

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 1

OBJETIVO GENERAL

Desarrollar una solución integral que implemente tipos de datos abstractos (TDA) y visualización de datos (Graphviz) bajo el concepto de programación orientada a objetos (POO).

OBJETIVOS ESPECÍFICOS

- Implementar POO para el desarrollo de la solución a través de lenguaje Python.
- Utilizar estructuras de programación secuenciales, cíclicas y condicionales
- Visualizar TDA's por medio de la herramienta Graphviz.
- Utilizar archivos XML como insumos para la lógica y comportamiento de la solución.

ENUNCIADO

Este problema consiste en optimizar un sistema de agricultura de precisión. La agricultura de precisión considera la variabilidad espacial y temporal en los campos de cultivo, utiliza tecnologías avanzadas como GPS, GNSS, SIG, sensores, drones, imágenes satelitales y máquinas inteligentes.

El beneficio de utilizar el sistema de agricultura de precisión radica en recopilar información para gestionar zonas específicas, optimizando insumos y aumentando la sostenibilidad y rentabilidad de las actividades agrícolas.

Para implementar el sistema de agricultura de precisión se utilizarán los siguientes componentes tecnológicos:

- Sensores de suelo fijos para medir la humedad del suelo a distintas profundidades, temperatura del suelo, salinidad, conductividad eléctrica, nutrientes clave y PH.
- Sensores de cultivo fijos en el campo para evaluar el estado del cultivo a través de índices vegetales, estrés hídrico/térmico, cobertura, biomasa y detección temprana de enfermedades.
- Estaciones base agrometeorológicas para monitorear condiciones atmosféricas en tiempo real (temperatura, humedad relativa, radiación solar, etc.).

El sistema de agricultura de precisión es soportado por una plataforma tecnológica construida en la nube, esta plataforma recibe la información recolectada por estaciones base agrometeorológicas, sensores de cultivo y sensores de suelo, esta información es almacenada y analizada a través de algoritmos de Machine Learning para implementar acciones que serán transmitidas a máquinas inteligentes para actuar en tiempo real y garantizar un riego oportuno y eficiente, la aplicación de fertilizante según demandas reales del suelo y de las plantas; además, este sistema permite implementar la detección temprana de problemas y maximizar el rendimiento de forma sostenible.

Para optimizar la infraestructura de recolección de datos agrometeorológicos, de suelo y de cultivo, se han adquirido estaciones base capaces de conectarse a Internet y alimentar la plataforma en la nube. Los sensores de suelo y los sensores de cultivo transmiten su información utilizando redes inalámbricas hacia las estaciones base para reducir costos.

Usted ha sido contratado para optimizar el uso de estaciones base, sensores de suelo y sensores de cultivo que alimentan el sistema completo de agricultura de precisión.

El problema de diseño de distribución de estaciones base consiste en determinar la cantidad óptima de estaciones base necesarias para recolectar toda la información en tiempo real, por lo que se trata de un problema combinatorio NP-Hard. Algunas de las situaciones comunes observadas cuando se resuelven instancias muy grandes de un problema NP-Hard son: fuerte requerimiento de tiempo y fuerte demanda de recursos de memoria, por tal razón, se propone utilizar una metodología de agrupamiento para resolver este problema.

La metodología de agrupamiento propuesta consiste en identificar una matriz de frecuencias para “n” estaciones base y “s” sensores de suelo ($F[n,s]$); y otra matriz de frecuencias para las mismas “n” estaciones base y “t” sensores de cultivo ($F[n,t]$) implementados en un campo agrícola “x”, identificando de esta forma qué estación base recibe datos de qué sensores de suelo y de cultivo.

El proceso de agrupamiento consiste en transformar las matrices de frecuencias $F[n,s]$ y $F[n,t]$ en sus respectivas matrices de patrones $F_p[n,s]$ y $F_p[n,t]$ y agrupar las filas con el mismo patrón, de esta forma se obtiene el uso óptimo de estaciones base para transmitir toda la información recolectada en tiempo real hacia la plataforma en la nube.

