
PROYECTO 2, COMPLETACION DE MISIONES Y RECONOCIMIENTO DE MAPAS DE COMBATE, IMPLEMENTANDO METODOS TDA Y GRAFICACION POR MEDIO DEL SOFTWARE GRAPHVIZ

202006629 – Wilber Steven Zúñiga Ruano

Resumen

Se presenta la solución para el manejo de archivos XML, el cual se procesa maneja objetos y es necesario crear una matriz para poder manipular los datos para lograr hacer varias operaciones con el mismo archivo, para eso usaremos las herramientas de Gestión de ficheros XML que se programó en el lenguaje Python utilizando la librería mini Dom, usando librerías que representan la información estructura en el fichero XML y cómo podemos obtener información de dicha estructura, como añadir o eliminar elementos o atributos y como modificar la información de guardada. Todo es una aplicación de consola, usando programación orientado a objetos y estructura de datos para almacenar la información y obtener un mejor manejo, la lista la cual se uso fue una lista ortogonal.

Palabras clave

Lista

Nodo

TDA

XML

Objeto

Abstract

We present the solution for handling XML files, which is processed handles objects and it is necessary to create a matrix to manipulate the data to achieve several operations with the same file, for that we will use the XML file management tools that were programmed in the Python language using the Element Tree library, using libraries that represent the information structure in the XML file and how we can get information from that structure, how to add or delete elements or attributes and how to modify the stored information. Everything is a console application, using object-oriented programming and data structure to store the information and manage it with EDD, the list which was used was an orthogonal list.

Keywords

List

Node

TDA

XML

Objects

Introducción

La estructura de datos es una forma especial de organización de los datos en la computadora para que puedan conseguir un uso eficaz. Diferentes tipos de estructuras de datos son adecuadas para diferentes tipos de aplicaciones, algunas son altamente dedicadas a tareas específicas.

La estructura de datos es una forma de procesamiento de grandes cantidades de datos de forma eficaz, usos como grandes bases de datos y servicios de Internet. Generalmente hablando, la estructura de datos es la clave del diseño de algoritmos eficientes. Algunos métodos de diseños formales y lenguaje de programación destacan la estructura de datos en lugar del algoritmo como factor principal de la aplicación y del diseño de software. En programación, una estructura de datos se puede declarar inicialmente ingresando una palabra reservada, luego ingrese un identificador y nombre de la estructura, luego un identificador para cada miembro de ella.

Desarrollo del tema

Una lista enlazada es una estructura de datos dinámica. La cantidad de nodos en una lista no es fija y puede crecer y contraerse a demanda. Cualquier aplicación que tenga que tratar con un número desconocido de objetos necesitará usar una lista enlazada. Una desventaja de una lista vinculada frente a una matriz es que no permite el acceso directo a los elementos individuales. Si desea acceder a un artículo en particular, debe comenzar por la cabecera y seguir las referencias hasta que llegue a ese artículo.

Puede ser dividido en secciones estructurales que doten de coherencia al discurso.

a. Lista Ortogonal o matriz ortogonal:

En este tipo de lista se utiliza para representar matrices. Los nodos contienen cuatro apuntadores. Uno para apuntar al nodo izquierdo (li), otro para apuntar al derecho (ld), otro al nodo inferior (lb) y por último un apuntador al nodo superior (la)...

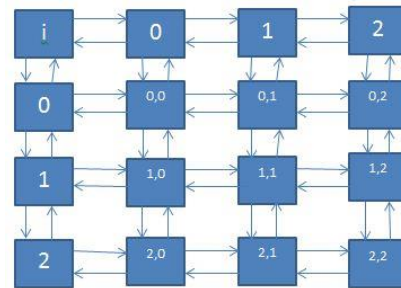


Figura 1. Matriz Ortogonal.

Fuente: Estructuras de datos, blogspot.

b. Matiz dispersa:

En programación una matriz dispersa o matriz rala o matriz hueca es una matriz de gran tamaño en la que la mayor parte de sus elementos son Nulo. Con matrices de gran tamaño los métodos tradicionales para almacenar la matriz en la memoria de una computadora o para la resolución de sistemas de ecuaciones lineales necesitan una gran cantidad de memoria y de tiempo de proceso. Se han diseñado algoritmos específicos para estos fines cuando las matrices son dispersas.

