PROYECTO 1 LABORATORIO – SISTEMA DE AGRICULTURA DE PRECISION

201906795 – Javier Ricardo Yllescas Barrios

Resumen

El programa presentado propone el desarrollo de un programa que reciba los datos de los campos agricolas con sus estaciones sensores y lecturas en un archivo XML. Y pueda generar un archivo de salida XML y graficar el contenido procesado.

Cabe destacar que el programa no utiliza listas ni arrays de Python, sino que emplea clases propias de listas y matrices, adaptadas para manejar los datos y operaciones necesarias para el agrupamiento y la produccion de los procesos asignados.

Primero obtiene el archivo XML donde se segmenta los datos y se almacenan en sus respectivos nodos y listas.

El programa construye matrices con los datos para luego hacer matrices patrones a partir de las frecuencias registradas, representando de manera estructurada el comportamiento de cada estación. Se procede a agrupar las estaciones que presentan patrones idénticos, con el objetivo de identificar redundancias y similitudes en los datos.

Luego se calcula la matriz reducida de frecuencias, optimizando así el uso de las estaciones base y mejorando la eficiencia del sistema.

Donde pueden expulsar un archivo de salida con todos los datos procesados y generar una grafica para ver las matrices generadas.

Palabras clave

Proyecto1, Listas con apuntadores, Apuntadores, Nodos, USAC.

Introducción

Este programa se centra en la gestión y análisis de datos provenientes de campos agrícolas, a través de la lectura y el procesamiento de información registrada por estaciones y sensores. Su relevancia radica en la necesidad de mejorar el monitoreo de cultivos, detectar patrones en el comportamiento de las estaciones y optimizar la recolección y uso de la información. En un contexto donde la agricultura de precisión cobra cada vez mayor importancia, herramientas como esta permiten automatizar procesos complejos y apoyar la toma de decisiones de manera más eficiente y confiable.

A diferencia de las estructuras tradicionales de Python, como listas o arrays, este programa utiliza clases propias de listas y matrices. Estas estructuras

personalizadas facilitan un manejo más organizado de los datos y permiten realizar operaciones específicas para agrupar información y generar resultados claros.

El sistema comienza con la lectura de un archivo XML que contiene los datos de los campos, estaciones y sensores. La información se segmenta y se almacena en nodos y listas adaptadas, lo que asegura un manejo ordenado y comprensible de los datos. Posteriormente, se construyen matrices a partir de las lecturas y se generan matrices patrón según las frecuencias registradas, reflejando de manera estructurada el comportamiento de cada estación.

Uno de los objetivos principales del programa es agrupar las estaciones que presentan patrones idénticos, con el fin de identificar redundancias y similitudes, optimizando el uso de las estaciones base. A partir de esto, se calcula una matriz reducida de frecuencias que mejora la eficiencia del sistema y facilita la visualización y análisis de los datos. Finalmente, el programa permite generar un archivo XML de salida con toda la información procesada y crear gráficos que representen las matrices, ofreciendo una interpretación clara y práctica de los resultados.

A partir de este enfoque surge la pregunta: ¿de qué manera un sistema automatizado, basado en estructuras de datos propias, puede mejorar la eficiencia y precisión en el análisis de información agrícola? El desarrollo del programa ofrece respuestas a esta interrogante, demostrando cómo organizar, agrupar y visualizar los datos contribuye a optimizar los recursos disponibles en la gestión de campos agrícolas.

Desarrollo del tema

Para una mejor compresion del proyecto se aplicaron los siguientes archivos para desarrollarlo.

Estructura del Proyecto 1 [Archivos]

-main.py

>Menutexto.py

>SistemaCentral.py

>SistemaArchivos.py

>ListaSimple.py

>>CampoAgricola.py

>>Matriz.py

>>NodosSistema.py

Donde:

main.py

Punto de entrada principal, gestiona el menú y el flujo del programa.

Menutexto.py

Define la clase Menu para mostrar menús y manejar entradas de usuario.

