
GUATERIEGOS 2.0: SIMULACIÓN DE RIEGO AUTOMATIZADO CON DRONES Y TDAs PROPIOS

201503980 – Gerson Ottoniel Batz Cocón

Resumen

El presente ensayo describe el desarrollo del proyecto GuateRiegos 2.0, un sistema que simula el riego automatizado en invernaderos utilizando drones asignados a hileras específicas. El objetivo es optimizar el uso de agua y fertilizante mediante un motor de simulación que respeta restricciones de tiempo y orden en los planes de riego. La propuesta emplea estructuras de datos propias (listas ligadas y colas), evitando el uso de listas o diccionarios nativos de Python, lo que refuerza la comprensión de la memoria dinámica. Se diseñó una interfaz web en Flask para cargar configuraciones en XML, ejecutar simulaciones, generar reportes en HTML y exportar resultados a XML. El impacto de esta solución se proyecta en el ámbito académico, técnico y ambiental, pues permite comprender la importancia de la programación estructurada y al mismo tiempo simula procesos agrícolas eficientes que pueden reducir desperdicios de recursos. Las conclusiones destacan el cumplimiento de los requisitos técnicos, el aprendizaje adquirido en el uso de TDAs, así como la aplicabilidad del modelo en contextos reales de agricultura de precisión.

Palabras clave

Simulación, drones, riego, TDAs, Flask

Abstract

This essay describes the development of the GuateRiegos 2.0 project, a system that simulates automated irrigation in greenhouses using drones assigned to specific rows. The goal is to optimize water and fertilizer usage through a simulation engine that respects time and order restrictions defined in irrigation plans. The proposal uses custom data structures (linked lists and queues), avoiding the use of Python's native lists or dictionaries, thus reinforcing the understanding of dynamic memory. A Flask web interface was designed to load XML configurations, run simulations, generate HTML reports, and export results to XML. The impact of this solution extends to academic, technical, and environmental domains, as it highlights the relevance of structured programming while simulating efficient agricultural processes that may reduce resource waste. The conclusions emphasize technical requirements fulfillment, learning in the use of custom TDAs, and the applicability of the model in real precision agriculture contexts.

Keywords

Simulation, drones, irrigation, TDAs, Flask

Introducción

La simulación de procesos agrícolas mediante drones constituye un ejemplo actual de cómo la tecnología puede aplicarse para optimizar recursos en la producción. En el marco del curso IPC2, se desarrolló GuateRiegos 2.0, un sistema orientado a objetos que implementa estructuras de datos dinámicas para representar invernaderos, plantas y drones. El propósito principal es mostrar el potencial de los TDAs en la solución de problemas reales, además de cumplir con los requisitos técnicos del curso. El ensayo presenta los fundamentos, el diseño del software, su motor de simulación y la relevancia de esta experiencia como aprendizaje académico y práctico.

Desarrollo del tema

a. Modelado de entidades y uso de TDAs

Se diseñaron clases que representan las entidades principales: Sistema, Invernadero, Hilera, Planta, Dron, Plan y PasoPlan. Para administrar colecciones, se implementaron TDAs propios: ListaSimple y Cola. Esto permitió controlar directamente la memoria dinámica y reforzar el aprendizaje de estructuras enlazadas. La decisión de no utilizar listas o diccionarios nativos obliga al estudiante a comprender en detalle cómo gestionar referencias y recorridos.

b. Motor de simulación

El motor implementado simula segundo a segundo el avance de los drones hacia sus objetivos. Cada dron se mueve a razón de un metro por segundo y riega durante un segundo exacto al llegar a su posición objetivo. La simulación garantiza que únicamente se realice un riego simultáneo por invernadero, mientras otros drones pueden seguir desplazándose. El resultado incluye el tiempo

óptimo, litros de agua, gramos de fertilizante consumidos y la eficiencia por dron.

c. Interfaz web y reportes

Se utilizó Flask como framework para ofrecer una interfaz web sencilla que permite cargar archivos XML de configuración, seleccionar invernaderos y planes de riego, ejecutar simulaciones y visualizar resultados. Los reportes se generan en HTML y muestran las instrucciones paso a paso, además de estadísticas globales y por dron. También se genera automáticamente un archivo salida.xml con la información solicitada en el enunciado.

d. Diagramas de diseño

El diseño incluye un diagrama de clases que muestra las relaciones entre las entidades y un diagrama de actividades que representa el algoritmo de simulación. Estos modelos fueron fundamentales para estructurar la solución antes de implementarla en código.

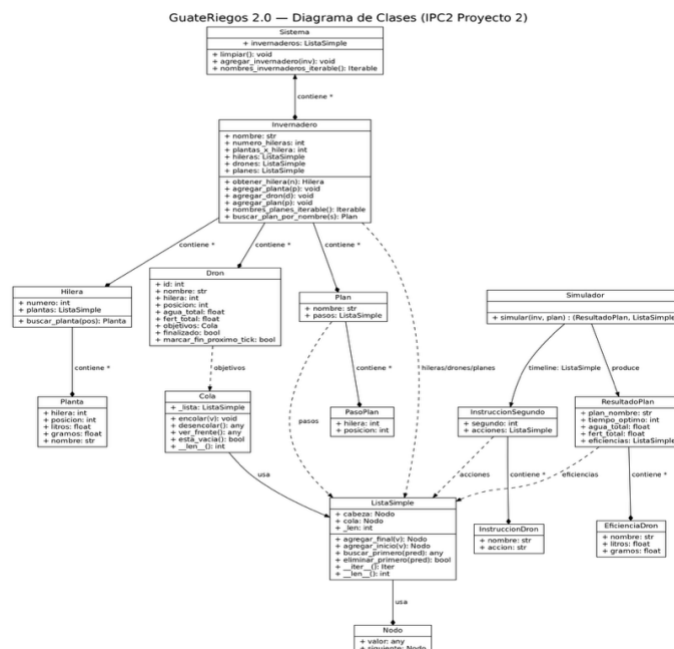


Figura 1. Diagrama de clases.

