

Universidad San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas
Introducción a la Programación y Computación 2

Ing. Claudia Liceth Rojas Morales
Ing. Marlon Antonio Pérez Türk
Ing. José Manuel Ruiz Juárez
Ing. Dennis Stanley Barrios Gonzalez
Ing. Edwin Estuardo Zapeta Gómez
Ing. Fernando José Paz González

Tutores de curso:
Angely Naomi Marroquín Tapaz
Diego Andrés Huite Alvarez
Hesban Amilcar Argueta Aguilar
Pedro Luis Pu Tavico
Angel Miguel García Urizar
Luis Antonio Castillo Javier



PROYECTO 2

OBJETIVO GENERAL

Modelar, documentar e implementar una solución al problema que se plantea utilizando las herramientas de desarrollo presentadas en clase y laboratorio.

OBJETIVOS ESPECÍFICOS

- Implementar una solución utilizando el lenguaje de programación Python.
- Utilizar estructuras de programación secuenciales, cíclicas y condicionales.
- Generar reportes con la herramienta Graphviz.
- Manipular archivos XML.
- Utilizar los conceptos de TDA y aplicarlos a memoria dinámica.
- Utilizar estructuras de programación propias.
- Utilizar el paradigma de programación orientada a objetos.

ENUNCIADO

Este problema consiste en optimizar un sistema de riego y aplicación de fertilizante robótico. La empresa GuateRiegos 2.0 ha creado un sistema innovador para optimizar la cantidad de agua, fertilizante y el tiempo necesario para mantener saludables los cultivos en invernaderos.

Un invernadero está formado por hileras o filas de cultivos. Cada hilera tiene X plantas y cada planta ocupa exactamente 1 metro cuadrado de espacio.

La empresa GuateRiegos ha construido robots regadores, estos robots son drones con un sistema de riego y aplicación de fertilizante incorporado, estos drones pueden ser programados para volar un metro hacia adelante, un metro hacia atrás, o bien, aplicar el fertilizante y el agua a la planta. En la Imagen 1 se observa un robot regador.



Imagen 1: Robot regador (Meta AI image)

Cada planta requiere una cantidad específica de litros de agua y una cantidad gramos de fertilizante para sobrevivir y crecer saludablemente. Además, cada planta tiene su propio calendario de riego.

El proceso de riego y aplicación de fertilizante automatizado desarrollado por GuateRiegos 2.0 consiste en colocar un robot regador al inicio de cada hilera de cultivos y configurar el plan de riego y aplicación de fertilizante que se desea aplicar al invernadero. En la imagen 2 se puede observar la configuración inicial para realizar el riego y aplicación de fertilizante en un invernadero de 3 hileras y 4 plantas en cada hilera.

El plan de riego se configura de la siguiente forma:

H1-P2, H2-P1, H2-P2, H3-P3, H1-P4 donde H representa la hilera y P representa la posición de la planta en esa hilera.

El riego de las plantas debe realizarse en el orden indicado según el plan, solamente pueden regar una planta en un momento dado, es decir, no se deben regar 2 o más plantas al mismo tiempo en un invernadero para optimizar la eficacia del uso del agua.

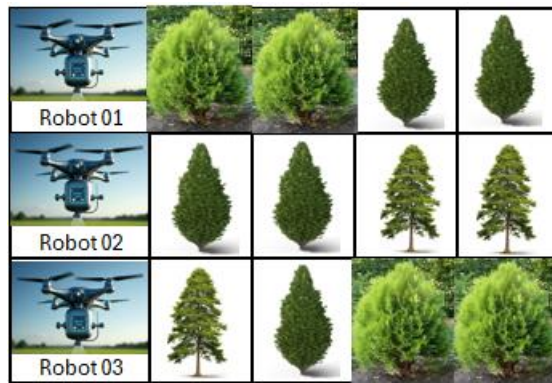


Imagen 2: Estado inicial de un invernadero preparado para utilizar el sistema de riego y aplicación de fertilizante robótico para un invernadero con 3 hileras de 4 plantas cada hilera

Usted ha sido contratado para desarrollar un software que genere las instrucciones para que un conjunto de “N” drones, asignados a cada hilera de un invernadero, rieguen y apliquen fertilizante a las plantas según el plan de riego y aplicación de fertilizante configurado y determine la cantidad de agua en litros y la cantidad de fertilizante en gramos utilizados aplicando las siguientes reglas:

1. Los drones regadores demoran 1 segundo en moverse 1 metro hacia adelante o hacia atrás.
2. Los drones regadores demoran 1 segundo en regar una planta.
3. Solamente 1 dron puede realizar la operación de regado en un momento dado en un invernadero.
4. Los riegos deben seguir el orden establecido en el plan de riego y aplicación de fertilizante configurado para el invernadero.
5. Para cada planta de cada hilera se establece la cantidad de agua en litros y la cantidad de fertilizante en gramos que se desea aplicar.

