

---

## RUTA OPTIMA DE UN TERRENO PARA EL ROBOT DE EXPLORACIÓN R2E2

---

201906558 – Eddy Fernando Díaz Galindo

### Resumen

Se realizó la implementación de un código optimo para el robot r2e2, dicho robot puede estar en terrenos que se necesitan explorar, para ello el análisis que se hizo para que completara la ruta; tuvo varias consideraciones que son el combustible gastado y la posición final. Debido a que el robot solo se puede mover según los puntos cardinales se utilizó expresiones condicionales de Python para que detecte el menor combustible posible.

Se utilizó programación orientada a objetos (POO), para modelar un programa que separa el problema en problemas más pequeños para tener mejores resultados, al utilizar este paradigma se analizó la mayoría de los conflictos que puede llegar a tener el robot.

También para que el usuario tenga mejor experiencia con el programa, se utilizó el software Graphviz para representar gráficos/grafos de la información del terreno a explorar. Si el usuario quiere tener la ruta que debe tomar el robot, se utilizó el metalenguaje XML.

### Palabras clave

Python, XML, Graphviz, r2e2, Grafos, nodos.

### Abstract

*It has been made the implementation of an optimal code for the r2e2 robot was carried out, said robot may be on land that needs to be explored, for this the analysis that was made to complete the route; He had several considerations which are the spent fuel and the final position. Because the robot can only move according to the cardinal points, Python conditional expressions are used to detect as little fuel as possible.*

*Object-oriented programming (OOP) was used to model a program that separates the problem into smaller problems to obtain better results. By using this paradigm, most of the conflicts that the robot may have were analyzed.*

*Also for the user to have a better experience with the program, the Graphviz software was used to represent graphs / graphs of the terrain information to be explored. If the user wants to have the path that the robot should take, the XML metalanguage was used*

### Keywords

*Python, XML, Graphviz, r2e2, graphs, nodes.*

## Introducción

Al analizar matemáticamente el terreno, se utilizó como primer apoyo un terreno cuadrado que fue de cinco filas por cinco columnas para luego establecer un algoritmo que al analizar dicho terreno pueda llegar también a analizar terrenos cuadrados o rectangulares o de mayor tamaño hasta de tamaño de cien por cien, según este análisis se encontraron mas problemas que al utilizar programación orientada a objetos se obtuvo que separar el terreno por más cuadros más pequeños que cada uno contienen el combustible gastado en cada lugar (nodos).

Los nodos fueron analizados según la posición en la que se encuentra en el terreno, si analizamos el nodo tres coma tres (3,3) del terreno anteriormente descrito, nos damos cuenta de que se puede mover hacia el norte, sur, este y oeste pero el terreno tiene mas nodos, el nodo uno coma uno (1,1) tiene la característica que se puede mover hacia el sur o al este pero ¿habrán otros nodos que se puedan mover de diferente manera?

## Desarrollo del tema

Se utilizó archivos XML un metalenguaje que define la sintaxis utilizada para definir etiquetas estructurados. Para que la modificación y la eliminación del archivo sean relativamente simples mediante programación en Python, este tiene una biblioteca incorporada ElementTree, que tiene funciones para leer y manipular XML. Con esta librería se extraen los datos para luego introducirla a la respectiva lista que fueron creadas.

Una lista es una estructura dinámica de datos que contiene una colección de elementos homogéneos (del mismo tipo) de manera que se establece entre

ellos un orden. Es decir, cada elemento, menos el primero, tiene un predecesor, y cada elemento, menos el último, tiene un sucesor.

Podemos distinguir, atendiendo a la organización de los nodos, entre:

Listas simplemente enlazadas: cada nodo tiene un apuntador que apunta al siguiente nodo.

Listas doblemente enlazadas: cada nodo dispone de un apuntador que apunta al siguiente nodo, y otro que apunta al nodo anterior.

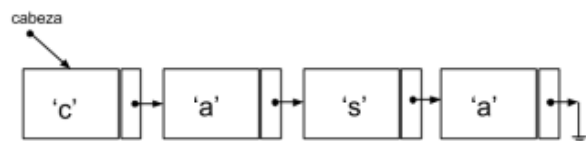


Figura 1. Lista simplemente enlazada.

Fuente: José Fager, W. Libardo Pantoja Yépez, 2014 y pag. 132.

