

---

## SOLUCIÓN CAMINO ÓPTIMO R2E2, MEDIANTE EL USO DE TDA Y GRAFICACIÓN DE LOS MAPAS CON GRAPHVIZ

---

202004750 – Pedro Alejandro Zetino Páez

### Resumen

Se presenta la solución de un algoritmo capaz de desplazarse por un camino con el menor gasto de combustible, esto fue posible leyendo el archivo mandado por el satélite en formato xml, este archivo permitió la lectura del terreno y posterior graficación en consola del camino correcto. Esto se programó en Python con el uso de distintas librerías y TDAs. La aplicación se maneja en consola, presentando un menú en el cual seleccionamos la acción a realizar, se leerán todos los terrenos registrados por el satélite. Una vez estos han sido cargados se puede generar un gráfico del terreno, esta será creada en formato png. La app fue realizada con POO y manejo de TDAs, esta solución permitió la lectura de la ruta más eficiente, graficación del terreno, selección del terreno a trabajar y la exportación de un archivo de salida en formato xml, que presenta el recorrido ideal para cada terreno.

### Palabras clave

- *TDA*
- *XML*
- *Python*

### Abstract

*We present the solution in an algorithm that can travel through the path with less use of fuel, this was possible with the help of a satellite that sends an archive xml that allowed the lecture of the terrain and plotting in console of it, with the right path. This was programmed in python, with the use of a few libraries and ADTs. The app is deployed in console, showing a menu where you select the function to do, it can read all the terrains registered by the satellite. Once they have been uploaded a graph of the land will be created in a png image. The app was made with OOP and the use of ADTs, this solution enabled the lecture of the most efficient path, plotting and graphing of the land, selection of the sale and the exportation of an xml type archive that shows the most efficient path for each terrain.*

### Keywords

- *ADT*
- *XML*
- *Python*

## Introducción

Se planteará en este informe la solución al problema planteado en el proyecto, este fue resuelto mediante el uso del lenguaje de programación Python, mediante el uso de programación orientada a objetos y manejo de datos con TDAs (Tipos de datos abstractos).

Con el uso del paradigma de programación POO se crearon varias clases para el manejo de los datos y el almacenamiento de estos, permitiendo esto un uso más eficiente de los objetos y un código limpio, con funciones específicas dependiendo de la clase.

Las clases implementadas fueron el main, un nodo para el manejo del terreno, otro para el manejo de las posiciones y dos listas, una enlazada simple y una doble, siendo estas las que manejaron los datos.

## Desarrollo del tema

El problema que se nos ha presentado es el siguiente:

“La Agencia Guatemalteca de Investigación Espacial (AGIE) ha diseñado un nuevo robot de exploración, llamado r2e2, que tiene la habilidad de explorar nuevos terrenos, este nuevo robot puede moverse en todo tipo de terrenos, sólo necesita más combustible para moverse en terrenos accidentados, y menos combustible en terrenos planos. El único problema con r2e2 es que solo puede moverse ortogonalmente, es decir, únicamente puede moverse en dirección Norte, Este, Sur y Oeste de su posición. R2e2 ha sido diseñado para comunicarse con el satélite Quetzal01 para tener una imagen del terreno que va a explorar, así que debe seleccionar la mejor ruta para llegar al punto de destino, r2e2 siempre debe seleccionar el camino que necesita el mínimo uso de combustible para completar su exploración, sin embargo, los científicos que están experimentando con r2e2

necesitan un programa que calcule el camino que utilice la mínima cantidad de combustible. La máxima cantidad de combustible que r2e2 puede almacenar son 9,999 unidades.”

La solución a esta problemática fue planteada en el lenguaje Python, en el cual con el uso de la librería “*Element tree*” comenzamos la lectura del archivo xml ingresado, mediante ciclos for. La información se almacenó en dos listas enlazadas, la primera fue de tipo simple y se guardó lo siguiente:

- Nombre del terreno.
- Dimensión x.
- Dimensión y.
- Posición inicial en x.
- Posición inicial en y.
- Posición final en x.
- Posición final en y.

Mientras que la segunda lista fue doble, teniendo como contenido:

- Posición en x.
- Posición en y.
- Gasto de combustible en la posición x,y.

