
Analizador de Muestras Provenientes del Suelo de Marte Para Encontrar Algún Tipo de Vida

Carné:202100639 – Francisco Daniel Peruch de León

Resumen

En este ensayo se describe el desarrollo e implementación de un programa, el cual tiene la finalidad de proporcionar una manera para poder analizar diferentes muestras. Dichas muestras son recogidas desde Marte y a partir de ellas el Instituto Tecnológico de Palín creo un dispositivo, el cual tomando las muestras recogidas en el suelo del planeta Marte generar una cuadrícula, de determinadas dimensiones, que muestra las posiciones de todos los organismos presentes en la muestra analizada. Usando dicha cuadrículas, las cuales tienen su información guardada en un archivo de tipo XML, se ha creado un programa el cual es capaz de analizarlas. Este estudio es de mucha utilidad para poder seguir expandiendo el conocimiento acerca del planeta Marte, debido a que se puede encontrar algo tan relevante como vida en el suelo marciano y con esto serían un gran avance para el país y con ello un gran avance para el mundo.

Palabras clave

Archivo XML: Contiene información de las muestras.

Lista Enlazada: Tipo de Dato Abstracto.

Nodos: Estructuras que almacenan información.

Muestras: Contienen información de los organismos.

Abstract

This essay describes the development and implementation of a program, which has the purpose of providing a way to analyze different samples. Said samples are collected from Mars and from them the Technological Institute of Palín created a device, which, taking the samples collected from the soil of the planet Mars, generates a grid, of certain dimensions, that shows the positions of all the organisms present on the planet. analyzed sample. Using these grids, which have their information stored in an XML file, a program has been created which is capable of analyzing them. This study is very useful to be able to continue expanding knowledge about the planet Mars, because life can be found in the Martian soil and with this it would be a great advance for the country and with it a great advance for the world.

Keywords

XML file: Contains information about the samples.

Linked List: Abstract Data Type.

Nodes: Structures that store information.

Samples: They contain information about the organisms.

Introducción

Se ha diseñado un programa el cual es capaz de analizar diferentes muestras que vienen almacenada en un recuadro el cual muestra la posición de cada uno de los organismos presentes en la muestra analizada. Dicha cuadrícula es proporcionada por el Instituto Tecnológico de Palín. Para el estudio se almaceno la información concerniente a las muestras en un archivo XML, el cual es ingresado a el programa para llevar a cabo los análisis correspondientes. Con este estudio se logra un avance en la investigación de vida en el planeta Marte, lo cuál no solo es un gran avance para el país sino también para el mundo.

Desarrollo del tema:

Para la implementación del programa solicitado para el análisis de las muestras, se utilizó el lenguaje de programación Python, con este lenguaje fue desarrollado el programa. También se utilizó otro recurso llamado Graphviz, el cuál fue de ayuda al momento de generar una imagen en el cuál sea más fácil el visualizar las diferentes muestras, con sus respectivos organismos, en sus determinadas posiciones.

a. Manejo de la Información:

Para el manejo de información de las muestras, fue necesario almacenar la información en un formato el cual fuese de fácil manejo al momento de ingresar al programa.

El Instituto Tecnológico de Palín fue el encargado de procesar las muestras recogidas en el suelo del

planeta Marte, y gracias al uso de un dispositivo para analizar las muestras, este mismo genero una cuadrícula en la cual se muestran las diferentes muestras junto con la distribución de cada uno de los organismos presentes. Esta información fue manejada a través de un archivo XML. Este archivo contenía de forma estructurada todo la información relacionada a las posiciones que ocupaban cada uno de los organismos en la muestras, así como el código para identificar a que tipo de organismo pertenece cada uno. Para el procesamiento de esto datos utilizo librería “minidom”.

Usando esta librería se fue extrayendo cada uno de los datos importantes para el procesamiento de la información en el programa.

b. Lectura de la Información:

Usando la librería “minidom”, se fue accediendo a cada una de las etiquetas del archivo XML.

Se utilizaron diferentes ciclos “for” para poder acceder de nivel en nivel a la información que guardaban las etiquetas.

Se utilizaron los métodos para poder tratar la información contenido en el archivo XML:

- getElementByTagName
- firstChild
- nodeValue

c. Estructuras para guardar la información:

Se crearon estructuras en las cuales se podían guardar la información de las muestras y de sus respectivos organismos.

Se crearon principalmente dos listas para guardar la información proveniente del archivo XML. El nombre de las listas son los siguientes:

- lista_organismos
- listaDeOrganismos

d. lista_organismos:

En esta lista se guardaron los datos de cada organismo, dichos datos son:

- nombre
- código

e. Creación de lista_organismos:

Al iniciar el programa se creo esta lista, la cual fue inicializada de clase Cola(), por medio de esta clase fue llamando a un nuevo objeto, el cual es de la Clase Nodo(), este ultimo a su vez llamo a un objeto de la clase Organismo().