El término estructura de datos se utiliza para referirse a una forma de organizar un conjunto de datos que se relacionan entre sí, sean estos simples o estructurados, con el objetivo de facilitar su manipulación y de operarlo como un todo.

Utilizando este tipo de estructura de datos se necesitó utilizar tres listas simplemente enlazadas, que contienen:

- Lista Simplemente del terreno: Contiene la información del nombre de cada terreno y la posición inicial del r2e2 y final del r2e2
- Lista Simplemente de posiciones: Contiene la información del tamaño del terreno, separado

por posiciones en x y y, junto con el su respectivo combustible.

- Lista Simplemente de archivo de salida: Contiene la información de la mejor ruta que debe tomar el r2e2 y muestra cuanto combustible gastará.

Sobre una estructura de datos se puede efectuar diferentes tipos de operaciones, entre las mas importantes y que fue utilizada en la realización de la solución

- Inserción: Es aquella mediante la cual se incluye un nuevo elemento en la estructura.
- Modificación: Permite variar parcial o totalmente el contenido de la información de los elementos de la estructura.
- Eliminación: Como su nombre lo indica, es la que permite suprimir elementos de la estructura
- Búsqueda: Permite determinar si un elemento se encuentra o no en la estructura.
- Consulta de la información: Permite obtener información de uno o más elementos de la estructura.

Estas listas además están enlazadas entre si, la lista del terreno depende de la lista de posiciones; para que funcionara correctamente el algoritmo de la mejor ruta, primero el usuario debe de ingresar el nombre del terreno a la lista del terreno para que la lista de posiciones ya conociendo este dato, pueda buscar la ruta de ese terreno.

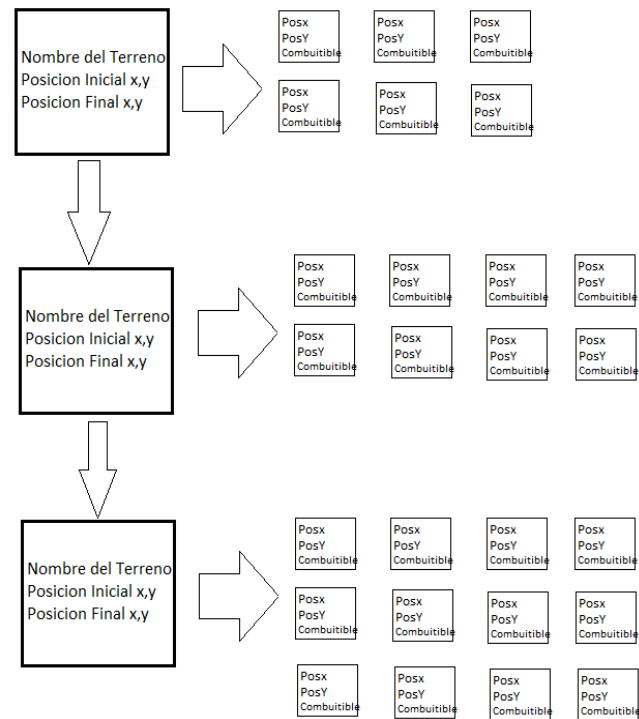


Figura 11. Listas dependiente de otra lista.

Fuente: Elaboración propia, 2021

El análisis que tuvo la lista simplemente de posiciones tuvo como prioridad buscar primeramente la posición inicial del r2e2, pero, la mayor prioridad que se puede notar si se lee el código, fueron los métodos de recorrer la lista que contiene el combustible para crear un método que pueda ordenar una ruta con el menor combustible.

Se había hablado anteriormente de que un nodo dependiendo de la ubicación en la que se encuentra puede tener características únicas, así que después de que el comportamiento de cada nodo, en listamos sus características:

- Esquina Superior Izquierda: La mayoría de los terrenos inician en esta posición y es de mucha importancia que lo analicemos, este

solamente se puede mover hacia el este y al sur.

- Esquina superior derecha: Si el r2e2 se encuentra en este nodo, este solamente se puede mover hacia el este y al sur
- Esquina inferior izquierda: Si el r2e2 se encuentra en este nodo este solamente se puede mover hacia el norte y al este
- Esquina inferior derecha: Si el r2e2 se encuentra en este nodo este solamente se puede mover hacia el norte y al oeste

Los bordes son aquellos nodos que están en los límites del terreno exceptuando las esquinas.

