explicado).
* **Guardado**: si no eres admin, role no se sobreescribe:
* 'role': canEditRole ? \_role.name : FieldValue.delete()

Esto evita que alguien “no admin” se cambie solo el rol desde la UI.

* **Reglas de seguridad**: ya dejamos que **admin** pueda editar cualquier /users/{uid}. Por eso la **pantalla de administración** (Opción B) es 100% factible sin tocar reglas.

**Qué tiene que hacer tu compañero “operador” para aparecer en la lista**

1. Abrir la app e iniciar sesión (Google/Email… como tengas auth).
2. Al ingresar por primera vez, se crea /users/{suUid} con rol “operador”.
3. Si quieres **forzar** que no edite nada más, déjalo con active: true y listo.
4. A partir de ese momento, en **Alta de Producción** lo verás en el dropdown de **Operador**.

Si quieres permitir que **supervisor** también aparezca como asignable (cuando falten operadores), cambia el filtro a:  
.where('role', whereIn: ['operador', 'supervisor'])

**Siguientes pasos propuestos**

1. **Users Admin Screen** (rápida y útil):
   * Drawer → “Administrar usuarios” (visible solo a administrador)
   * Lista con: nombre, email, rol (dropdown), activo (switch), botón guardar.
   * Así no dependes de la consola para promover/bajar roles.
2. **NewProjectPartScreen** (ya la hicimos multi-P/N):
   * Si quieres, añadimos “**agregar P/N a proyecto existente**” (ya está), y validaciones (evitar duplicados por numeroParte).
3. **Almacén / Stock** (nuevo Drawer):
   * Colección bodegas y inventario o stock (por P/N y bodega).
   * En **Alta de Producción** mostrar “Stock en bodega X: N” y bloquear si no hay disponibilidad (o mostrar warning).
4. **Gantt Producción** (nuevo Drawer):
   * Base: production\_daily con fecha/fechaCompromiso/inicio/fin.
   * Mostramos un timeline por proyecto y part number.
   * Primero una vista simple (ListView por día/semana), luego pasamos a un gráfico.
5. **Operador**:
   * Ya tienes “Mis órdenes” y “Editar orden”.
   * Podemos añadir “subir evidencia” (foto), “pausar/reanudar”, etc.
   * Si quieres que **pass + fail ≤ cantidad** se valide **también en cliente**, lo añadimos (ya lo revisan rules del servidor).

**Resumen corto (para que no se te pierda el hilo)**

* **Todos los usuarios nuevos** se crean como **operador** automáticamente al iniciar sesión (lo hace ProfileScreen).
* **Solo un admin** puede cambiar roles. Hoy puedes hacerlo:
  + A mano en la **Consola** de Firestore (rápido), o
  + Construyendo la **pantalla de Administración de Usuarios** (te la hago cuando me digas).
* El **dropdown de Operador** en el alta muestra docs de /users con role: 'operador' y active: true.
* Ya tienes **alta de proyecto + múltiples P/N** en una sola pantalla (con batch y posibilidad de elegir proyecto existente).
* Lo siguiente natural: **Admin de usuarios**, **Almacén/Stock**, y **Gantt**.

Si quieres, empiezo con la **pantalla de Administración de Usuarios** (lista y editable por admin) y te la paso lista para pegar.