En cada objeto de la clase Organismo(), se fueron agregando en su constructor los datos de código y nombre de los diferentes organismo existentes en la muestra, además en el constructor también se inicializo un atributo color, el cual a través del uso

de la librería “random”, se pudo crear un código hexadecimal.

Esta inicialización de los objetos de la clase Organismo(), fueron base para la creación de los diferentes nodos de la lista_organismos, estos objetos fueron introducidos dentro de diferentes nodos. Estos nodos son de la clase Nodo().

Se usaron estos nodos para la creación de la lista_organismos, para el manejo de esta lista se utilizo una lista simplemente enlazada que tiene un comportamiento de cola.

Una vez creada la lista_organismos, al momento de leer el archivo se fue introduciendo los datos en dicha lista.

f. Creación de la listaDeOrganismos:

Debido a que la lista denominada lista_organismos es de la clase Cola(), teniendo nodos que guardan objetos de la clase Organismos(). Esta lista tiene como atributo otra lista, la cual es una lista simplemente enlazada y con el comportamiento de una cola.

La lista simplemente enlazada denominada listaDeOrganismos es una lista de la clase ColaMuestras(), la cual tiene nodos de la clase NodoMuestras(), que a su vez guarda objetos de la clase Muestra().

En los objetos de clase Muestra(), se inicializan en el constructor. Dicho constructor pide como

parámetros los valores de fila, columna y código_organismo. Para conseguir estos valores, se accede de la manera correspondiente en el archivo XML, y se va ingresando estos valores dentro de cada lista.

Debido a que existe la posibilidad de que en las muestras vengan diferentes tipos de organismos, se utilizo una función llamada IngresarPosicionesOrganismos(), la cual recibe como parámetros los valores de fila, columna y código del organismo. Dicha función pertenece a la clase Cola() de la cual es perteneciente la lista “lista_organismo”.

A través de la función antes citada se puede hacer un filtro de cada uno de los organismos y así poder almacenar los organismos de la misma clase en la lista “listaDeOrganismos”, que a su vez esta lista pertenece a un nodo que guarda la información de del organismo del tipo correspondiente a los organismos guardados en la listaDeOrganismos.

g. Creación de las imágenes con las posibles posiciones en las cuales pueden sobrevivir un organismo:

Para la resolución de este problema se debe tomar en cuenta la regla para que un organismo pueda sobrevivir la cual es: “Un organismo “X” solo puede extenderse si existe otro Organismo “Y” que sirva para alimentar la expansión, de lo contrario, el organismo “X” no podrá extenderse por las muestras y no sobrevivirá.”

Algo a tomar en cuenta de la regla antes mencionada es lo siguiente: Para que un organismo pueda extenderse se debe primero

confirmar si hay otro organismo de su mismo tipo que lo ayude a encerrar a uno o más organismo de otro tipo.

En conclusión el o los organismo de un tipo diferente al organismo que se esta analizando, deben quedar atrapados de forma horizontal, vertical o diagonal entre los organismo del tipo que se están analizando.

Para implementar una solución a este problema se debe acceder a cada una de las listas que tiene cada tipo de organismo, de esta manera se puede tomar un organismo y su posición para analizar. Luego compararlo con los demás organismos y sus posiciones determinadas y así hacer un recorrido para buscar en que posiciones se puede agregar un organismo determinada para cumplir con la regla establecida para la supervivencia de este.

h. Graficar:

Para la creación del tablero con los organismos distribuidos de la manera correspondiente, se hizo uso de la librería Graphviz, la cual proporciona diferentes maneras de graficar dependiendo lo que se necesite.

Conclusiones

- Para el desarrollo e implementación del programa solicitado se debe utilizar diferentes librerías, las cuales son de utilidad al momento de manejar la información proveniente del archivo XML y al generar las soluciones

solicitadas para determinar los lugares donde puede prosperar un organismo.

- Se debe hacer uso de la memoria dinámica, debido a que no se conoce la cantidad de información que se tendrá que manejar, por lo cual si se usa memoria estática pueden ocurrir dos problemas, el primero que el espacio reservado no sea suficiente o que se de más y se desperdicien recurso para el buen funcionamiento del programa.
- Se debe tratar de manejar la información que va a utilizar en programa en diferentes estructuras, de esta manera se tiene un orden adecuado y se facilita el poder acceder a información que será útil para la realización de las diferentes funcionalidades.

Diagrama de Clases:

