|  |
| --- |
| **EXPLORADOR DE NUEVOS TERRENOS – R2E2** |
| **202010033 – Josué Rodolfo Morales Castillo** |

**Resumen**

Se desarrolló un programa el cual puede leer archivos con extensión .xml, esto para poder obtener la información del archivo de una manera que fuese fácil y ordenada. Este archivo cuenta con nombre de terrenos y posiciones la cuales cuentan con una cantidad de gasolina, el objetivo principal del programa consiste en determinar la ruta más optima para que R2E2 llegue a su destino utilizando el menor gasto de combustible y a la vez obtener la ruta que siguió en el camino óptimo.

Para resolver lo anteriormente mencionado se utilizó el lenguaje de programación Python, ya que este nos ayuda a crear estructuras dinámicas de una manera más fácil, estas estructuras dinámicas fueron posible realizarlas gracias a que se utilizaron conceptos de TDA. Cabe resaltar que para guardar los datos del archivo XML a memoria, se utilizaron listas enlazadas.

Una lista enlazada simple es una colección de nodos que tienen una sola dirección y que en conjunto forman una estructura de datos lineal.

**Palabras clave**

Nodo, lista enlazada, matriz, objeto, tupla.

***Abstract***

*The program was developed to read files with .xml extension, this was made to get in an easy and clear way the information of the file. This file has field names and positions that have amount of gasoline, the main goal of this program is to determine the most optimal route for R2E2 to reach its destiny using the least amount of gasoline and at the same time get the route that took for the optimal way.*

*To solve what was said before, it was used the python programming language, because it helps us to create dynamic structures in an easier way, these dynamic structures were done using ADT concepts. To save the data from the XML file to memory, linked lists were used.*

*A simple linked list is a node collection that have only one direction and together they form a linear data structure.*

***Keywords***

*Node, linked list, matrix, object, tuple.*

**Introducción**

Las estructuras de datos son una forma de organizar los datos en la computadora, de manera que nos permitan realizar operaciones con ellas de forma muy eficiente.

Las estructuras de datos pueden organizarse en muchas formas diferentes; el modelo matemático o lógico de una organización particular de datos recibe el nombre de estructura de datos.

Las Estructuras de datos pueden clasificarse en lineales y no lineales. Una Estructura de datos es lineal si sus elementos forman una secuencia o, en otras palabras, una lista lineal.

Con estas estructuras tenemos la posibilidad de administrar todo tipo de datos sin ningún tipo de obstáculo, algo que en la actualidad se usa en la red para poder llevar a cabo, por ejemplo, los sistemas de indexado de contenidos. Juegan un papel clave en la creación de los mejores algoritmos, así como en su uso con lenguajes de programación que se benefician de ellas.

**Desarrollo del tema**

La estructura de datos es fundamental para cualquier sistema que disponga el equipo, de modo que pueden existir muchas formas de estructura y de organización que se puedan implementar a los datos almacenados en la computadora.

Al realizar una organización de los datos se tiene la capacidad de aplicar una serie de operaciones específicas de modo que da al usuario herramientas para el desarrollo del software. Es por ello por lo que a continuación se presentan algunos tipos de estructuras de datos y operaciones que se les pueden aplicar:

1. **Vector**

* Es un conjunto de elementos que se encuentran estructurado de una forma especial y específica.
* De una forma general cada elemento que se disponga es del mismo tipo.
* Se puede acceder a estos elementos mediante la aplicación de un entero como un índice de manera que se tenga que señalar el elemento que se desee.
* Puede presentar algunas implementaciones básicas las cuales pueden dar las palabras de la memoria adyacente de los elementos que se encuentre en cada Re-arreglo.
* Con cada modificación que se realice se puede cambiar o variar el tamaño de la longitud.
* También puede disponer de una longitud fija determinada.

1. **Vector Asociativo**

* Es una variable caracterizada por ser flexible.
* Su flexibilidad es mayor que el de una matriz.
* Da la opción de agregar pares nombre valor.
* También permite eliminar pares nombre valor.
* Cuenta con una tabla de hash.
* Facilita el arreglo asociativo que se realiza.

1. **Grafo**

* Es una estructura de datos que se encuentra conectada.
* Se encuentra constituidas por nodos.
* Cada nodo que dispone posee un valor especifico.
* También los nodos contienen referencias de otros nodos.
* Tiene la capacidad de aplicarse para dar una representación de redes.
* Puede dar referencia entre cada nodo.
* Dispone de algunas conexiones las cuales contienen direcciones, es decir, algunos de puntos de entrada y salida.

