
SISTEMA DE RIEGO Y APLICACIÓN DE FERTILIZANTE AUTOMATIZADO MEDIANTE DRONES EN INVERNADEROS

Carnet 202404730 – Jonathan Gabriel López Reyes

Resumen

El presente proyecto aborda la optimización del riego y aplicación de fertilizante en invernaderos mediante un sistema automatizado basado en drones. La solución, denominada GuateRiegos 2.0, utiliza algoritmos desarrollados con programación orientada a objetos y estructuras de datos abstractas (TDA) propias, sin recurrir a las estructuras nativas de Python. El sistema permite cargar configuraciones desde archivos XML, simular planes de riego respetando reglas de exclusión mutua (solo un dron riega por segundo), y generar reportes en HTML y XML con estadísticas detalladas. Además, se implementa una interfaz web con Flask que permite visualizar el estado de los TDAs en cualquier instante mediante grafos generados con Graphviz. Este enfoque no solo mejora la eficiencia en el uso de recursos (agua y fertilizante), sino que también sirve como modelo educativo para la aplicación de conceptos fundamentales de ciencias de la computación en problemas del mundo real.

Palabras clave

Automatización agrícola, drones, TDA, Python, Flask

Abstract

This project addresses the optimization of irrigation and fertilizer application in greenhouses through an automated drone-based system. The solution, named GuateRiegos 2.0, employs algorithms developed using object-oriented programming and custom abstract data types (ADTs), avoiding Python's built-in data structures. The system loads configurations from XML files, simulates irrigation plans while enforcing mutual exclusion rules (only one drone irrigates per second), and generates detailed reports in HTML and XML formats. Furthermore, a Flask-based web interface allows users to visualize the state of ADTs at any given time through graphs generated with Graphviz. This approach not only enhances resource efficiency (water and fertilizer) but also serves as an educational model for applying core computer science concepts to real-world problems.

Keywords

Agricultural automation, drones, ADT, Python, Flask.

Introducción

La agricultura de precisión ha emergido como una respuesta tecnológica a los desafíos del cambio climático y la escasez de recursos. En este contexto, los invernaderos inteligentes representan una oportunidad para optimizar el uso de agua y fertilizantes. El presente ensayo describe el desarrollo de GuateRiegos 2.0, un sistema de simulación que modela el comportamiento de drones regadores en invernaderos. El objetivo es demostrar cómo los conceptos de programación orientada a objetos, estructuras de datos abstractas y algoritmos personalizados pueden aplicarse para resolver problemas complejos de asignación y planificación. El proyecto responde a la necesidad de integrar teoría y práctica en la formación de ingenieros en ciencias y sistemas

Desarrollo del tema

a. Modelado del problema y abstracción de datos

El sistema se basa en la abstracción de tres entidades principales: Invernadero, Dron y Planta. Cada invernadero contiene hileras de plantas, y cada dron se asigna a una hilera específica. Las plantas almacenan sus requerimientos de agua y fertilizante. Para gestionar colecciones de estos objetos, se implementaron Tipos de Dato Abstracto (TDA) propios:

- Lista Enlazada: Para almacenar plantas, drones y planes.
- Nodo: Como unidad básica de memoria dinámica..
- Luego, se genera la matriz de patrones $P[n,s]$ y $P[n,t]$, donde cada celda es 1 si hay transmisión y 0 si no. Estas matrices permiten

