# Entrega - DESAFIO PRACTICO #3

## Introducción al sistema

API REST segura construida con Spring Boot que implementa autenticación JWT y OAuth2 (GitHub).

## Estructura del proyecto y paquetes

Se utilizan los paquetes: config, controller, dto, model, repository, service. El punto de arranque es JwtBasicApplication.java.

## Flujo de autenticación (JWT y OAuth2)

1. El usuario se registra con username/clave (BCrypt). 2. Al hacer login se emite un JWT y un refresh token. 3. Endpoints protegidos usan @PreAuthorize y el filtro JwtAuthenticationFilter valida el token. 4. OAuth2 (GitHub) permite iniciar sesión; si el usuario no existe se crea y se emite un JWT en el success handler.

## Seguridad con roles y @PreAuthorize

Se usan roles (por ejemplo ROLE\_USER) en el UserDetails y anotaciones @PreAuthorize en controladores para limitar acceso a recursos.

## application.properties

El fichero incluye configuración de base de datos, JWT y OAuth2 (client-id y client-secret). En entregas públicas no se deben publicar secretos.

## Pruebas (ejemplo en Postman)

1) POST /api/auth/register -> {username,password,...}\n2) POST /api/auth/login -> recibe token y refreshToken\n3) GET /api/users (con Authorization: Bearer <token>) -> lista usuarios\n4) POST /api/auth/refresh -> enviar refresh token en body para renovar access token

## Conclusiones

Implementación funcional que cubre los requisitos del enunciado. Se añadió endpoint de refresh y un script SQL de ejemplo.