SistemaCentral.py

Lógica central del sistema, carga y segmenta datos, gestiona campos agrícolas.

SistemaArchivos.py

Maneja la lectura y escritura de archivos XML (entrada/salida).

CampoAgricola.py

Define la clase CampoAgricola y lógica para matrices y reducción de datos.

ListaSimple.py

Implementa una lista enlazada simple y operaciones básicas sobre ella.

Matriz.py

Implementa la estructura de matriz usando listas simples para almacenar datos.

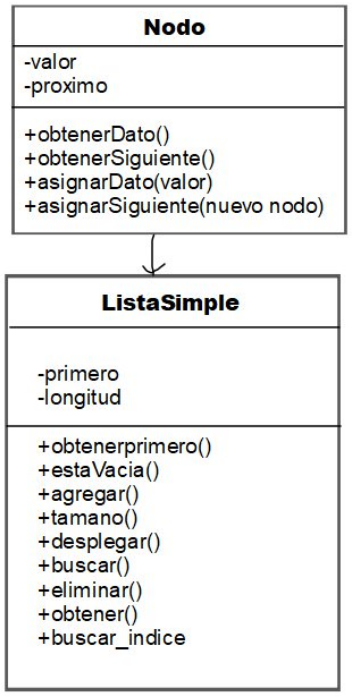
NodosSistema.py

Define nodos y estructuras de datos para todo el proyecto.

Listas

Al desarrollar la practicas se evitaron usar listas y arreglos de python por lo cual se crearon clases especiales para simular listas aplicando los conceptos aprendidos en clase como son los nodos, apuntadore, clases en python y herencia.

Se tuvo que implementar la siguiente estructura para las listas



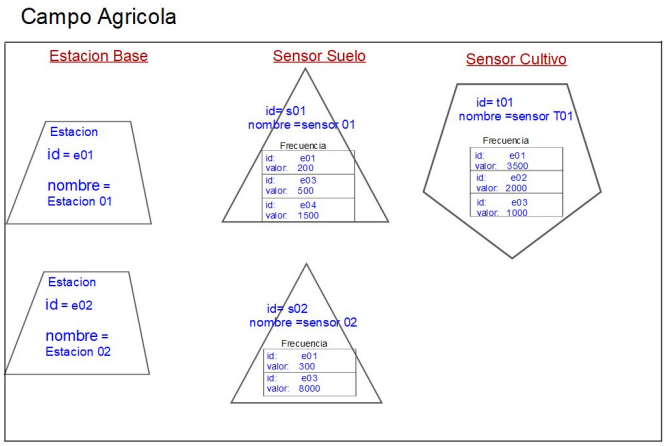
Donde:

Nodo: cada elemnto de la lista es un objeto Nodo, que guarda un valor y una referencia al siguiente Nodo.

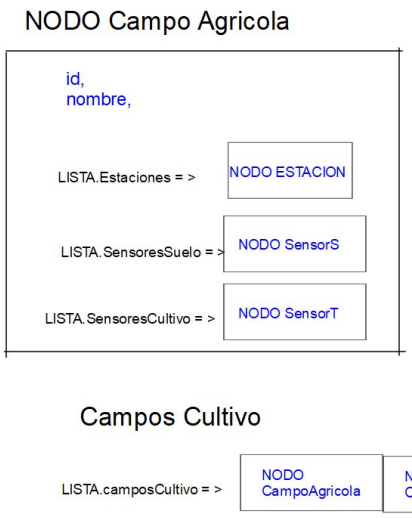
ListaSimple: es la estructura principal que administra los Nodos, almacenandolos en primero. Se va guiando atraves de los punteros de Nodo y las funciones de la misma.

CampoAgricola

Antes de seguir agregando nodos la siguiente imagen es un diagrama a mano alzada para explicar la estructura del nodo mayor que contiene los datos del campo.