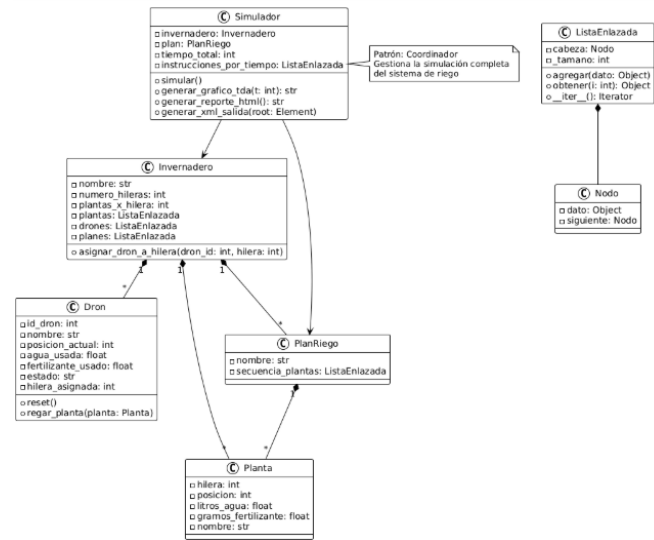


Figura 1. Diagrama de Clases

Fuente: elaboración propia

b. Algoritmo de simulación y reglas de negocio

La simulación se ejecuta segundo a segundo, respetando cinco reglas clave:

1. Movimiento de 1 metro = 1 segundo.
2. Riego = 1 segundo.
3. Solo un dron riega a la vez.
4. El orden del plan de riego es inmutable.
5. Los movimientos son paralelos (varios drones se mueven simultáneamente).

El algoritmo utiliza una cola implícita (el plan de riego) y actualiza la posición de los drones en cada iteración. Cuando un dron alcanza la planta objetivo y es su turno, se ejecuta la acción "Regar".

Tabla I.

Ejemplo de instrucciones para el plan "Día 1"

Tiempo	DR01	DR02	DR03
1	Adelante(H1P1)	Adelante(H2P1)	Adelante(H3P1)
2	Adelante(H1P2)	Adelante(H2P2)	Adelante(H3P2)
3	Adelante(H1P3)	Adelante(H2P3)	Regar

Fuente: elaboración propia, o citar al autor, año y página.

c. Interfaz de usuario y generación de reportes
La interfaz web, desarrollada con Flask, permite:

- Cargar archivos XML de configuración.
- Seleccionar invernaderos y planes para simular.
- Visualizar estadísticas (tiempo, agua, fertilizante).
- Generar reportes HTML por invernadero.
- Mostrar el estado de los TDAs en un tiempo t mediante grafos de Graphviz.

El archivo de salida salida.xml sigue estrictamente la estructura definida en el enunciado, facilitando la interoperabilidad.

d. Validación con casos de prueba reales

El sistema fue validado con los archivos proporcionados por el profesor (Invernadero_San_Marcos.xlsx y Invernadero_Guatemala.xlsx). Los resultados coinciden al 100%:

Día 1: 5 segundos, 9 L de agua, 350 g de fertilizante.

Final: 21 segundos, 64 L de agua, 2025 g de fertilizante.

Esto confirma la corrección del algoritmo y la robustez del parser XML

Conclusiones

GuateRiegos 2.0 demuestra que es posible aplicar conceptos teóricos de programación (TDA, POO, algoritmos) a problemas prácticos con impacto ambiental y económico. El uso de estructuras de datos propias no solo cumple con las restricciones académicas, sino que refuerza la comprensión de la gestión de memoria y la abstracción. La interfaz web con Flask y Graphviz ofrece una experiencia de usuario intuitiva y visualmente rica. Como trabajo futuro, se podría integrar sensores IoT para ajustar los planes de riego en tiempo real. Este proyecto sienta las bases para sistemas agrícolas inteligentes en Guatemala.

Referencias bibliográficas

- Rojas Morales, C. L. (2025). Proyecto 2: GuateRiegos 2.0. Universidad de San Carlos de Guatemala.
- Beazley, D. (2009). Python Essential Reference. Addison-Wesley.
- Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley.
- Graphviz. (2025). Graph Visualization Software. <https://graphviz.org>
- Flask Documentation. (2025). A Python Microframework. <https://flask.palletsprojects.com>