Ejemplo:

Para el invernadero que se muestra en la imagen 2, se determina que cada planta requiere 1 litro de agua y 100 gramos de fertilizante y se configura el siguiente plan de riego y aplicación de fertilizante: H1-P2, H2-P1, H2-P2, H3-P3, H1-P4 y se asignan los drones regadores de acuerdo con la siguiente tabla:

Hilera	Dron
H1	DR01
H2	DR02
H3	DR03

Tabla 1 – Asignación de drones a hileras de un invernadero

Instrucciones enviadas a cada Dron para ejecutar el plan de riego y aplicación de fertilizante por unidad de tiempo: [H1-P2, H2-P1, H2-P2, H3-P3, H1-P](#)

Tiempo	DR01	DR02	DR03
1 segundo	Adelante (H1P1)	Adelante (H2P1)	Adelante (H3P1)
2 segundos	Adelante (H1P2)	Esperar ¹	Adelante (H3P2)
3 segundos	Regar	Esperar	Adelante (H3P3)
4 segundos	Adelante (H1P3)	Regar	Esperar ²
5 segundos	Adelante (H1P4)	Adelante (H2P2)	Esperar
6 segundos	Esperar	Regar	Esperar
7 segundos	Esperar	FIN ³	Regar
8 segundos	Regar		FIN
9 segundos	FIN		

Tabla 2 – Detalle de instrucciones enviadas a cada dron para realizar el plan de riego y aplicación de fertilizante configurado para cada invernadero

Información para aplicar el plan de riego y aplicación de fertilizante robotizado al invernadero:

- Tiempo para regado óptimo: 8 segundos
- Agua requerida por dron:

DR01 - 2 litros
DR02 - 2 litros
DR03 - 1 litro
TOTAL: 5 litros

- Fertilizante requerido por dron:

¹ El dron DR02 ya se encuentra en la planta que debe regar, pero no riega porque no es el momento de acuerdo con el plan de riego y aplicación de fertilizante.

² El dron DR03 ya se encuentra en la planta que debe regar, pero no riega porque no es el momento de acuerdo con el plan de riego y aplicación de fertilizante.

³ El dron finalizó su tarea automatizada y debe regresar al estado inicial (inicio de la hilera).

DR01 - 200 gramos
DR02 - 200 gramos
DR03 - 100 gramos
TOTAL: 500 gramos

Archivo de Entrada y Salida

Los archivos de entrada y salida consistirán en archivos con extensión y estructura xml, a continuación, se muestra un ejemplo de archivo de entrada:

Ejemplo de Entrada: entrada.xml

```
<?xml version="1.0"?>
<configuracion>
  <listaDrones>
    <dron id=1 nombre='DR01' />
    <dron id=2 nombre='DR02' />
    <dron id=3 nombre='DR03' />
    ...
  </listaDrones>
  <listaInvernaderos>
    <invernadero nombre='Invernadero Zacapa'>
      <numeroHileras> 3 </numeroHileras>
      <plantasXhilera> 4 </plantasXhilera>
      <listaPlantas>
        <planta hilera=1 posicion=1 litrosAgua=1 gramosFertilizante=100> ciprés
      </planta>
        <planta hilera=1 posicion=2 litrosAgua=1 gramosFertilizante=100> ciprés
      </planta>
        <planta hilera=1 posicion=3 litrosAgua=1 gramosFertilizante=100> ciprés
italiano </planta>
        <planta hilera=1 posicion=4 litrosAgua=1 gramosFertilizante=100> ciprés
italiano </planta>
        <planta hilera=2 posicion=1 litrosAgua=1 gramosFertilizante=100> ciprés
italiano </planta>
        <planta hilera=2 posicion=2 litrosAgua=1 gramosFertilizante=100> ciprés
italiano </planta>
        <planta hilera=2 posicion=3 litrosAgua=1 gramosFertilizante=100> ciprés
de tarout </planta>
        <planta hilera=2 posicion=4 litrosAgua=1 gramosFertilizante=100> ciprés
de tarout </planta>
        <planta hilera=3 posicion=1 litrosAgua=1 gramosFertilizante=100> ciprés
de tarout </planta>
        <planta hilera=3 posicion=2 litrosAgua=1 gramosFertilizante=100> ciprés
itiano </planta>
        <planta hilera=3 posicion=3 litrosAgua=1 gramosFertilizante=100> ciprés
</planta>
        <planta hilera=3 posicion=4 litrosAgua=1 gramosFertilizante=100> ciprés
</planta>
      </listaPlantas>
      <asignacionDrones>
        <dron id=1 hilera=1/>
        <dron id=2 hilera=2/>
        <dron id=3 hilera=3/>
      </asignacionDrones>
    </invernadero>
  </listaInvernaderos>
</configuracion>
```

```

    <planesRiego>
      <plan nombre='Semana 1'> H1-P2, H2-P1, H2-P2, H3-P3, H1-P4 </plan>
      <plan nombre='Semana 2'> H3-P1, H1-P3, H2-P4, H3-P4, H2-P3, H1-P1, H3-P2
    </plan>
      ...
    </planesRiego>
  </invernadero>
  ...
</listaInvernaderos>
</configuracion>

