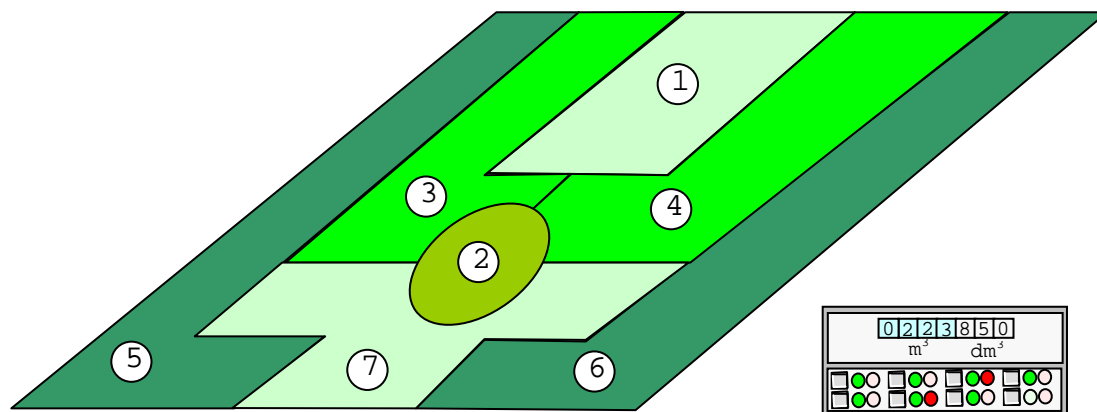


## Proyecto RIEGO.

Se desea controlar el sistema de riego de un parque con un computador. El parque dispone actualmente de 7 zonas de riego independientes. Cada una de ellas tiene una línea de riego propia que cuando está abierta suministra 2 litros/s de agua. Su flujo es controlado por una electroválvula que acciona el computador. A las válvulas de las líneas se accede desde el computador a través del registro de I/O 16#0100# que puede ser leído y escrito por el computador. Este registro es de 16 bits, y la línea "i" se abre si se establece a 1 el bit "i" del registro, y se cierra si se establece a 0 el mismo bit. El estado que se lee del mismo bit representa la presencia o ausencia de flujo en la línea, lo cual depende del estado de la válvula, de que haya presión en la red de agua y de que no haya obturación en la línea.



El control de riego se lleva a cabo abriendo y cerrando las líneas de riego, teniendo en cuenta el nivel de humedad específica de cada zona. Para ello se han instalado bajo tierra un sistema de transductores de humedad, que proporcionan una medida lineal codificada, de forma que cuando se lee el código 4095, la humedad es del 100%, y si se lee 0, la humedad es del 0%. Al conjunto de sensores de humedad se accede a través de dos registros de 16 bits: El registro de requerimiento y estatus (Reg\_Req&St) es de lectura y escritura y el registro de datos (Reg\_Data) es de solo lectura. Ambos son del bus I/O y sus direcciones son 16#0102# y 16#0104#. Para requerir el nivel de humedad de la zona i se debe establecer el bit i en Reg\_Req, y luego esperar unos segundos a que la medida haya concluido y el dato de humedad esté disponible para su lectura en el registro Reg\_Data. El protocolo que debe seguirse para realizar una lectura de la humedad de la zona i contiene los siguientes pasos: en primer lugar se establece el bit i del registro Reg\_Req a 1; en segundo lugar, hay que vigilar el estado que se lee en el mismo bit i del mismo registro Reg\_Req, que será 0 si aun no se ha realizado la medida y 1 si la medida se ha realizado, por último, cuando ya sabemos que la medida se ha realizado, se lee del registro Reg\_Data el valor de la medida.

Cada zona, en función del cultivo que tiene, presenta unas necesidades de humedad diferentes que se formula mediante una humedad mínima  $Min\_Hum_i$  que pueden ser programadas por el jardinero. La zona i se considera que requiere riego si su humedad se encuentra por debajo de  $Min\_Hum_i$ . Cuando esto ocurre, debe abrirse la correspondiente manguera durante un período que será función de la hora del día de que se trate: si se encuentra dentro del intervalo horario comprendido entre las 9:00 y las 17:00 horas, la manguera debe estar abierta 20 minutos, mientras que si se encuentra fuera de éste intervalo horario, la manguera debe estar abierta sólo 10 minutos. En los 30 minutos posteriores a una fase de riego no deben hacerse medidas de los

niveles de humedad de la zona, ya que no es significativa por no haber penetrado aún el agua hasta el sensor.