Ejemplo:

Para un campo de 5 estaciones base, 3 sensores de suelo y 1 sensor de cultivo tenemos:

$$F[n,s] = \begin{matrix} & s_1 & s_2 & s_3 \\ \begin{matrix} n_1 \\ n_2 \\ n_3 \\ n_4 \\ n_5 \end{matrix} & \begin{bmatrix} 200 & 300 & 0 \\ 0 & 0 & 6000 \\ 500 & 8000 & 0 \\ 1500 & 0 & 1500 \\ 0 & 0 & 2000 \end{bmatrix} \end{matrix} \quad F[n,t] = \begin{matrix} & t_1 \\ \begin{matrix} n_1 \\ n_2 \\ n_3 \\ n_4 \\ n_5 \end{matrix} & \begin{bmatrix} 3500 \\ 2000 \\ 1000 \\ 950 \\ 3200 \end{bmatrix} \end{matrix}$$

Entonces, sus matrices de patrones serían:

$$F_p[n,s] = \begin{matrix} & s_1 & s_2 & s_3 \\ \begin{matrix} n_1 \\ n_2 \\ n_3 \\ n_4 \\ n_5 \end{matrix} & \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \end{matrix} \quad F_p[n,t] = \begin{matrix} & t_1 \\ \begin{matrix} n_1 \\ n_2 \\ n_3 \\ n_4 \\ n_5 \end{matrix} & \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \end{matrix}$$

Se puede observar que las frecuencias para las estaciones n_1 y n_3 ; así como para las estaciones n_2 y n_5 tienen el mismo patrón tanto en $F_p[n,s]$ como en $F_p[n,t]$, por lo tanto, podrán agruparse y optimizar del uso de 5 estaciones base al uso de 3 estaciones base para transmitir en tiempo real toda la información recabada por los sensores de suelo y los sensores de cultivo. La frecuencia para las matrices con las estaciones reducidas ($F_r[n,s]$ y $F_r[n,t]$) se obtiene al sumar las frecuencias de las estaciones agrupadas.

$$F_r[n,s] = \begin{matrix} & s_1 & s_2 & s_3 \\ \begin{matrix} n_{1,3} \\ n_{2,5} \\ n_4 \end{matrix} & \begin{bmatrix} 700 & 8300 & 0 \\ 0 & 0 & 8000 \\ 1500 & 0 & 1500 \end{bmatrix} \end{matrix} \quad F_r[n,t] = \begin{matrix} & t_1 \\ \begin{matrix} n_{1,3} \\ n_{2,5} \\ n_4 \end{matrix} & \begin{bmatrix} 4500 \\ 5200 \\ 950 \end{bmatrix} \end{matrix}$$

Se debe diseñar un programa que acepte “X” campos agrícolas con sus respectivas estaciones base, sensores de suelo y sensores de cultivo, así como las frecuencias de transmisión de datos para producir el uso óptimo de estaciones para cada campo agrícola.

REPORTES

Se deberá utilizar la herramienta Graphviz para crear un grafo que muestre de manera gráfica las matrices de frecuencia $F[n,s]$ y $F[n,t]$, las matrices de patrones $F_p[n,s]$ y $F_p[n,t]$ o las matrices reducidas $F_r[n,s]$ y $F_r[n,t]$ para cualquier campo agrícola configurado en la aplicación a desarrollar.

Archivos 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:

```
<?xml version="1.0"?>
<camposAgricolas>
  <campo id="01" nombre="Campo agrícola 01">
    <estacionesBase>
      <estacion id="e01" nombre="Estacion 01"/>
      <estacion id="e02" nombre="Estacion 02"/>
      <estacion id="e03" nombre="Estacion 03"/>
      <estacion id="e04" nombre="Estacion 04"/>
      <estacion id="e05" nombre="Estacion 05"/>
      ...
    </estacionesBase>
    <sensoresSuelo>
      <sensorS id="s01" nombre="Sensor S01">
        <frecuencia idEstacion="e01"> 200 </frecuencia>
        <frecuencia idEstacion="e03"> 500 </frecuencia>
        <frecuencia idEstacion="e04"> 1500 </frecuencia>
      </sensorS>
      <sensorS id="s02" nombre="Sensor S02">
        <frecuencia idEstacion="e01"> 300 </frecuencia>
        <frecuencia idEstacion="e03"> 8000 </frecuencia>
      </sensorS>
      <sensorS id="s03" nombre="Sensor S03">
        <frecuencia idEstacion="e02"> 6000 </frecuencia>
        <frecuencia idEstacion="e04"> 1500 </frecuencia>
        <frecuencia idEstacion="e05"> 2000 </frecuencia>
      </sensorS>
      ...
    </sensoresSuelo>
    <sensoresCultivo>
      <sensorT id="t01" nombre="Sensor T01">
        <frecuencia idEstacion="e01"> 3500 </frecuencia>
        <frecuencia idEstacion="e02"> 2000 </frecuencia>
        <frecuencia idEstacion="e03"> 1000 </frecuencia>
        <frecuencia idEstacion="e04"> 950 </frecuencia>
        <frecuencia idEstacion="e05"> 3200 </frecuencia>
      </sensorT>
    </sensoresCultivo>
  </campo>
  ...
</camposAgricolas>
```

```

        </sensorT>
        ...
    </sensoresCultivo>
</campo>
...
</camposAgricolas>

