

# Reporte de Corrección de Bug:

## Asignación y Visualización de Roles de Usuario

---

### Problema Identificado

---

Se detectó un bug crítico en el sistema CMMS donde los roles de usuario no se asignaban correctamente al crear o modificar usuarios, y tampoco se mostraban en la interfaz de "Gestión de Perfiles (Usuarios)". Esto afectaba la gestión de permisos y la visibilidad de la información de los usuarios.

### Análisis y Causa Raíz

---

La investigación reveló que el problema residía en la serialización y deserialización de los datos de usuario y rol entre el frontend (React) y el backend (Django REST Framework).

#### Backend (Django REST Framework)

- `cmms_api/serializers.py`: El `UserSerializer` no manejaba adecuadamente la relación entre el modelo `User` de Django y el modelo `Usuarios` personalizado, que contiene el campo `idrol` (el rol del usuario). Específicamente:
  - En la creación (`create` method), no se garantizaba que se creara una instancia de `Usuarios` asociada al `user` recién creado, o que se le asignara un rol por defecto si no se proporcionaba explícitamente.
  - En la actualización (`update` method), no se actualizaba correctamente el `idrol` de la instancia de `Usuarios` existente, o no se creaba una si no existía.

- El campo `nombrero1` (nombre del rol) no se estaba exponiendo correctamente como un campo de solo lectura que obtuviera su valor del modelo `Roles` a través de la relación `idro1`.

## Frontend (React)

- `src/pages/ProfilesView.tsx` : La tabla de visualización de perfiles de usuario no estaba renderizando el nombre del rol porque el campo `nombrero1` no estaba disponible o no se estaba accediendo correctamente desde los datos recibidos del backend.

## Solución Implementada

---

Se realizaron las siguientes modificaciones para corregir el bug:

### Backend (Django REST Framework)

1. `cmms_api/serializers.py` : Se modificó el `UserSerializer` para:
  - Asegurar que, al crear un nuevo usuario, siempre se cree o actualice una instancia de `Usuarios` asociada.
  - Asignar el `idro1` proporcionado durante la creación/actualización. Si no se proporciona un rol, se intenta asignar el rol 'Operador' por defecto, o el primer rol disponible si 'Operador' no existe.
  - Asegurar que el campo `nombrero1` se obtenga correctamente del modelo `Roles` a través de la relación `idro1` en el `UsuariosSerializer` anidado, haciéndolo accesible para el frontend.
  - Se añadió `transaction.atomic()` para asegurar la integridad de los datos al crear usuarios y sus perfiles asociados.

### Frontend (React)

1. `src/pages/ProfilesView.tsx` : Se ajustó la tabla de visualización para renderizar el campo `nombrero1` que ahora es proporcionado por el backend a través del `UserSerializer`.

# Verificación

---

Se realizaron las siguientes pruebas para verificar la corrección:

1. **Creación de Nuevo Usuario:** Se creó un nuevo usuario a través de la interfaz de "Gestión de Perfiles (Usuarios)" y se verificó que el rol seleccionado se asignara y se mostrara correctamente en la tabla.
2. **Edición de Usuario Existente:** Se editó un usuario existente y se cambió su rol. Se confirmó que el rol se actualizara y se reflejara correctamente en la tabla.
3. **Inspección de Base de Datos:** Se accedió a la shell de Django y se verificó directamente en la base de datos que los usuarios tuvieran sus perfiles `Usuarios` asociados y que el `idrol` estuviera correctamente vinculado.

# Conclusión

---

El bug de asignación y visualización de roles de usuario ha sido corregido exitosamente. El sistema ahora maneja los roles de usuario de manera robusta tanto en el backend como en el frontend, mejorando la funcionalidad y la experiencia del usuario en la gestión de perfiles.