- Bordes Izquierdos: Si el r2e2 se encuentra en este nodo este solamente se puede mover hacia el norte, este y sur
- Bordes derechos: Si el r2e2 se encuentra en este nodo este solamente se puede mover hacia el norte, este y sur
- Bordes superiores: Si el r2e2 se encuentra en este nodo este solamente se puede mover hacia el sur, este y oeste
- Bordes inferiores: Si el r2e2 se encuentra en este nodo este solamente se puede mover hacia el norte este y oeste

Finalmente se describe el ultimo caso que puede tener un nodo.

- Centrales: Si el r2e2 se encuentra en este nodo este solamente se podrá mover hacia cualquiera de los puntos cardinales.

Aun así, se encontraron más problemas que se solucionaron agregándole más sentencias de condición, uno de los problemas es cuando un

nodo se debe de mover hacia otro nodo, pero los siguientes nodos tienen igual valor de combustible.

Este algoritmo comparado con otros algoritmos puede tener mejores o peores factores que influyen en la eficiencia de un algoritmo, pero tomas en cuenta factores que sean externos al algoritmo como la computadora donde se ejecuta

El Hardware. El procesador es lo que más se toma en cuenta ya que el programa tiene muchas sentencias de control y también la frecuencia de trabajo, memoria, discos, etc.

El Software. Puede variar dependiendo del sistema operativo y hasta lenguaje de programación, pero tenemos que recordar que Python es un lenguaje de alto nivel.

El XML de salida que fue creado contiene el nombre del terreno que se está evaluando en la lista de terrenos, este contiene la suma del combustible de la ruta óptima.

### Graphviz:

La intención del diseño original es realizar un diseño automático de gráficos dirigidos / gráficos no dirigidos para este caso se utiliza una tabla que simula un terreno, al utilizar esta herramienta debemos tener bien definidos como funciona este sub-lenguaje

### Conclusiones

Aunque existen métodos de búsqueda de ordenamiento de números (bubblesort, Quicksort), estos analizan todo el terreno y tendríamos que reestructurar el análisis del algoritmo; si se aplica estos métodos de ordenamiento, tendríamos que quitar los casos de cada nodo.

Existen modelos matemáticos que resuelven caminos para el r2e2 pero consumen mas recursos de la computadora.

### Referencias bibliográficas

José Fager, (2014). *Estructura de datos*. Proyecto LATIn.

### Extensión:

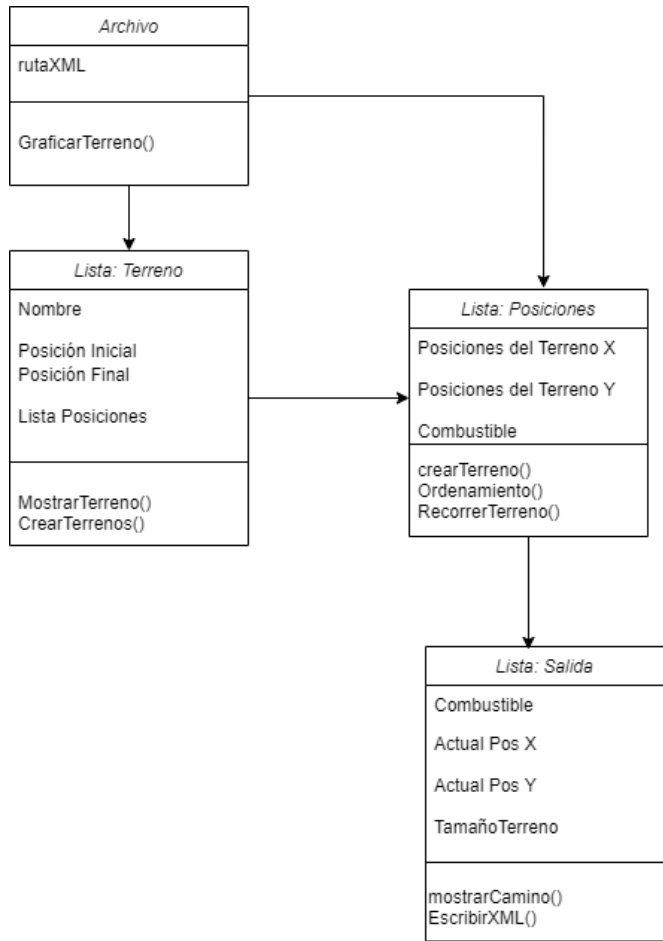


Figura III. Diagrama de Clases.

Fuente: Elaboración propia, 2021