

---

## PROYECTO 1 – ALGORITMO RECORRIDO DE EXPLORACIÓN DEL ROBOT R2E2

---

201701548 – Arnoldo Luis Antonio González Camey

### Resumen

El proyecto consiste en la realización de un algoritmo para solventar el problema del nuevo robot de exploración r2e2, el cual cuenta con la habilidad de explorar y desplazarse en cualquier tipo de terreno, únicamente utilizando más combustible en ciertos casos, la restricción que tiene r2e2 es que puede solo moverse ortogonalmente.

El algoritmo debe ser capaz de encontrar el camino más adecuado para r2e2 con el cual llegue a su destino final utilizando la menor cantidad de combustible posible; el algoritmo debe ser planteado de forma lógica y aplicando los conceptos de la programación orientada a objetos, la utilización de los conceptos de tipo de dato abstracto y la implementación en la memoria dinámica.

Se hizo uso de una lista doblemente enlazada donde se guardan las clases que contienen atributos necesarios en nodos para seguidamente recorrerlos y mediante una serie de validaciones encontrar el camino idóneo, a su vez realizar el reporte del camino tomado por el r2e2 en un archivo XML y una gráfica detallando el terreno y el recorrido en Graphviz.

### Palabras clave

Algoritmo – TDA – POO – XML – Nodo

### Abstract

*The project consists of the realization of an algorithm for solving the problem of the new exploration robot r2e2, the robot has the ability to explore and move in any terrain type, only using more fuel in some cases, the restriction than has r2e2 is it can only move orthogonally.*

*The algorithm must be able to find the best way for r2e2 with which it reaches its final destination using the least amount possible of fuel; the algorithm must be logically proposed and applying the concepts of objects oriented programming, using the concept of abstract data types and the implementation of the dynamic memory.*

*A doubly linked list was used in which the classes containing the attributes necessary in nodes, then traversing the nodes and of some validations find the perfect way, made the report with the travel taken for r2e2 in an XML file and a graph detailing the terrain and the path in Graphviz*

### Keywords

Algorithm – ADT – OOP – XML – Node

## Introducción

El siguiente proyecto consta de la realización de un algoritmo que solventa el problema planteado en el lenguaje de programación Python, implementando los conceptos de TDA, POO, el uso de XML y Graphviz. El algoritmo consiste en solventar el problema del robot r2e2 el cual cuenta con la habilidad de movilizarse en diferentes terrenos, pero solo lo puede realizar ortogonalmente.

Para ello se implementó una lista doblemente enlazada en donde se guarda una clase, que contiene los atributos de las posiciones del terreno, esto se va almacenando en unos nodos con indicadores los cuales se van recorriendo y luego de unas validaciones se obtiene el camino que debe seguir r2e2 para llegar a su destino final utilizando la menor cantidad de combustible.

Al finalizar se muestra los reportes solicitados tanto en XML como con la herramienta Graphviz.

## Desarrollo del tema

La Agencia Guatemalteca de Investigación Espacial (AGIE) ha diseñado un nuevo robot de exploración, llamado r2e2, el cual tiene la habilidad de desplazarse en terrenos irregulares utilizando solamente más combustible, el programa consiste en calcular el camino que debe tomar r2e2 para llegar a su destino utilizando la menor cantidad de combustible, el movimiento de r2e2 es ortogonal, es decir, se mueve norte, sur, este, oeste de su posición.

Al iniciar el programa se le muestra al usuario un menú en el cual debe de seleccionar la opción deseada entre cargar el archivo, procesar el terreno, escribir el archivo de salida, mostrar los datos del estudiante o generar la grafica del terreno en Graphviz.

### a. Cargar archivo

La información del terreno se encuentra en un archivo XML, en este se encuentra el nombre del terreno, sus dimensiones, la posición inicial y final de r2e2 y las coordenadas de cada segmento del terreno con su respectivo valor de gasto de combustible. Si el usuario elige la opción del menú cargar archivo se procede a leer el archivo por medio de ElementTree el cual es una librería de Python que permite analizar y navegar por un documento XML, para ello se abre el archivo y mediante la sintaxis adecuada y un ciclo for se almacenan los datos en variables temporales para seguido almacenarlos en las respectivas clases y nodos correspondientes.

Para almacenar los datos en una lista doblemente enlazada se hace uso del siguiente algoritmo, primero que nada se crea un nuevo nodo y se guarda en la clase correspondiente, se aumenta en uno el tamaño de la lista y se valida si inicio en nulo, si es así se procede a guardar al nodo como el primero de la lista y a su vez el último de la lista, si esto no se cumple se guarda en una variable temporal la posición inicial y se procede a recorrer la lista por medio de un ciclo hasta que es sea nulo, en el momento que el nodo siguiente sea nulo acaba el ciclo y almacena al nuevo nodo como el siguiente de la posición actual y como el anterior del nuevo nodo al nodo actual y el final de la lista se convierte en el nodo que se acaba de ingresar.