Describiendo seria 3 listas principales serian 1) Estaciones base, 2) Sensores Suelo y 3) sensores cultivo, estas listas serian los pilares para desarrollar el proyecto.

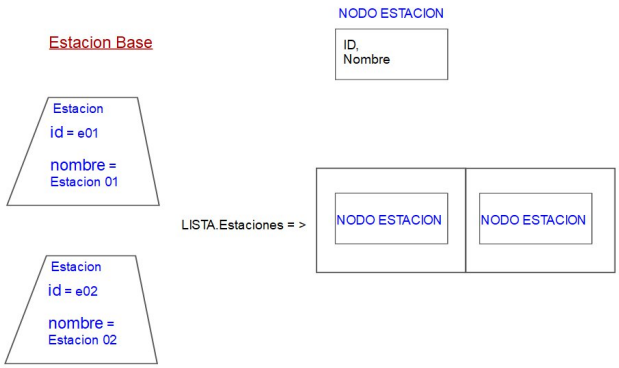


Tomaria el id y el nombre junto con las listas agrupadas en un nodo respectivamente para cada lista.

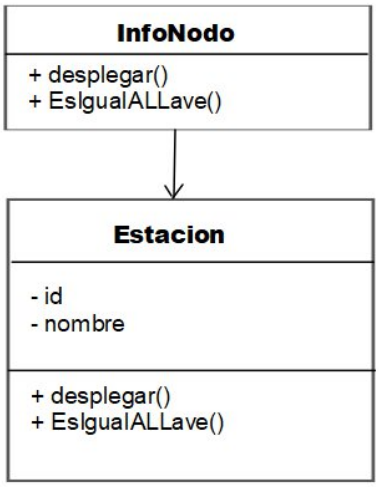
Luego serian agrupado en en una lista campos cultivo donde estarian los nodos campo agricola

Lista Estaciones base

La estructura general para la lista estaciones base seria.

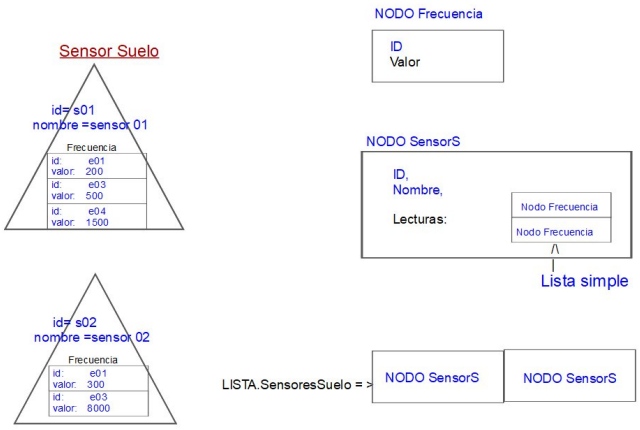


Contaria con nodo estacion con id y nombre de la estacion

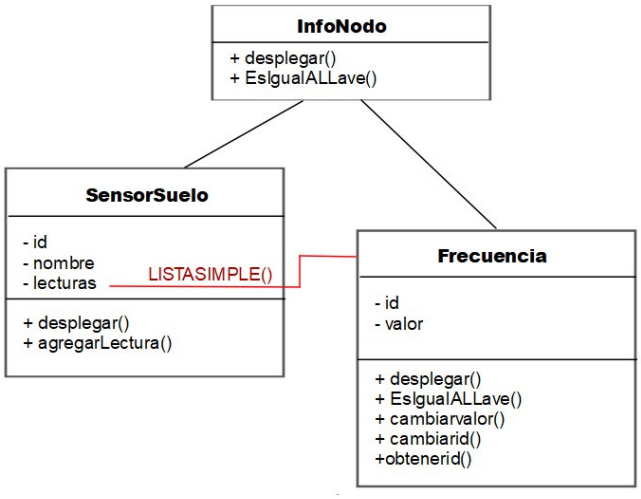


Lista Sensores Suelo

La estructura general para la lista sensorse suelo.