```

Cada archivo de entrada cargado creará toda la configuración del sistema eliminando las configuraciones previas.

Ejemplo de Salida: salida.xml

```

<?xml version="1.0"?>
<datosSalida>
  <listaInvernaderos>
    <invernadero nombre='Invernadero Zacapa'>
      <listaPlanes>
        <plan nombre='Semana 1'>
          <tiempoOptimoSegundos> 8 </tiempoOptimoSegundos>
          <aguaRequeridaLitros> 5 </aguaRequeridaLitros>
          <fertilizanteRequeridoGramos> 500 </fertilizanteRequeridoGramos>
          <eficienciaDronesRegadores>
            <dron nombre='DR01' litrosAgua=2 gramosFertilizante=200/>
            <dron nombre='DR02' litrosAgua=2 gramosFertilizante=200/>
            <dron nombre='DR03' litrosAgua=1 gramosFertilizante=100/>
          </eficienciaDronesRegadores>
          <instrucciones>
            <tiempo segundos=1>
              <dron nombre='DR01' accion='Adelante (H1P1)' />
              <dron nombre='DR02' accion='Adelante (H2P1)' />
              <dron nombre='DR03' accion='Adelante (H3P1)' />
            </tiempo>
            ...
          </instrucciones>
        </plan>
        ...
      </listaPlanes>
    </invernadero>
    ...
  </listaInvernaderos>
</datosSalida>

```

REPORTES

Reporte HTML - ReporteInvernaderos.html

Para cada invernadero configurado en el sistema, se debe generar un reporte en HTML que muestre por cada plan de riego y aplicación de fertilizante configurado en dicho invernadero, los drones asignados a cada hilera (tabla 1), las instrucciones enviadas a cada dron en el tiempo (tabla2) y las estadísticas de uso de agua y fertilizante totales y por dron.

Este reporte se deberá generar por cada invernadero configurado un archivo de salida en formato HTML. queda a discreción del estudiante como representar la información, pero debe considerar que debe observarse la actividad de cada Dron.

Reporte de TDAs

Utilizando la herramienta Graphviz, se deberá crear un grafo mostrando el estado de los TDAs utilizados para generar la funcionalidad de un plan de riego y aplicación de fertilizante. Este grafo debe poder generarse en cualquier momento y debe mostrar el estado de los TDAs en ese momento por medio de una opción en la interfaz de usuario que será descrita en la siguiente sección. El siguiente es un ejemplo de una secuencia de trabajo graficada.



Imagen 3 – Grafo representativo de TDA

INTERFAZ DE USUARIO

Se debe crear una interfaz de usuario, en la que se puedan gestionar todas las acciones necesarias para la ejecución lógica del proyecto, queda a discreción del estudiante como elaborar su propio diseño para la aplicación, pero debe considerar la facilidad de uso por parte del usuario.