### b. Procesar Terreno

Después que los datos de la coordenada del terreno y el valor del gasto de combustible se encuentran en los nodos correspondientes si el usuario desea procesar el terreno se le pedirá el nombre del terreno a procesar, luego mediante un ciclo while el cual

compara el nombre ingresado por el usuario con los nodos de terrenos y devuelve el índice que tiene dicho terreno, seguido se procede con otro método que es el encargado de buscar la mejor ruta para eso se inicializa una nueva variable con el inicio del nodo y otra con el final del nodo, luego se encuentra el tamaño del terreno; seguido se valida si r2e2 aun tiene combustible y si la posición actual del nodo es igual al índice del terreno, se crean unas variables auxiliares las cuales sirven para saber si en la posición que se encuentra actualmente el nodo tiene derecha, izquierda, arriba o abajo. Primero valida la derecha si el nodo siguiente es nulo o su coordenada en “y” es menor a la coordenada actual del nodo en “y” estable que no hay una posición a la derecha. Luego continua con el lado izquierdo, para este se valida si el nodo anterior al actual es nulo o si el valor de la coordenada en “y” del nodo anterior es mayor a la coordenada en “y” del actual si alguno de los dos se cumple, este procede a establecer que no existe un lado izquierdo a la posición actual del nodo. Ahora se procede averiguar si existe una posición abajo del nodo actual para esto se hace ayuda de la variable que contiene la dimensión en “y” del terreno y mediante un ciclo que recorre la lista hasta que la posición actual sumado la dimensión en “y” del terreno sea igual a alguna otra posición en el terreno si esto ocurre se establece que existe un nodo justo debajo de dicha posición. Por último, se procede a realizar la validación para saber si existe una posición arriba de la posición actual; para esto igualmente que con la posición de abajo se hace uso de la dimensión en “y” del terreno y mediante un ciclo que recorre la lista hasta que la posición actual restado la dimensión en “y” sea igual a alguna otra y si esto sucede se establece que esa posición se encuentra justo encima de la posición actual.

Continuando con el método se valida que posiciones si existen y se inserta un nuevo nodo en otra lista doblemente enlazada, inmediatamente se procede a ordenar los nodos por su valor de gasto combustible por medio de un algoritmo denominado Bubble Sort el cual toma el primer nodo y lo compara con el siguiente si es mayor lo cambia, si no lo deja como esta y procede con los otros dos, y así hasta ordenar toda la lista. Al finalizar el ordenamiento se obtiene la posición elegida para ser la siguiente; se le suma el valor del combustible de la posición elegida a una variable y se procede a repetir todo el método con la posición elegida, hasta que r2e2 encuentre su destino final, en dado caso que r2e2 no logre encontrar el camino se mostrará un mensaje señalando que el robot no pudo explorar dicho terreno. Al finalizar se muestra el gasto de combustible de dicho viaje, el camino tomado por r2e2 en consola y el tamaño del terreno que exploró.

Si el usuario decide que r2e2 vuelva a explorar el mismo terreno y para evitar que se crea cierto conflicto con el terreno ya explorado y las coordenadas transitas por el robot se hace uso de un método llamado limpieza el cual se encarga de borrar los rastros anteriores que fue dejando el robot en su travesía.

### c. Escribir archivo de salida

Si el usuario desea escribir el archivo de salida debe seleccionar la opción en el menú y escribir el nombre que desea que el archivo tenga, al hacerlo primero se valida si el usuario anteriormente procesó algún terreno para realizar el archivo de salida del último terreno procesado, si esto es verdadero se guarda la estructura básica de un archivo XML en variables temporales, también se guarda el nombre del terreno.

Posteriormente se hace la instancia de un método encargado de guardar las coordenadas y el valor del combustible gastado en esa parte del trayecto del recorrido tomado por r2e2 en una variable que se retorna al finalizar dicho método. Al finalizar se unen las variables en una sola y se hace la instancia al método open y write los cuales sirven para escribir un archivo.

#### d. Datos del estudiante

Como cuarta opción el menú para el usuario es elegir ver los datos del estudiante para esto simplemente se hace la instancia a un método que imprime dichos datos en la consola.

#### e. Generar gráfica

Si se pretende generar el recorrido tomado por r2e2 y el terreno explorado por el mismo en una imagen, el usuario debe seleccionar en el programa la opción generar gráfica y en la lista de terrenos seleccionar el número del terreno que desea graficar, esta opción sirve para generar una imagen en formato png, para ello se hace uso de un método el cual consiste en crear la estructura básica de un archivo en Graphviz, con el tipo forma que se quiere que tengan los nodos, el color, el tipo de flecha. Primero se van creando variables denominadas filas, tantas como filas existan en el terreno, lo mismo para las columnas, seguido de esto se enlaza la raíz previamente creada con la fila uno y la columna uno. Posteriormente se hace uso del método rank el cual consiste en alinear un nodo seguido de otro nodo, eso se realiza para la raíz y las columnas.