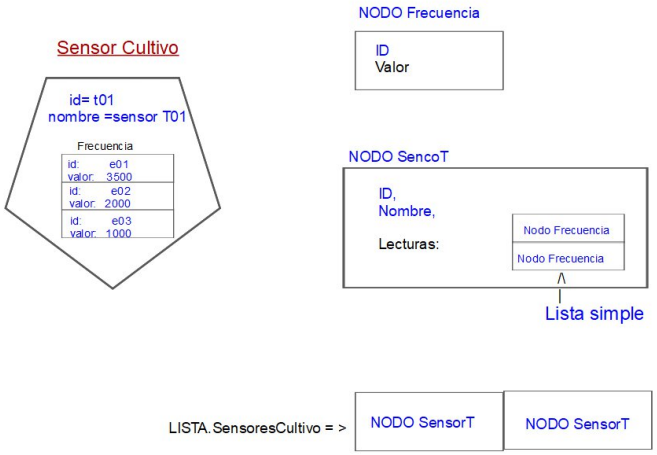


Cuenta con un nodo de frecuencias que se almacena dentro de una lista de lectura para luego agregarla al nodo SensorS y pasarlo a la Lista sensores suelo.

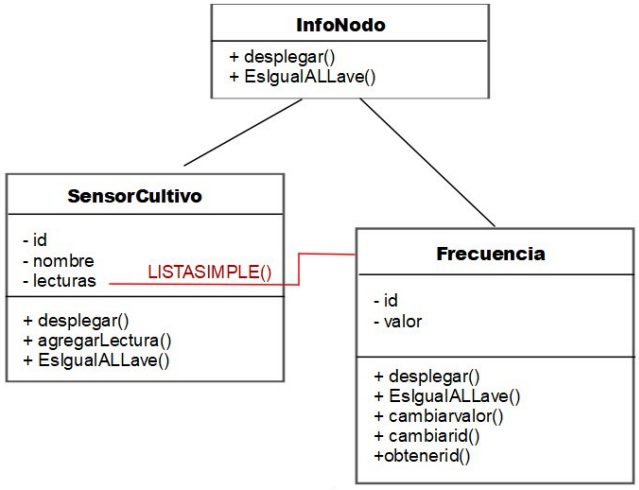


Lista Cultivo

Estructura general para la lista sensor cultivo.

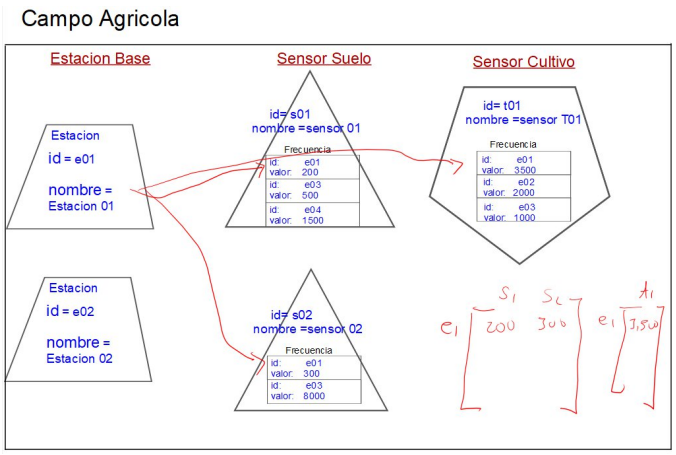


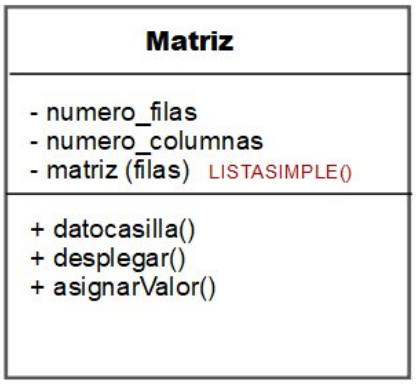
Su nodo primario es el de frecuencias, tambien se almacenan en lecturas y luego en un nodo sensor cultivo para guardarlas en una lista sensores cultivos.

