# 윰 NetworkLab Platform - Reporte de Avances

Fecha: 19 de Septiembre, 2025

**☑** LOGIN, ROL DE USUARIO, REDIRECCIÓN DE DASHBOARDS Y GUARDS

#### **Arquitectura Implementada**

Backend Stack	
Runtime	Node.js + Express + TypeScript
Base de Datos	MySQL (pool de conexiones)
Autenticación	JWT + bcrypt
Configuración	dotenv, CORS, morgan

<b>⊋</b> Frontend Stack	
Framework	Angular (Standalone Components)
Servicios	HttpClient + Interceptors
Estado	LocalStorage para token/usuario
JI	Estilo minimal y responsive

## **≅** Cambios Realizados - Seguridad y Roles

Área	Detalle	Estado
AuthController.login	Obtiene role desde institution_users + roles, genera JWT con rol y responde en formato plano {'success, token, user'}	COMPLETO
AuthController.refreshToken	Unificado el formato de respuesta a {'success, token, user'}. Rol recalculado desde BD	COMPLETO
AuthController.verifyToken	Respuesta plana {'success, user'} con rol consistente (token o BD)	COMPLETO
database.ts / Model.User.create	Pool de conexiones MySQL con mysql2/promise y variables de entorno. Ajuste de tipos para enviar role: 'student'	COMPLETO
Frontend Auth	Login devuelve Observable, redirige según rol (admin $\rightarrow$ /admin-dashboard, otros $\rightarrow$ /dashboard)	COMPLETO
RoleGuard	Lee rol desde localStorage.role o user.role; soporta expectedRole y allowedRoles	COMPLETO

### **♥** Endpoints Afectados

P Autenticación			
Método	Endpoint	Función	
POST	/auth/login	Login de usuarios $\rightarrow$ {'success, token, user'}	
POST	/auth/refresh	Refresh de tokens $\rightarrow$ {'success, token, user'}	
GET/POST	/auth/verify	Verificación de token → {'success, user'}	

ET /api/users Listado de usuarios ET /api/users/:id Obtener usuario por ID	Vlétodo	Endpoint	Función
ET /api/users/:id Obtener usuario por ID	Tetodo	Liidpoiiit	Tancion
· · · · · · · · · · · · · · · · · · ·	SET	/api/users	Listado de usuarios
· · · · · · · · · · · · · · · · · · ·		/ani/usors/tid	Obtopor usuario por ID
OST /api/users Crear usuario	) L I	/api/users/.iu	Obtener usuano por 10
	OST	/api/users	Crear usuario

### **¹**≡ Próximos Pasos Sugeridos

Prioridad	Módulo	Descripción
ALTA	Middleware de autorización	Proteger rutas /api/admin/* en Express usando req.user.role === 'admin'
ALTA	HTTP Interceptor	Adjuntar Authorization: Bearer y refrescar en 401 automáticamente
MEDIA	Expiración/Warn	Mostrar aviso de sesión por expirar y logout seguro

**MEDIA** 

Swagger

Documentar endpoints y flujos de auth

#### Resultados Obtenidos

- Inicio de sesión con rol funcional y redirección correcta por perfil
- Formato de respuestas unificado en login y refreshToken
- Rutas protegidas por **AuthGuard** y **RoleGuard**; equipment con acceso por múltiples roles
- Pool MySQL estable y tipado TS corregido en creación de usuario
- Estructura de rutas saneada (paths base y comodines consistentes)

© 2025 NetworkLab Platform — Reporte de avances: Login, Roles y Rutas Protegidas