Luego con otro método que es el encargado de generar los nodos internos se comienza a recorrer la

lista de posiciones en el índice correspondiente; se procede determinar si el grafo que se está realizando es de el terreno antes o después de ser explorado por r2e2. Se crean los nodos necesarios dependiendo las dimensiones del terreno seleccionado y se enlazan unos con otros con otros ciclos que validan si la posición del nodo está arriba, abajo, izquierda o derecha; al finalizar el método se retorna la variable que contiene todos los datos recopilados a lo largo del proceso. Luego se hace la instancia del método open y write los cuales sirven para escribir un archivo con extensión dot, seguido de esto se hace la instancia del método system el cual hace referencia a la consola y ejecuta un comando para convertir ese archivo dot a una imagen png, luego se abre la imagen por medio del método startfile.

### Conclusiones

- La utilización de listas doblemente enlazadas al principio puede ser algo confuso debido a que se debe tener muy presente en el nodo en que se encuentra, como moverse en la lista, pero son una herramienta muy útil en ciertos aspectos, ya que siempre tienes referencias al anterior y al siguiente.
- Las listas son dinámicas, no se establece cual es su dimensión al crearlo como se hace en un array, estás pueden ser de dos posiciones como de mil, esto sirve para no apartar espacio de memoria para datos que quizás nunca entren y tampoco te limitan en la cantidad de datos a ingresar
- La utilización de Graphviz como parte del reporte es una gran ayuda visual para representar el recorrido, también sirve para comprender de una manera más evidente como están conectados los nodos entre sí.

## Anexos

```
<<<<<<<<< Menú >>>>>>>>>
1. Cargar Archivo
2. Procesar Terreno
3. Escribir Archivo Salida
4. Mostrar Datos del Estudiante
5. Generar Gráfica
6. Salir
Elige una opción: 1
```

Figura 1. Menú del programa

Fuente: elaboración propia

```
Opción Cargar Archivo:
Ingrese la Ruta del Archivo: entrada.xml
Archivo cargado con éxito
```

Figura 2. Opción 1 Cargar Archivo

Fuente: elaboración propia

```
Elige una opción: 2

Opción Procesar Terreno:
Escribe el nombre del terreno a procesar: terreno1
1      2      5      3      2
4      1      4      2      6
3      1      1      3      3
5      2      3      1      2
2      1      1      1      1

- Calculando el tamaño del terreno

<<<Tamaño Terreno>>>
25

- Calculando la ruta

<<<Gasto combustible>>>
11

<<<Combustible Sobrante>>>
9988

<<<Camino Elegido por R2e2 en terreno1 >>>
0      1      0      0      0
0      1      0      0      0
0      1      1      0      0
0      0      1      0      0
0      0      1      1      1
```

Figura 3. Opción 2 Procesar Terreno

Con sus salidas en consola

Fuente: elaboración propia

```
Elige una opción: 3

Opción Escribir Archivo Salida:
Ingrese la Ruta Relativa del Archivo: salida.xml
```

Figura 4. Opción 3 Escribir archivo salida

Fuente: elaboración propia

```
Elige una opción: 4

Opción Datos del Estudiante:
> Arnoldo Luis Antonio González Camey
> 201701548
> Introducción a la Programación y Computación 2 Sección "D"
> Ingeniería en Ciencias y Sistemas
> 6to Semestre
```

Figura 5. Opción 4 Datos del estudiante

Fuente: elaboración propia

```
Elige una opción: 5

Opción Generar Gráfica
1 ) terreno1
2 ) terreno2
3 ) terreno3
Digite el número de la ruta del terreno a graficar: 1
```

Figura 6. Opción 5 Generar gráfica

Lista de terrenos cargados mediante el archivo

Fuente: elaboración propia

```
<terrenos>
  <terreno nombre="terreno1">
    <dimension>
      <m>5</m>
      <n>5</n>
    </dimension>
    <posicioninicio>
      <x>1</x>
      <y>2</y>
    </posicioninicio>
    <posicionfin>
      <x>5</x>
      <y>5</y>
    </posicionfin>
    <posicion x="1" y="1">1</posicion>
    <posicion x="1" y="2">2</posicion>
    <posicion x="1" y="3">5</posicion>
    <posicion x="1" y="4">3</posicion>
    <posicion x="1" y="5">2</posicion>
    <posicion x="2" y="1">4</posicion>
    <posicion x="2" y="2">1</posicion>
    <posicion x="2" y="3">4</posicion>
    <posicion x="2" y="4">2</posicion>
    <posicion x="2" y="5">6</posicion>
    <posicion x="3" y="1">3</posicion>
    <posicion x="3" y="2">1</posicion>
    <posicion x="3" y="3">1</posicion>
    .
    .
    .
```