Fuente: elaboración propia.

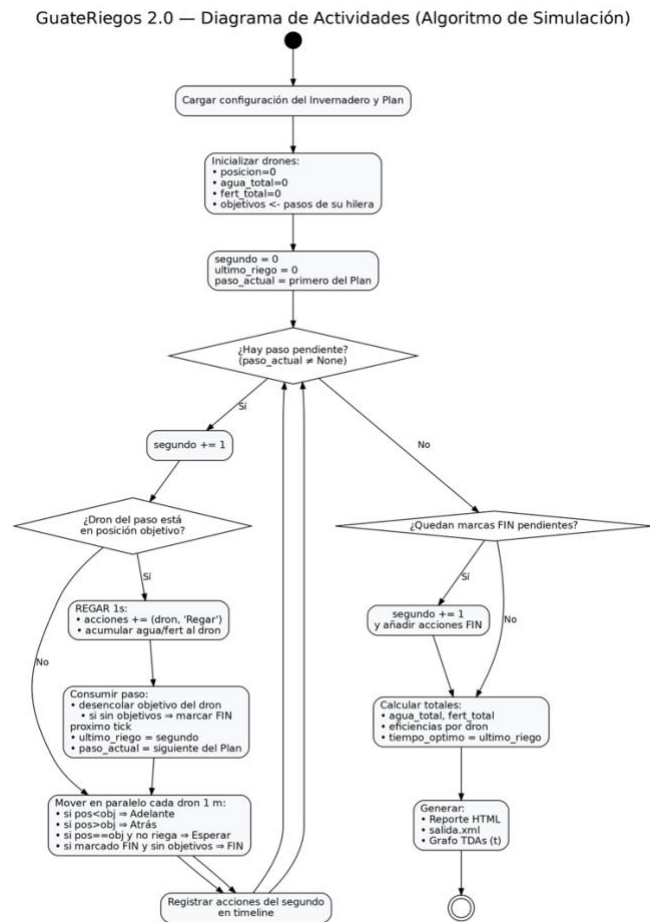


Figura 2. Diagrama de actividades

Fuente: elaboración propia.

e. Validación del sistema

Una parte esencial del proyecto fue validar que el motor de simulación funcionara correctamente bajo diferentes escenarios. Para ello se elaboraron casos de prueba con distintos números de drones, hileras y planes de riego. En cada ejecución se verificó que los drones respetaran las restricciones: movimiento de un metro por segundo, riego exclusivo por un solo dron en cada instante y avance paralelo de los demás. Los resultados obtenidos fueron consistentes con lo planteado en el enunciado, garantizando tiempos óptimos de riego y cálculos precisos de agua y

fertilizante. Esta fase de validación permitió detectar errores tempranos y ajustar la lógica del simulador, reforzando la importancia de aplicar pruebas incrementales en proyectos de ingeniería de software.

f. Integración de reportes y salida XML

Otro aspecto relevante fue la generación de reportes claros y útiles. El reporte en HTML no solo muestra las instrucciones segundo a segundo, sino que también permite un análisis visual de las acciones ejecutadas por los drones. A nivel técnico, se aplicó Jinja2 para renderizar las tablas de datos, lo que facilita su lectura y mejora la experiencia del usuario. Por su parte, el archivo salida.xml consolida la información de cada simulación en un formato estructurado, reutilizable y transportable, lo cual refleja buenas prácticas en la interoperabilidad de sistemas.

g. Impacto académico y profesional

Este proyecto va más allá de cumplir con un requerimiento académico: fomenta competencias prácticas alineadas con el ejercicio profesional de un ingeniero en sistemas. El uso de TDAs propios fortalece el dominio de estructuras de datos, la implementación con Flask desarrolla habilidades en aplicaciones web, y la aplicación de Graphviz introduce al estudiante en la visualización técnica de modelos. En conjunto, estos elementos forman una experiencia integral que conecta la teoría con la práctica, contribuyendo a la formación de profesionales capaces de abordar problemas reales con soluciones técnicas robustas.

Conclusiones

El proyecto GuateRiegos 2.0 permitió aplicar conceptos de programación orientada a objetos y estructuras de datos dinámicas en un problema realista. Se evidenció que el uso de TDAs propios brinda un mayor entendimiento del manejo de memoria, mientras que la simulación de riego con drones ilustra aplicaciones prácticas en agricultura de precisión. La interfaz web y los reportes refuerzan la usabilidad y la presentación de resultados. Como conclusión, se destaca la importancia de combinar la teoría de estructuras con proyectos aplicados, generando competencias técnicas y visión de impacto ambiental y social.

Referencias bibliográficas

C. J. Date, (1991). *An introduction to Database Systems*. Addison-Wesley Publishing Company, Inc.

C. J. Date (1991). *An introduction to Database Systems*. Addison-Wesley Publishing Company.

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). *Design Patterns*. Addison-Wesley.

Flask Documentation. (2025).
<https://flask.palletsprojects.com>

Graphviz Documentation. (2025).
<https://graphviz.org>

Python Software Foundation. (2025). Python 3.10 Documentation. <https://docs.python.org/3/>