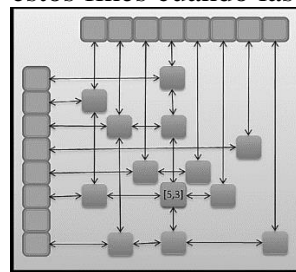


Figura 2. Matriz dispersa.

Fuente: Weissstein, 2010, Wikimedia.

El problema que nos presenta es el siguiente:

La empresa Chapín Warriors, S. A. ha desarrollado equipos automatizados para rescatar civiles y extraer recursos de las ciudades que se encuentran inmersas en conflictos bélicos. Con el fin de realizar las misiones de rescate y extracción, Chapín Warriors, S. A. ha construido drones autónomos e invisibles para los radares llamados ChapinEyes. Los ChapinEyes sobrevuelan las ciudades y construyen un mapa bidimensional de la misma, este mapa bidimensional consiste en una malla de celdas, donde cada celda es identificada como un camino, un punto de entrada, una unidad de defensa, una unidad civil, un recurso o una celda intransitable

Para poder completar el sistema de rescate y extracción, la empresa Chapín Warriors, S.A., lo ha contratado para crear un sistema de control que recibirá la malla de celdas del dron ChapinEye, a continuación, el sistema de control será capaz de definir una serie de misiones e indicará los robots que pueden completar la misión, así como los caminos que deben seguir en la malla de celdas para ejecutar las misiones.

Para la solución de este problema se usó el lenguaje Python, el cual importamos las librerías para leer el archivo, usamos la librería de “mini Dom” para la manipulación de archivos XML y poder pasarlas listas enlazadas que es el tema principal del proyecto. Para todo este proceso usamos programación orientado a objetos, el cual creamos varias clases y métodos de herencia.

Esto nos ayudó a crear una mejor estructura de datos, las clases fueron las siguientes:

- main
- Cuadrato
- Robot
- Ciudad
- nodoCuadrato
- nodoRobot
- nodoCiudad

- listaCuadritos
- listaRobots
- listaCiudad

La clase principal es main, donde se manejan todos los métodos, se inicia manteniendo un menú en ejecución con las opciones que el programa ofrece, alelegir alguna entra a un submenú, a método de ejemplo, la opción número 1 que es cargar el archivo, pide la ruta del archivo XML posteriormente la pasa a una variable y usando la librería miniDom lo convierte en un archivo miniDom XML que es manipulable y de fácil acceso a todos sus atributos. Primero se obtiene todas las ciudades que vienen en el archivo junto con sus cuadritos seguidamente se almacenan creando un objeto de tipo ciudad el cual contiene una lista ortogonal de objetos tipo cuadrato que a su vez cada ciudad contiene una lista enlazada simple de objetos robots, luego se procede a correr el menú de ciudades cargadas selecciona la ciudad, luego la misión a elegir y por último el dron a utilizar en la misión. Procede a recorrerse y hacer el camino necesario para llegar a su objetivo.

Los métodos más importantes dentro de la clase main:

1. Menú: El cual contiene todo el texto del menú y lo mantiene en ejecución, tiene las opciones de carga de archivo, mostrar pisos cargados y salir.
2. cargaArchivo: Este método anteriormente explicado es el encargado de recibir el archivoXML y obtener todos los atributos necesarios, extraerlos y pasarlos a las variables requeridas, y lleva todo el manejo de las listas enlazadas.

Tabla I.

Variables principales

CATEGORÍA	CATEGORÍA
Documentt	miniDom(XML)
listaCiudad	Lista enlazada de ciudades
listaRob	Lista enlazada de drones.

Fuente: elaboración propia.

Conclusiones

- Las estructuras de datos son un medio para manejar grandes cantidades de datos de manera eficiente para usos tales como grandes bases de datos y servicios de indización de Internet.
- Un grafo es una estructura de datos conectada compuesta por nodos. Cada nodo contiene un valor y una o más referencias a otros nodos. Los grafos pueden utilizarse para representar redes, dado que los nodos pueden referenciarse entre ellos.
- Una ventaja del uso de listas enlazadas es la optimización de memoria.