El guardado de la información en estas listas fue posible gracias al enfoque de la programación orientada a objetos, con la cual creamos varias clases con las cuales fue más fácil la manipulación de los datos. Entre las clases creadas podemos contar:

- Nodo. (Conteniendo posiciones x,y combustible)
- Nodoterreno.
- Simple.
- Doble.
- Main. (Esta clase contiene el menú)

El menú del programa fue realizado para ser operando en consola, con una interfaz amigable

que nos permite navegar entre las distintas opciones, en caso de error seremos regresados al menú.

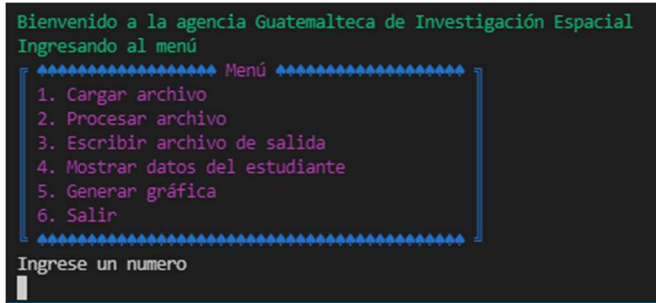


Figura 1. Menú principal

Fuente: Elaboración propia.

El archivo en formato xml fue leído mediante varios ciclos for anidados, guardando estos los valores de los distintos atributos en variables y al momento de terminar la iteración general, mediante un método estos eran mandados a ambas listas.



Figura 2. Ciclos de lectura al archivo.

Fuente: Elaboración propia.

Los datos recibidos nos permitían utilizar un algoritmo con el cual se determinó el mejor camino para ahorrar combustible.

Para la graficación del terreno el programa se apoyó en la aplicación graphviz, teniendo esta que ser instalada en el sistema y posteriormente agregarla a la variable de entorno "PATH", para esto permitir que fuera ejecutada desde Python usando el interprete de comandos con el método "os.system", el archivo creado posee la extensión ".dot", este es el que procesa graphviz, creando una imagen png, la cual es luego abierta por el método "os.startfile" en un programa capaz de mostrar imágenes.

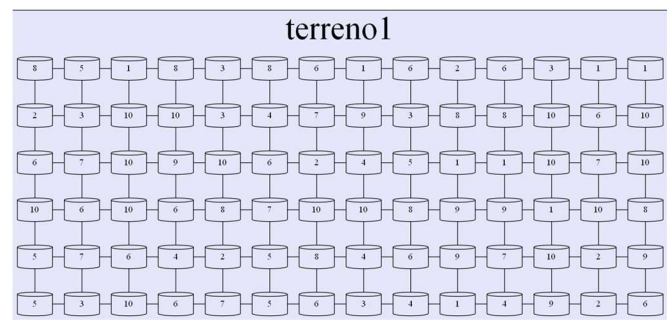


Figura 3. Gráfica del terreno de prueba.

Fuente: Elaboración propia

Aparte de generar el archivo tipo png que muestra el terreno antes de realizar el camino ideal, escribe el código en un archivo con extensión "dot" el cual es modificable y presenta la estructura para realizar graficación de terrenos con nodos en 2 dimensiones.

## Conclusiones

El correcto uso del lenguaje de programación Python, junto al paradigma de la POO y los TDAs permite la solución de una infinidad de problemas, junto a la optimización de una gran cantidad de procesos. En este caso específico se realizó un algoritmo de

búsqueda de el camino óptimo para reducir el gasto de combustible en un robot. Un ejemplo más cercano a la realidad de este programa sería un algoritmo que nos presenta la ruta más corta para llegar a un destino dentro de nuestra ciudad.

Las ventajas presentadas por estas aplicaciones de la programación es la capacidad de llevar a la realidad algo que teníamos nada más en la imaginación.

Algo importante a resaltar, es que para el manejo de los TDAs debemos comprender primero el concepto, de lo contrario no podremos utilizarlos de forma eficiente.

### **Referencias bibliográficas**

Amer-Yahia, S., Koudas, N., Marian, A., Srivastava, D., & Toman, D. (2005, August). Structure and content scoring for XML. In *Proceedings of the 31st international conference on Very large data bases* (pp. 361-372).