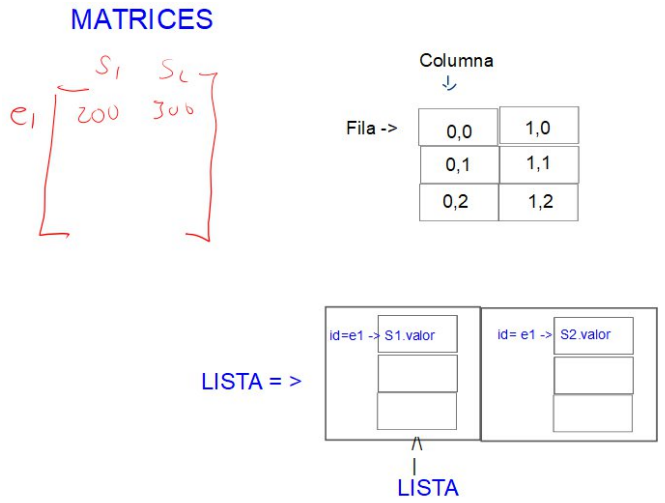


Matrices

Para el agrupamiento de los datos en matrices se aplico el siguiente diagrama



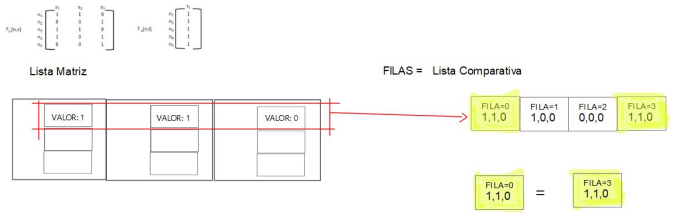




Se crearon las matrices a partir de dos listas la primera almacena las filas y luego se agrega a otra lista para que cuenten como columnas.

Se crearon clases matrices para acceder cada valor como si fuera un excel dando su ubicacion numero de columna y numero de fila y obtener sus valores ubicados en esa casilla especifica.

Construcción matrices reducidas



Se utilizaron listas auxiliares para alamcenar temporalmente los datos agrupados.

Primero se obtuvo los primeros valores de cada columna para luego concatenar sus valores en un texto para luego realizar asi con todas la filas.

Luego la lista auxiliar se compara cada nodo si los textos eran iguales entonces las filas eran iguales.

Como ejecutar el Proyecto 1.

Para ejecutar el proyecto necesitas tener python instalado y saber usar CMD

Paso 1: Busca la carpeta donde este el documento main.py

Paso 2: Ejecuta CMD apartir de esa ruta del sistema

Paso 3: Colocar en las lineas de comando el siguiente ocmando "**python main.py**" luego presiona Enter.

Veras una ventana parecida a esta

```
Proyecto IPC2
Javier Ricardo Yllescas Barrios
201906795
************************************
*         Sistema DataAgro         *
************************************
* 1. Cargar archivo                *
* 2. Procesar archivo              *
* 3. Escribir archivos salida      *
* 4. Mostrar datos del estudiante  *
* 5. Generar gráfica               *
* 6. Salida                        *
************************************

Ingrese una opción:
```

Paso 4:Ingres la opcion deseada del 1 al 6.