Referencias bibliográficas

- C. J. Date, (1991). *An introduction to Database Systems*. Addison-Wesley Publishing Company, Inc.
- Narciso Martí, Estructura de datos y métodos algorítmicos, Madrid, McGraw Hill, 2003
- Román Martínez/Elda Quiriga, Estructura de datos: referencias prácticas, México, Thomson Learning, 2001.

Anexos

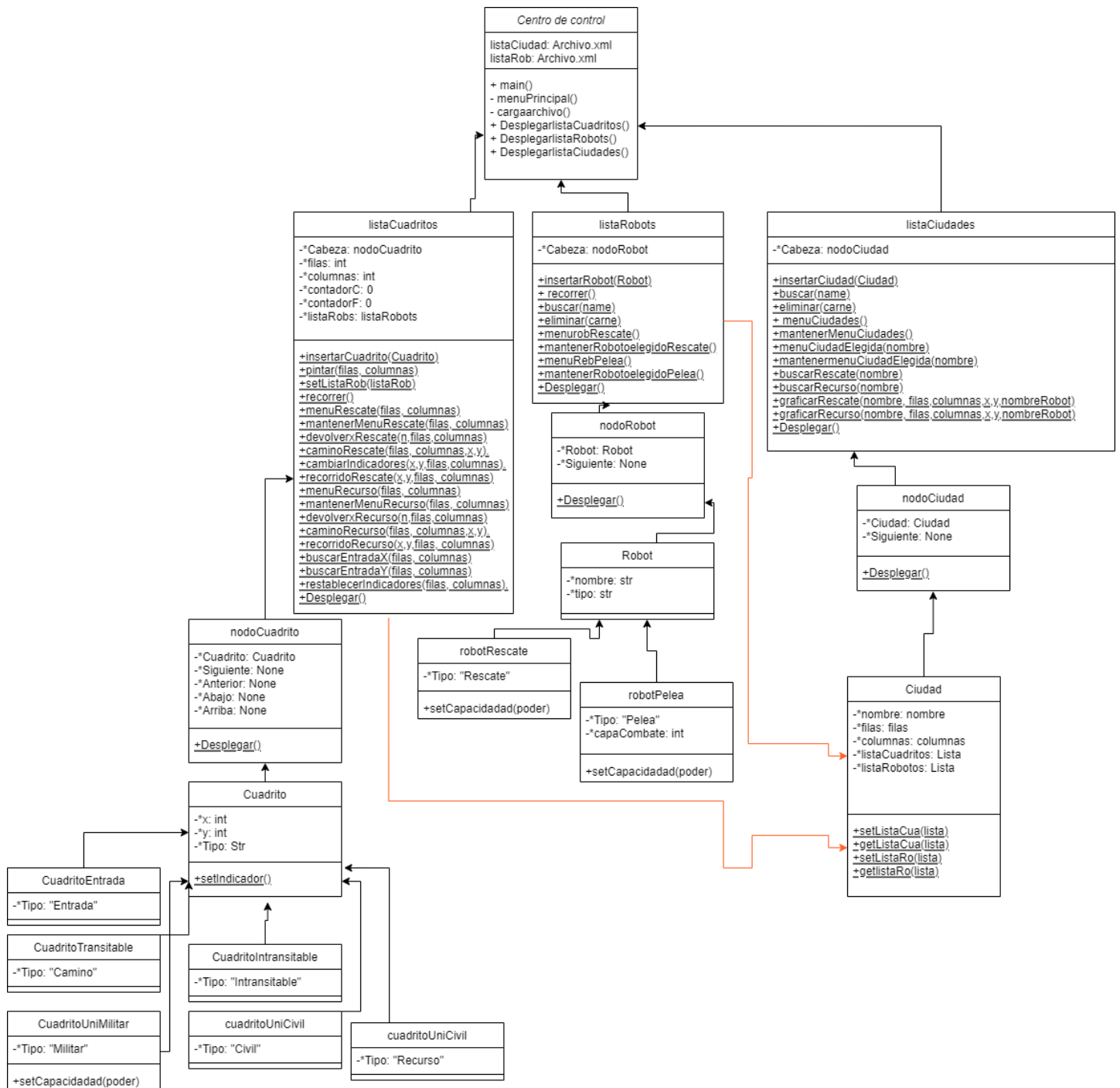


Diagrama de clases del proyecto.

Fuente: Elaboración propia.