# 🗺️ Mapa Mental: Arquitectura de la API de Riego

* **💧 API SISTEMA DE RIEGO (Nodo Central - Spring Boot)**
  + **Tecnologías:** Java 17, Spring Boot, JPA/Hibernate, MySQL.
  + **Objetivo:** Plataforma centralizada para la gestión, monitoreo y análisis de operaciones de riego agrícola.
  + \*\* शाखा: 🔐 SEGURIDAD (Spring Security + JWT)\*\*
    - **Autenticación (Flujo JWT)**
      * 1. Login: Usuario envía credenciales (/api/auth/login).
      * 2. Token: Servidor valida y devuelve un token JWT.
      * 3. Peticiones: Cliente envía el token en la cabecera Authorization para acceder a rutas protegidas.
    - **Autorización (Basada en Roles)**
      * 👑 ADMIN: Control total del sistema. Gestión de usuarios, fincas y roles.
      * 📊 ANALISTA: Acceso a datos agregados, dashboards, generación de reportes y creación de tareas.
      * 🔧 OPERARIO: Ejecución de tareas y registro de datos de campo (riegos, mantenimientos, etc.).
  + \*\* शाखा: ⚙️ MÓDULOS PRINCIPALES (Endpoints de Gestión)\*\*
    - **Gestión de Estructura Agrícola**
      * /api/farms (Fincas)
      * /api/sectors (Sectores de riego)
      * /api/equipments (Equipos: bombas, válvulas)
      * /api/watersources (Fuentes de agua)
    - **Registro de Operaciones de Campo**
      * /api/irrigation (Registros de Riego)
      * /api/fertilization (Registros de Fertilización)
      * /api/maintenances (Registros de Mantenimiento)
      * /api/operationlogs (Bitácora de Operaciones manuales)
      * /api/precipitations (Registros de Lluvia)
    - **Sensores y Alertas**
      * /api/humiditysensors (Gestión de Sensores de Humedad)
      * /api/humidityalerts (Registro de Alertas de Humedad)
    - **Administración de Usuarios (Solo ADMIN)**
      * /api/admin/users (CRUD de usuarios, asignación de roles y fincas)
      * /api/admin/dashboard (Estadísticas generales de usuarios)
  + \*\* शाखा: 📈 MÓDULOS DE ANÁLISIS Y REPORTES (Roles: ANALISTA, ADMIN)\*\*
    - **Analíticas (Datos para Dashboards y Gráficos)**
      * GET /api/analytics/...
        + .../summary: Totales de agua y horas por sector (ideal para gráficos de barras).
        + .../timeseries: Datos diarios de riego para un sector (ideal para gráficos de líneas).
        + .../records: Registros detallados y paginados de riegos (ideal para tablas de datos).
    - **Reportes (Generación y Descarga de Archivos)**
      * GET /api/reports/generate
        + **Formatos Soportados:** PDF, CSV.
        + **Tipos de Reportes:**

WATER\_BALANCE (Balance Hídrico: Riego vs. Lluvia).

OPERATIONS\_LOG (Bitácora unificada de todas las operaciones).

PERIOD\_SUMMARY (Resumen de métricas clave en un período).

* + \*\* शाखा: ✨ CARACTERÍSTICAS TRANSVERSALES (Servicios de Soporte)\*\*
    - **📝 Auditoría (AuditService)**
      * Registra automáticamente todos los cambios importantes (Quién, Qué y Cuándo).
      * Endpoint: GET /api/audit/change-history.
    - **🔔 Notificaciones (NotificationService)**
      * Alertas en tiempo real para usuarios sobre eventos relevantes (ej. nueva tarea asignada, alerta de lluvia).
      * Endpoint: GET /api/notifications.
    - **🌐 Integraciones con Servicios Externos**
      * **Servicio de Clima:** Consulta APIs de clima para obtener pronósticos y datos actuales.
      * **Servicio de Geocodificación:** Obtiene automáticamente las coordenadas (latitud/longitud) de una finca a partir de su dirección.
    - **📱 Sincronización Móvil**
      * Endpoint (/api/mobile/sync) para que la aplicación móvil de los operarios pueda enviar datos capturados offline.