1. **Árbol**

* Consiste en un caso diferente o especifico de grafo.
* Se encuentra en la aplicación de los ciclos que no se permiten.
* Dispone de un camino a partir de un nodo hasta otro nodo.
* El nodo de partida se conoce como raíz.
* Presenta una colección de árboles el cual es comúnmente conocida como bosque.

1. **Clase**

* Es una plantilla específica.
* Aplicada para la elaboración de objetos de datos.
* Está basado en un modelo que es predefinido.
* Se emplea como una representación abstracta de conceptos.
* Introducen diversos campos como lo son los registros y las operaciones.
* Da la posibilidad de realizar una consulta por el valor de dichos campos.
* También puede cambiar los valores específicos.

1. **Pilas**

Una pila típica es un área de la memoria de los computadores con un origen fijo y un tamaño variable. Al principio, el tamaño de la pila es cero. Un puntero de pila, por lo general en forma de un registro de hardware, apunta a la más reciente localización en la pila; cuando la pila tiene un tamaño de cero, el puntero de pila de puntos en el origen de la pila.

Las dos operaciones aplicables a todas las pilas son:

* Una operación apilar, en el que un elemento de datos se coloca en el lugar apuntado por el puntero de pila, y la dirección en el puntero de pila se ajusta por el tamaño de los datos de partida.
* Una operación desapilar: un elemento de datos en la ubicación actual apuntado por el puntero de pila es eliminado, y el puntero de pila se ajusta por el tamaño de los datos de partida

1. **Cola**

La particularidad de una estructura de datos de cola es el hecho de que sólo podemos acceder al primer y al último elemento de la estructura. Así mismo, los elementos sólo se pueden eliminar por el principio y sólo se pueden añadir por el final de la cola.

Operaciones básicas:

* Crear: se crea la cola vacía.
* Encolar (añadir, entrar, push): se añade un elemento a la cola. Se añade al final de esta.
* Desencolar (sacar, salir, pop): se elimina el elemento frontal de la cola, es decir, el primer elemento que entró.
* Frente (consultar, front): se devuelve el elemento frontal de la cola, es decir, el primero elemento que entró.

1. **Listas enlazadas simples**

Es una estructura de datos lineal y dinámica que se compone de un conjunto de nodos en secuencia enlazados mediante un apuntador o referencia.

* La lista simple tiene un apuntador inicial
* El último nodo de la lista apunta a nulo
* Se recorren lógicamente los nodos de la lista simple
* Se ubica la posición definitiva de un nuevo nodo
* Se mantiene el orden lógico de los datos
* El nuevo dato se agrega al final de la lista simple
* Cada nodo tiene 2 secciones:
  + Dato (puede ser simple o compuesto)
  + Apuntador o referencia que enlaza al siguiente nodo en secuencia lógica

1. **Operaciones con estructuras de datos**

* **Recorrido**

Implica el acceder a cada registro una única vez, aunque uno o más ítems del registro sean procesados. (Este acceso o procesamiento también se denomina a veces por el término «visitar» el registro).

* **Búsqueda**

Implica la localización de un registro caracterizado por una determinada clave o también el acceso a todos los registros que cumplan una o más condiciones.

* **Inserción**

Cuando añadimos nuevos registros a la estructura.

* **Eliminación**

Operación de borrado de un registro de la estructura.

* **Ordenación**

Es la operación de clasificar los registros conforme a un orden lógico determinado (por ejemplo, alfabéticamente, de acuerdo con una clave de nombre, o numérica, de acuerdo con alguna clave de número, tal como número de [Seguridad Social](https://www.ecured.cu/Seguridad_Social) o de inventario).

* **Mezcla**

Es la operación de combinar dos archivos previamente ordenados en uno único que también lo está.

**Conclusiones**

Las estructuras de datos sirven para el manejo de grandes cantidades de datos.

Existen dos tipos de estructuras de datos, las lineales y no lineales, dependerá qué se quiere realizar para su uso.

La lisa enlazada simple es de las estructuras más fáciles de manejar, además que es lineal.

**Referencias bibliográficas**

Seymour Lipschuts, (2002). *Ph D. Estructura de datos.* Ediotrial Revolucionaria.

Gomez, Jennifer, (2020). *Estructura de datos: Descripción, ejemplos y más.* [https://tecnoinformatic.com/c- /estructura-de-datos/](https://tecnoinformatic.com/c-programacion/estructura-de-datos/)

Román Martínez/Elda Quiriga, (2001). *Estructura de datos: referencia prácticas*, México, Thomson Learning.

Narciso Martí,(2003). *Estructura de datos y métodos algorítmicos*, Madrid, McGraw Hill.

**Anexos**

**A picture containing text, receipt, screenshot

Description automatically generated**