Si durante el riego de una zona se interrumpe el suministro de agua, el sistema debe mantener la fase de riego, manteniendo la válvula abierta y vigilando que el tiempo de riego se satisfaga. En cualquier caso, transcurrido un periodo de 30 minutos desde que se finalizó el riego (bien porque se terminó el riego, o bien porque se produjo un fallo) se debe medir el nivel de humedad (mientras que no se ha regado pudo llover y ya puede no ser necesario el riego), e iniciar una nueva fase de riego.

Se dispone de un panel de control del sistema de riego. Por cada zona existe un conmutador y dos pilotos (rojo y verde). El conmutador habilita o inhibe el programa riego de la zona. El piloto verde se ilumina cuando se está regando la zona, y el piloto rojo se ilumina cuando hay un error en el control de la línea de la zona (este error puede ser que no fluye agua con la válvula abierta, que fluye agua con la válvula cerrada o que el sistema de medida de la humedad realiza medidas incorrectas). A los conmutadores y pilotos se accede por dos registros de 16 bits. El Reg\_Control es un registro de lectura y escritura I/O de dirección 16#0106#, que cuando se lee retorna en el bit i el estado del conmutador i del panel de control, mientras que si se escribe en él, se establece el estado de los pilotos verdes (si el bit i se establece a 1 el piloto se enciende, y si se establece a 0 el piloto se apaga). El Reg\_Error es un registro de solo escritura del bus I/O de dirección 16#108# y controla el estado de los pilotos rojos (si el bit i se establece a 1 el correspondiente piloto rojo se enciende, y si se establece a 0 el piloto se apaga).

El panel de control tiene también un display alfanumérico del consumo de agua debido al riego del parque. Para presentar este valor se disponen de dos registros de 16 bits y de solo escritura Display\_m3 y Display\_dm3, y a los que se accede por la dirección I/O 16#010A# y 16#010C#.

Aunque en la organización actual del parque sólo hay 7 zonas de riego instaladas, debe diseñarse el programa, de forma que este número sea configurable sin necesidad de modificar el programa.

Puede ser interesante, proponer una versión del software en la que el panel de control sea una ventana en la pantalla del ordenador que pueda controlarse con el teclado o el ratón.

## **Objetivos**

Los objetivos de la práctica son:

- La especificación, el análisis orientado a objetos de los patrones de interacción entre objetos, y del diseño de la aplicación para ser implementado en Java. La herramienta a utilizar para estas actividades es UML.
- El diseño e implementación de una interfaz gráfica de usuario utilizando la familia de componentes SWT de Eclipse.
- La documentación que se genera debe ser la adecuada para que un programador que conozca el lenguaje de programación Java, pueda codificar cualquier clase sin que tenga que tomar decisiones de diseño.

## **Tareas:**

- Tarea 1: Realizar la especificación de usuario y detallada de la aplicación utilizando diagramas de caso de uso, y diagramas de contexto.
- Tarea 2: Realizar el análisis de la aplicación, formulando el diagrama de clases de la aplicación:
- Identificar las clases que constituyen la aplicación, y de cada una de ellas identificar los atributos que constituyen su estado y los métodos que constituyen su interfaz. Establecer entre las clases las relaciones de herencia, agregación, dependencia y visibilidad que existen. Documentar todos los elementos que se incluyan (clases, atributos, métodos y relaciones).
  - Describir mediante diagramas de actividad los métodos cuya lógica sea compleja.
  - Describir mediante diagramas de estado las clases que tenga un comportamiento complejo.
- Tarea 3: Mostrar los patrones de interacción básicos entre objetos. Documentar mediante diagramas de secuencias las interacciones que tengan una complejidad que lo requiera.
- Tarea 4: Validar el análisis mediante la comprobación de que todos los casos de uso pueden ser implementados utilizando las clases propuestas.
- Tarea 5: Realizar el diseño de la aplicación para que sea implementado utilizando Java:
- Mapear los tipos y los contenedores del análisis a tipos Java.
  - Realizar el diseño de concurrencia de las aplicaciones y proponer el framework de de ejecución. Identificar las clases que son activas y los threads que requieren la ejecución.
- Tarea 6: Diseñar la interfaz de usuario utilizando la familia de componentes SWT de Eclipse.
- Tarea 7: Generar el esqueleto de código Java comprobando la completitud del diseño.
- Tarea 8: Recopilar en un breve documento (20 páginas) la información útil para el programador que se ha generado en las anteriores tareas.