La interfaz de usuario debe permitir realizar las siguientes operaciones:

- Debe existir una opción para cargar archivos con la configuración de invernaderos.
- Se debe poder seleccionar un invernadero y un plan de riego y aplicación de fertilizante, según lo configurado en el archivo de entrada, para poder simular el proceso de riego.
 - Mostrar las estadísticas del proceso: tiempo óptimo para realizar el riego, litros de agua ocupados por dron, gramos de fertilizante ocupados por dron.
 - Generar y mostrar el reporte html del proceso de riego y aplicación de fertilizante automatizado para el invernadero y plan seleccionado.
 - Permitir al usuario definir un tiempo "t" en segundos. Generar y mostrar la gráfica de estado de los TDAs en dicho momento utilizados para el proceso de riego y aplicación de fertilizante automatizado para el invernadero y plan seleccionado. Permitir cambiar el valor de "t" con el objetivo de evaluar el algoritmo utilizado para optimizar el proceso de riego y aplicación de fertilizante en el invernadero y plan seleccionados.
- Generar reporte HTML (ver sección Reporte HTML)
- Generar archivo de salida con los resultados para todos los invernadero y planes de riego y aplicación de fertilizante configurados.
- Apartado de ayuda (Incluir información del estudiante, Acerca de la aplicación y enlace hacia la documentación).
- Se tomará en cuenta la creatividad del estudiante.
- La interfaz de usuario debe ser web utilizando Flask.

CONSIDERACIONES

Se deberá realizar la implementación utilizando programación orientada a objetos, algoritmos desarrollados por el estudiante e **implementación de estructuras a través de Tipos de Dato Abstracto (TDA) propios del estudiante**; que permita almacenar la información de los archivos de entrada y poder interactuar con dicha información. El estudiante deberá abstraer la información y definir qué estructuras implementar que le faciliten la solución. Por lo que puede implementar pilas, colas, listas simples, dobles, circulares, listas de listas, listas n-enlazadas o listas ortogonales para poder solventar el proyecto. No está permitido el uso de estructuras propias de Python (list, dict, tuple, set).

Debe utilizarse versionamiento para el desarrollo del proyecto. Se utilizará la plataforma **Github** en la cual se debe crear un repositorio en el que se gestionará el proyecto. Se deben realizar 4 releases o versiones del proyecto (se recomienda realizar una por semana del tiempo disponible) mínimo. **Se deberá agregar a su respectivo auxiliar como colaborador del repositorio**. El último release será el release final y se deberá de realizar antes de entregar el proyecto en la fecha estipulada.

DOCUMENTACIÓN

Para que el proyecto sea calificado, el estudiante deberá entregar la documentación utilizando el formato de **ensayo definido** para el curso. En el caso del proyecto, el ensayo debe tener entre 4 y 7 páginas de contenido, este máximo no incluye los apéndices o anexos donde se pueden mostrar modelos y diseños utilizados para construir la solución. Es obligatorio incluir el **diagrama de clases** que modela la solución de software presentada por el estudiante y los **diagramas de actividades** con los principales algoritmos implementados en la solución.

RESTRICCIONES

- Solo se permitirá la utilización de los IDEs discutidos en el laboratorio.
- **Uso obligatorio de programación orientada a objetos (POO) desarrollada por completo por el estudiante. De no cumplir con la restricción, no se tendrá derecho a calificación.**
- El nombre del repositorio debe de ser **IPC2_Proyecto2_#Carnet**.
- El estudiante debe entregar la documentación solicitada para poder optar a la calificación.
- Los archivos de entrada no podrán modificarse.
- Los archivos de salida deben llevar la estructura mostrada en el enunciado obligatoriamente.
- Deben existir 4 releases uno por cada semana, de esta manera se corrobora el avance continuo del proyecto. **Se definirá una penalización por cada release faltante.**
- Se calificará de los cambios realizados en el cuarto release. Los cambios realizados después de ese release no se tomarán en cuenta.
- Cualquier caso de copia parcial o total tendrá una nota de 0 y será reportada a Escuela de Ciencias y Sistemas.
- Para dudas concernientes al proyecto se utilizarán los foros en UEDI de manera que todos los estudiantes puedan ver las preguntas y las posteriores respuestas.
- **Utilización del framework Flask** para desarrollar la interfaz de usuario.
- **NO HABRÁ PRÓRROGA.**

ENTREGA

- La entrega será el **26 de septiembre** a las 11:59 pm como máximo.
- La entrega será por medio de la UEDI.
- La documentación debe estar subida en el repositorio en una carpeta separada.
- Para entregar el proyecto en UEDI se deberá subir un archivo de texto con el link del repositorio.