
SISTEMA DE AGRICULTURA DE PRECISIÓN CON POO, TDA Y GRAPHVIZ PARA AGRUPAMIENTO DE ESTACIONES

202405828 – Danny Josué González Lémus

Resumen

Este ensayo presenta el diseño e implementación de un sistema de agricultura de precisión que procesa datos de estaciones base, sensores de suelo y de cultivo para construir matrices de frecuencias $F[n,s]$ y $F[n,t]$, derivar patrones binarios F_p y agrupar estaciones con igual comportamiento simultáneo. La solución se implementó en Python bajo POO, utilizando Tipos de Datos Abstractos propios (Nodo y Lista enlazada) para modelar todas las colecciones, sin recurrir a estructuras nativas. El procesamiento se alimenta de archivos XML con la estructura exigida por el enunciado, y la salida produce un XML reducido cuyas estaciones corresponden a grupos con el mismo patrón; las frecuencias se re-suman por sensor. La visualización se realiza con Graphviz para F , F_p y Fr . La propuesta enfatiza la correcta separación de responsabilidades (parsing, modelado, matrices, agrupamiento, escritura y visualización), el cumplimiento estricto del formato de entrada/salida y la trazabilidad del proceso por medio de un menú de consola con seis opciones.

Palabras clave

POO, TDA, XML, Graphviz, Agricultura de precisión

Abstract

This essay presents the design and implementation of a precision agriculture system that processes data from base stations and soil/crop sensors to build frequency matrices $F[n,s]$ and $F[n,t]$, derive binary patterns F_p , and cluster stations that share the same simultaneous behavior. The solution is implemented in Python using OOP and custom Abstract Data Types (Node and linked List) to model all collections, avoiding native structures. Input files follow the required XML schema and output is a reduced XML whose stations correspond to groups with the same pattern; frequencies are re-summed per sensor. Visualization with Graphviz covers F , F_p and Fr . The proposal highlights clear separation of concerns (parsing, modeling, matrices, clustering, writing and visualization), strict compliance with I/O formats, and traceable processing through a six-option console menu.

Keywords

OOP, ADT, XML, Graphviz, Precision agriculture

Introducción

La agricultura de precisión demanda integrar datos heterogéneos con garantías de trazabilidad. Este trabajo aborda la construcción de matrices de frecuencia a partir de insumos XML, su binarización en patrones Fp y el agrupamiento de estaciones que comparten comportamiento en suelo y cultivo. El enfoque combina POO con TDA propios (Nodo y Lista) para asegurar control total sobre la gestión de colecciones, y Graphviz para la visualización del proceso (F, Fp y Fr). El propósito es evidenciar una arquitectura modular: dominio, matrices, agrupador y E/S XML, más una CLI de seis opciones que guía al usuario. Se destacan decisiones de diseño que fomentan mantenibilidad y evaluación objetiva.

Desarrollo del tema

- a) Diseño POO y TDA. Se definieron entidades de dominio (CampoAgricola, EstacionBase, sensores y Frecuencia) y TDA (Nodo, Lista) para toda colección. Las matrices heredan de una base común que permite acceso indexado sobre listas enlazadas. El encapsulamiento y la separación de capas aseguran claridad: io (Parser/Writer XML), core (matrices y agrupador), viz (Graphviz) y cli.
- b) Construcción de F y Fp. A partir de sensores de suelo y cultivo se construyen F[n,s] y F[n,t]; la binarización define Fp como 1 si la frecuencia es mayor que cero y 0 en otro caso. Esta decisión uniforma el criterio de presencia/ausencia sin afectar la reducción por suma posterior.
- c) Agrupamiento por patrón conjunto. Dos estaciones se agrupan solo si coinciden sus filas en Fp_suelo y Fp_cultivo. Se recorre la matriz por filas, marcando estaciones asignadas y creando grupos disjuntos. Este criterio evita falsos positivos por coincidencias parciales.
- d) Reducción Fr y salida XML. Para cada grupo, se suman las filas de F en suelo y cultivo, generando Fr.

Las estaciones reducidas se renombran tomando el id del primer miembro y concatenando nombres; se reescriben frecuencias por sensor, respetando el esquema de salida. Graphviz genera vistas F/Fp/Fr para validación visual.

Tabla I.

Variables principales del sistema

CATEGORÍA	CATEGORÍA
VARIABLE	DESCRIPCIÓN
F[n,s]	Frecuencias suelo por estación × sensor
F[n,t]	Frecuencias cultivo por estación × sensor
Fp	Binarización de F (>0 = 1; 0 = 0)
Fr	Suma por grupos (suelo y cultivo)

Figura 1. Esquema del flujo F → Fp → agrupamiento → Fr.
Fuente: elaboración propia.

Conclusiones

1) Arquitectónica: La combinación de POO y TDA propios (Nodo/Lista) permitió cumplir la restricción de no usar colecciones nativas, manteniendo cohesión por capas (io, core, viz, cli) y favoreciendo pruebas y mantenimiento.

2) Algorítmica: El criterio de patrones simultáneos en suelo y cultivo (Fp_s, Fp_t) reduce ruido y evita agrupamientos espurios; el pipeline $F \rightarrow Fp \rightarrow \text{agrupamiento} \rightarrow Fr$ garantiza determinismo y trazabilidad en la reducción.

3) Operativa: El menú de seis opciones, la E/S XML estricta y las gráficas con Graphviz entregan un flujo reproducible y auditable en laboratorio; el diseño admite extensiones (nuevas estrategias de agrupamiento) sin impactar el resto del sistema

Referencias bibliográficas

1. Date, C. J. (1991). An Introduction to Database Systems. Addison-Wesley.
2. Ellson, J., Gansner, E. R., Koutsofios, E., North, S. C., & Woodhull, G. (2004). Graphviz and Dynagraph—Static and Dynamic Graph Drawing. Graph Drawing Software, Springer.
3. Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). Design Patterns. Addison-Wesley.
4. Python Software Foundation. (2024). xml.etree.ElementTree — The ElementTree XML API. <https://docs.python.org/>
5. Rossum, G. van, & Drake, F. L. (2024). The Python Language Reference. Python Software Foundation.