Cargar archivo

```
Ingrese una opción: 1
>> Menu Cargar un archivo.

************************************
*         Cargar archivo           *
************************************
Ingrese la ruta del archivo:
Ingrese el nombre del archivo:
```

Apareceran dos opciones una ingresando la ruta donde esta el archivo y otra con el nombre de archivo incluyendo la extension xml ejemplo(archivo.xml)

Procesar archivos

```
- Frecuencia>> ID: 0,0  |  Valor: 0
- Frecuencia>> ID: 0  |  Valor: 0
Columna 2 ----------
- Frecuencia>> ID: e01,e03  |  Valor: 700
- Frecuencia>> ID: 0,0  |  Valor: 0
- Frecuencia>> ID: e04  |  Valor: 1500
----------[Fin Matriz]----------

>>>> Matriz Suelo REDUCIDA CULTIVO
----------[Incio Matriz]----------
Columna 0 ----------
- Frecuencia>> ID: e01,e01  |  Valor: 4500
- Frecuencia>> ID: e02,e02  |  Valor: 5200
- Frecuencia>> ID: e04  |  Valor: 950
----------[Fin Matriz]----------
################################################


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

Ingrese una opción:
```

Muestra en la consola todos los procesos para generar las listas y matrices.

Escribir archivos salida

```
Ingrese una opción: 3
>> Menu Escribir archivo de salida.

************************************
*         Escribir archivo         *
************************************
Ingrese la ruta del archivo:
Ingrese el nombre del archivo:
```

Pedira la ruta y nombre del archivo (coloca la extension xml al archivo) luego generar un archivo con los datos procesados.

Mostrar datos del estudiante

Opcion 4 muestra los datos del estudiante en las consola

```
Ingrese una opción: 4
>> Menu Mostrar datos del estudiante

////////////////////////////////
> Nombre: Javier Ricardo Yllescas Barrios
> Carnet: 201906795
> Introduccion a la Programación y Computación 2
> Sección: C
> 4to. Semestre
> Github: https://github.com/Javier-201906795/IPC2_Proyecto1_201906795
> Documentacion:
> https://github.com/Javier-201906795/IPC2_Proyecto1_201906795/blob/mair
////////////////////////////////
```

Generar grafica

Opcion 5 genera un grafica

Para que se genera la grafica es necesario tener instalado **Graphviz**

```
************************************
*         Sistema DataAgro         *
************************************
* 1. Cargar archivo                *
* 2. Procesar archivo              *
* 3. Escribir archivos salida      *
* 4. Mostrar datos del estudiante  *
* 5. Generar gráfica               *
* 6. Salida                        *
************************************

Ingrese una opción: 5
```

Y la ultima opcion es la 6 que finaliza el programa.

Conclusiones

Las listas enlazadas son una alternativa a los arrays, con mucha mas facilidad y un manejo de la informacion con un seguimiento constante de la ubicacion de los datos.

Al crear clase nodos se pueden almacenar y procesar los datos de una manera mas ordenada eficiente proque se pueden evitar ciclos repetitivos sin llamar a funciones externas.

La aplicacion de los conceptos de listas y nodos resulto muy enriquecedor, pero su nivel de complegidad va aumentando a la hora de almacenar muchos nodos dentro de listas, dentro de otras listas, llevando un acceso mas lento a la informacion y una complejidad para procesar los datos.

Comentarios:

Al desarrollar el programa sin el uso de listas y aplicando la tecnica de apuntadores resulto un reto a al hora de mantener la logica en que nodo esta ubicado, ya que despues de almacenar los datos en varias listas con sus nodos respectivos y la opciones de obtener siguiente no daba mucha informacion solo a la hora de depurar el codigo se podia saber en que nodo se encontraba manejando.

Referencias bibliográficas

Universidad de San Carlos de Guatemala. (s.f.). Enunciado proyecto 1. https://uedi.ingenieria.usac.edu.gt/campus/pluginfile.php/258349/mod_resource/content/1/%5BIPC2%5DProyecto202502_v2.pdf

Argueta, Hesban. (s.f.). *Clases laboratorio IPC2* [Repositorio en GitHub]. GitHub. https://github.com/Hes-007/IPC2-2S2025/tree/main.

Ruiz Juarez, J. M. (s.f.). Contenido Unidad 1. https://uedi.ingenieria.usac.edu.gt/campus/course/view.php?id=2547

Graphviz. (s.f.). Documentacion para crear graficas. https://graphviz.org/documentation