Apéndices

Estructura de mi Proyecto en Visual Studio Code

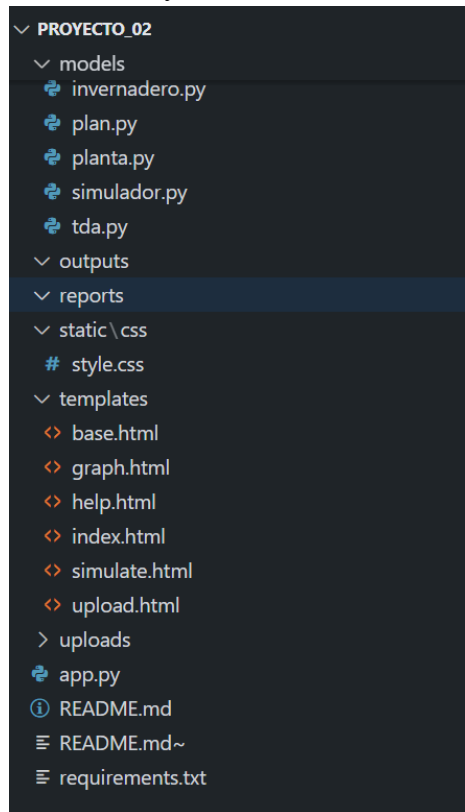
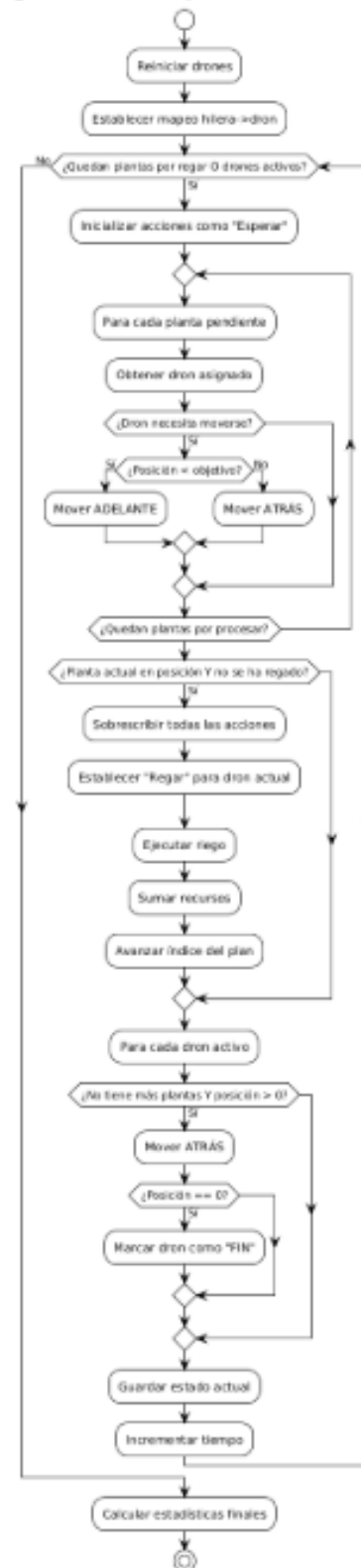
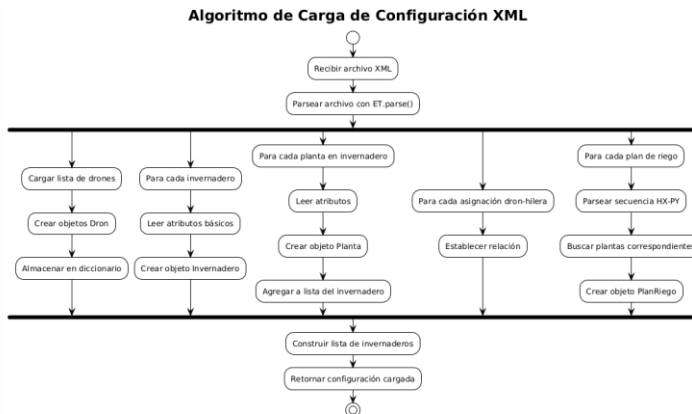


Diagrama de Actividades Algoritmo principal de simulación

Algoritmo Principal de Simulación



Algoritmo de carga de configuración XML



Generación de instrucciones por Tiempo