Figura 7. Archivo de Entrada

Archivo XML con los datos del terreno a explorar

Fuente: elaboración propia

```
<terreno nombre="terreno1">
  <dimension>
    <m>5</m>
    <n>5</n>
  </dimension>
  <posicioninicio>
    <x>1</x>
    <y>2</y>
  </posicioninicio>
  <posicionfin>
    <x>5</x>
    <y>5</y>
  </posicionfin>
  <combustible>11</combustible>
  <posicion x="1" y="2">2</posicion>
  <posicion x="2" y="2">1</posicion>
  <posicion x="3" y="2">1</posicion>
  <posicion x="3" y="3">1</posicion>
  <posicion x="4" y="3">3</posicion>
  <posicion x="5" y="3">1</posicion>
  <posicion x="5" y="4">1</posicion>
  <posicion x="5" y="5">1</posicion>
</terreno>
```

Figura 8. Archivo de Salida

Archivo XML con los del trayecto de r2e2

Fuente: elaboración propia

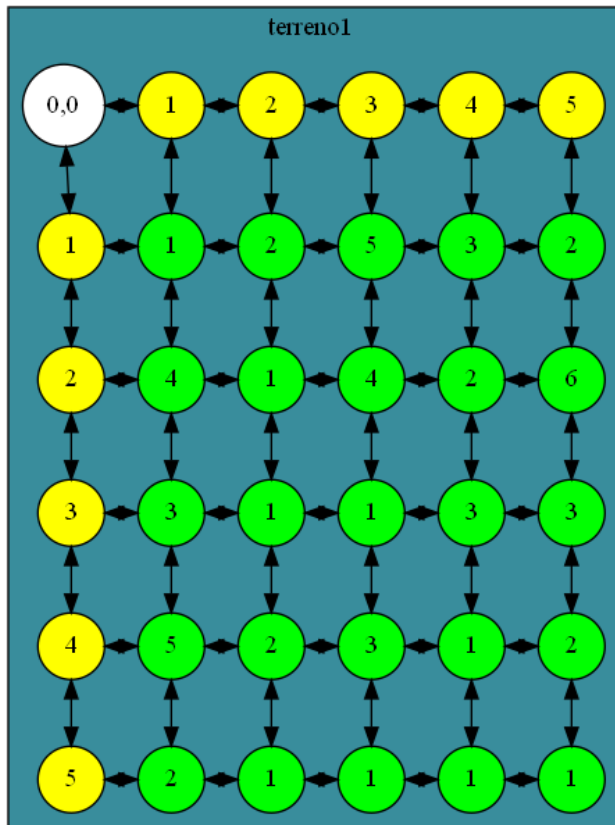


Figura 9. Gráfica terreno inexplorado

Fuente: elaboración propia

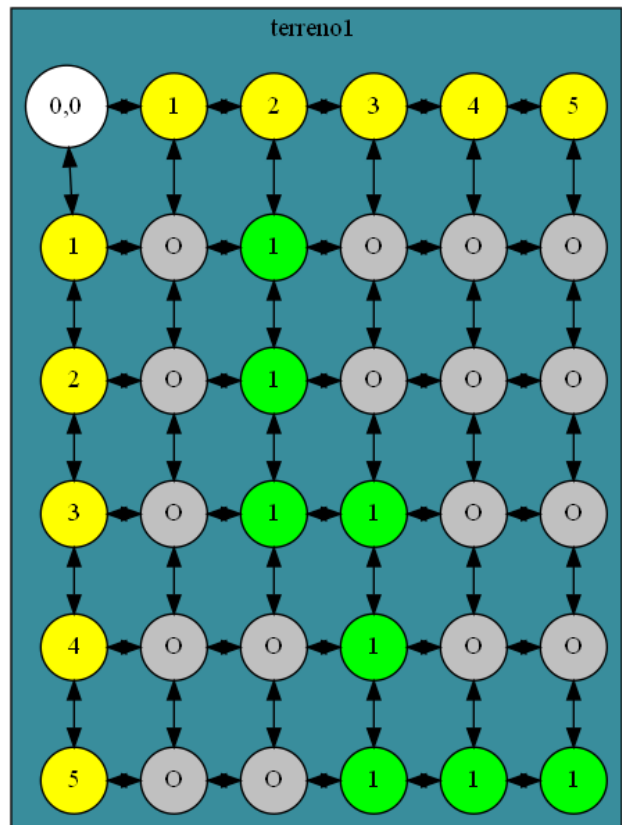


Figura 10. Gráfica terreno explorado

Gráfica del camino tomado por r2e2

Fuente: elaboración propia