**03. Diagrama de Arquitectura de Software (UML)**

**Control de riego automático**

* Como **agricultor**, quiero que el sistema me informe sobre el cuidado y el tratamiento de mi riego para tener mejores resultados sobre la cantidad de agua que solicita según las necesidades y estado del riego.

**Monitoreo en tiempo real**

* Como **usuario**, quiero ver los datos de humedad, temperatura y estado de mi riego en mi aplicación, para tomar mejores decisiones sobre el riego.

**Alertas y notificaciones**

* Como **usuario**, quiero recibir notificaciones cuando no haya confirmado realizar las tareas del cuidado del riego, para poder actuar a tiempo.

**Registro histórico**

* Como **usuario**, quiero consultar el historial de riego y humedad, para analizar el consumo de agua y mejorar la planificación.

**Flujo normal**:

1. El usuario ingresa a la aplicación.
2. Selecciona la opción "Consultar riego”".
3. La aplicación envía la orden al sistema.
4. El sistema ejecuta la orden y le da indicaciones.

**Flujo alternativo**:

* Si no hay conexión, la aplicación muestra error y guarda la orden para reenviarla.

**Excepciones**:

* Error en la petición ingresa los datos → notificación inmediata al usuario.

**Requerimientos**

* **Monitoreo en tiempo real** → alertas garantizan información constante.
* **Control manual** → Flexibilidad para que el usuario intervenga en cualquier momento.
* **Notificaciones** → Prevención ante problemas o niveles críticos de humedad.
* **Histórico de datos** → Facilita la toma de decisiones y optimiza el consumo de agua