Restrepo, J. H., & Sánchez, J. J. (2004). Aplicación de la teoría de grafos y el algoritmo de Dijkstra para determinar las distancias y las rutas más cortas en una ciudad. *Scientia et technica*, 10(26), 121-126.

Gansner, E. R. (2009). Drawing graphs with Graphviz. Technical Report, Technical Report.

## Anexos

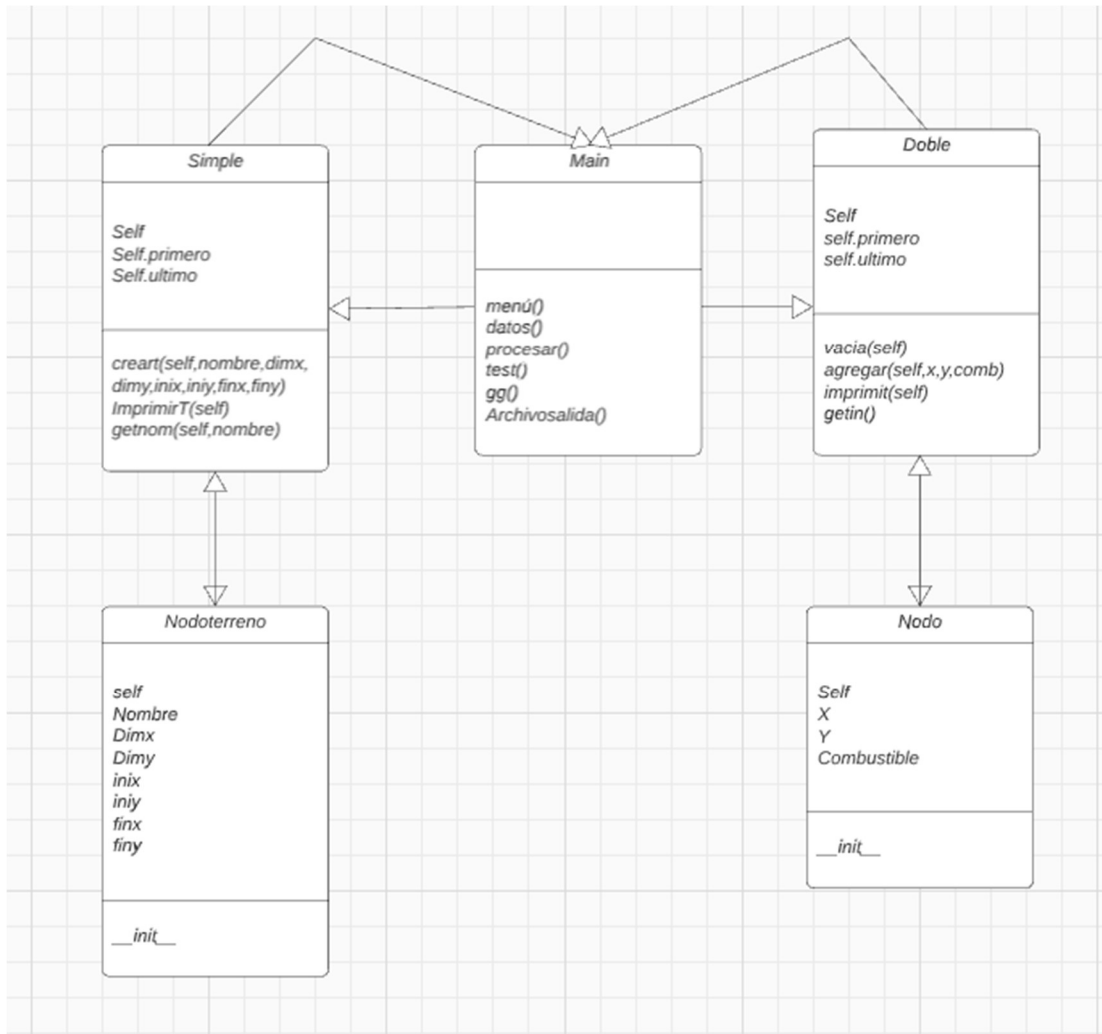


Figura 4. Diagrama de clases

Fuente: Elaboración propia