Perfecto, Chema. Aquí tienes el módulo de autenticación funcional en FastAPI, con generación y verificación de **JWT**, adaptado al estilo modular del proyecto PlatSentinel.

**✅ api/app/rutas/autenticacion.py**

from fastapi import APIRouter, Depends, HTTPException, status

from fastapi.security import OAuth2PasswordRequestForm

from jose import jwt, JWTError

from datetime import datetime, timedelta

from app.configuracion import settings

router = APIRouter(prefix="/auth", tags=["Autenticación"])

# Usuarios simulados (puede reemplazarse por DB o LDAP)

usuarios = {

"admin": {"username": "admin", "password": "segura123", "rol": "admin"},

"tecnico": {"username": "tecnico", "password": "token456", "rol": "tecnico"}

}

@router.post("/login")

def login(form\_data: OAuth2PasswordRequestForm = Depends()):

user = usuarios.get(form\_data.username)

if not user or user["password"] != form\_data.password:

raise HTTPException(status\_code=401, detail="Credenciales inválidas")

expiration = datetime.utcnow() + timedelta(minutes=settings.duracion\_token)

payload = {

"sub": user["username"],

"rol": user["rol"],

"exp": expiration

}

token\_jwt = jwt.encode(payload, settings.clave\_jwt, algorithm=settings.algoritmo\_jwt)

return {

"access\_token": token\_jwt,

"token\_type": "bearer"

}

**✅ api/app/dependencias.py (extracto para proteger rutas)**

from fastapi import Depends, HTTPException, status

from fastapi.security import OAuth2PasswordBearer

from jose import jwt, JWTError

from app.configuracion import settings

oauth2\_scheme = OAuth2PasswordBearer(tokenUrl="/auth/login")

def obtener\_usuario\_actual(token: str = Depends(oauth2\_scheme)):

try:

payload = jwt.decode(token, settings.clave\_jwt, algorithms=[settings.algoritmo\_jwt])

return payload

except JWTError:

raise HTTPException(

status\_code=status.HTTP\_401\_UNAUTHORIZED,

detail="Token inválido o expirado"

)

**🧪 Cómo probar**

1. Ejecuta la API: uvicorn app.main:app --reload
2. En Swagger (/docs) ve a POST /auth/login
3. Usa usuario admin y clave segura123
4. Obtendrás un access\_token
5. Copia el token, ve a cualquier ruta protegida y haz clic en “Authorize” → Bearer <token>

¿Quieres que prepare ahora una ruta protegida de ejemplo (/usuarios/me) usando esta autenticación? ¿O seguimos con otro módulo como escaneo\_web.py o generador\_informes.py?