```

Ejemplo de Salida:

```

<?xml version="1.0"?>
<camposAgricolas>
  <campo id="01" nombre="Campo agrícola 01">
    <estacionesBaseReducidas>
      <estacion id="e01" nombre="Estacion 01, Estacion 03"/>
      <estacion id="e02" nombre="Estacion 02, Estacion 05"/>
      <estacion id="e04" nombre="Estacion 04"/>
      ...
    </estacionesBaseReducidas>
    <sensoresSuelo>
      <sensorS id="s01" nombre="Sensor S01">
        <frecuencia idEstacion="e01"> 700 </frecuencia>
        <frecuencia idEstacion="e04"> 1500 </frecuencia>
      </sensorS>
      <sensorS id="s02" nombre="Sensor S02">
        <frecuencia idEstacion="e01"> 8300 </frecuencia>
      </sensorS>
      <sensorS id="s03" nombre="Sensor S03">
        <frecuencia idEstacion="e02"> 8000 </frecuencia>
        <frecuencia idEstacion="e04"> 1500 </frecuencia>
      </sensorS>
      ...
    </sensoresSuelo>
    <sensoresCultivo>
      <sensorT id="t01" nombre="Sensor T01">
        <frecuencia idEstacion="e01"> 4500 </frecuencia>
        <frecuencia idEstacion="e02"> 5200 </frecuencia>
        <frecuencia idEstacion="e04"> 950 </frecuencia>
      </sensorT>
      ...
    </sensoresCultivo>
  </campo>
  ...
</camposAgricolas>

```

La aplicación deberá contar con un menú en consola, con las siguientes opciones

Menú principal:

1. Cargar archivo
2. Procesar archivo
3. Escribir archivo salida
4. Mostrar datos del estudiante
5. Generar gráfica
6. Salida

1. **Cargar Archivo:** Esta Opción solicitará la ruta y el nombre del archivo a cargar.

Opción cargar archivo

Ingrese la ruta del archivo:

Ingrese el nombre del archivo:

2. **Procesar el Archivo:** Esta opción será la encargada de procesar la información cargada en memoria, durante el proceso se deben ir mostrando mensajes al usuario para tener el conocimiento de lo que está pasando en el sistema.

➤ Cargando campo agricola 01

➤ Creando estación base e01

...

3. **Escribir Archivo de salida:** Esta opción será la encargada de escribir el archivo con la salida específica.

Opción generar archivo de salida

Ingrese la ruta del archivo:

Ingrese el nombre del archivo:

4. **Mostrar datos del estudiante:** Mostrar los datos del estudiante, carné, nombre, curso, carrera, semestre y enlace de acceso a documentación.

➤ Nombre del estudiante

➤ Carnet del estudiante

➤ Introducción a la Programación y Computación 2

➤ Sección del estudiante

➤ 4to. Semestre

➤ Enlace a documentación

5. **Generar gráfica:** El programa deberá permitir que el usuario elija un campo agrícola ingresado en el archivo de entrada, luego debe solicitar si se desea graficar la matriz

de frecuencias, la matriz de patrones o la matriz reducida de dicho campo agrícola. Finalmente, se mostrará la gráfica (ver sección “REPORTES”).

CONSIDERACIONES

Se deberán implementar listas anidadas creadas por el estudiante, creando clases Nodo y clases lista, de tal manera que al recorrer las listas se pueda obtener la información de cada campo agrícola, y en base en estas listas se podrá generar la información y el reporte gráfico de cada campo agrícola.

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). **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 estudiante¹. De no cumplir con la restricción, no se tendrá derecho a calificación.
- El nombre del repositorio debe de ser **IPC2_Proyecto1_#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.

¹ No debe utilizar estructuras nativas de Python como listas de Python, diccionarios y tuplas.

- 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.
- **NO HABRÁ PRÓRROGA.**

ENTREGA

- La entrega será el **29 de agosto** 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.