
R2E2 TDA-IMPLEMENTATION

202001534 – Erwin Fernando Vásquez Peñate

Resumen

Las listas enlazadas son estructuras de datos fundamentales conformados con nodos, los anteriormente mencionados son un conjunto de secuencias que guardan datos de manera arbitraria, los cuáles son referenciados mediante punteros, estos punteros pueden apuntar al nodo anterior o posterior, según sea la naturaleza de la lista enlazada.

Los archivos de texto plano como por ejemplo los xml, últimamente son muy utilizados y esto se debe a que su estructura es muy amigable tanto como para lectura, como para implementar Programación orientada a objetos (POO).

En términos generales la necesidad de ver gráficamente lo que ocurre en nuestras aplicaciones es muy importante, ya que de esta manera es mucho más fácil entenderlo y poder ayudar a los demás usuarios a entenderlo, por eso la implementación de herramientas como graphviz es innovador e indispensable a la vez.

Dadas las necesidades anteriores y el impacto en la programación es necesaria la implementación conjunta de las herramientas mencionadas.

Palabras clave

- Listas Enlazadas
- POO
- Estructura Xml
- Graphviz

Abstract

The linked lists are a peculiar data structures, that are conformed by nodes, that contains specific information, like objects, typed attributes, etc. The linked lists also have pointers, this is very important, because with these pointers the linked list can be always in communication with her nodes and the information persist.

The text plain files, like xml, in the present are very used by the developer community, because these files are very helpful, with storing information and be able to read it easily, in addition they are a good example of Object-oriented programming (POO).

In the abstract applications is necessary to control application flow, so with this need, we turn to alternative options as support tools, like graphviz and with this, we can observe in a graphic way the operation of our work.

Given the above needs and the impact on programming, the joint implementation of the aforementioned tools is necessary.

Keywords

- Linked Lists
- POO
- Xml Structure
- Graphviz

Introducción

Conociendo la importancia de diversas herramientas para la creación de aplicaciones muy abstractas, podemos adentrarnos a un mundo nuevo y lleno de incógnitas que mediante investigación y sobre todo mucha práctica pueden ser respondidas de una manera muy dócil, basándonos en el problema general, nos encontramos en una situación donde una empresa necesita mediante un satélite poder comunicarse mediante coordenadas con un robot que mediante las indicaciones podrá moverse (solamente ortogonalmente) sobre diversos tipos de terrenos, conociendo también que cada porción de terreno necesita una cantidad específica de gasolina para su traslado, hay algunas condiciones iniciales, tales como que el robot tiene un límite de gasolina y por lo tanto se necesita mediante un algoritmo calcular la ruta óptima para que el recorrido pueda ser llevado a cabo de una manera satisfactoria y ahorrando la mayor cantidad de gasolina posible.

Desarrollo del tema

Al inicio podemos encontrarnos perdidos dentro de un mundo nuevo lleno de incógnitas, pero mediante el adentramiento a este “nuevo mundo” llamado programación podemos darnos cuenta que siempre se aprenden nuevas cosas y nuevas habilidades, que seguramente antes no teníamos.

Hablando un poco del problema central de la aplicación, necesitamos tener 2 cosas en la mente, la primera es que tiene que ser una aplicación amigable para el usuario por lo tanto debeos contar con un menú interactivo y fácil de utilizar, mientras más fácil sea de utilizar para el usuario, será mucho mejor la repercusión y aceptación que pueda tener el proyecto,

Al implementar herramientas como archivos con extensión “Xml”, no solo tenemos que estar preparados para su posible lectura y escritura, sino que también conocer cada una de sus características, y en este proyecto hablar de los archivos xml es dar con “la punta del icebeg”, pues mediante el conocimiento de su estructura podemos darnos cuenta, que un archivo “Xml” no es más que una jerarquía sintáctica de objetos, ya que mientras más nos vallamos sumergiendo en el archivo encontraremos distintas ramas, las cuales pueden llamarse “nodos”.

Para adentrarnos un poco más a la estructura de la aplicación se muestra un posible prototipo para el menú principal de la aplicación y posteriormente pueden exponerse cada una de las funciones.

Menú principal:

1. Cargar archivo
2. Procesar archivo
3. Escribir archivo salida
4. Mostrar datos del estudiante
5. Generar gráfica
6. Salida

Figura 1. Prototipo menú principal.

Fuente: Universidad de san Carlos de Guatemala, Enunciado
Proyecto 01, página 07.

Según se vea el problema inicial se puede intentar resolver de muchas formas, pero muchas de ellas son poco efectivas y mucho más complejas, para ello hay que entender con que se está tratando.

Teniendo un prototipo del menú podemos darle un punto de partida a nuestro flujo de aplicación, empezando por lo menos abstracto y dejar lo más complejo para evitar la pérdida innecesaria de tiempo e ideas.

a. Manejo de archivos Xml

La lectura del “xml” podemos adelantarnos a pensar que puede ser muy complicada y compleja pero la realidad es otra ya que con distintas librerías del lenguaje Python podemos leerlos e incluso escribir archivos de esta naturaleza en cuestión de minutos, a continuación, se presentan algunas recomendaciones de librerías para la lectura y escritura de archivos “xml”.

Tabla I.

Librería para Lectura “XML”

FROM	IMPORT
Xml.dom	minidom

Tabla II.

Librería para Escritura “XML”

IMPORT	AS
Xml.etree.ElementTree	ET

Con la lectura del “xml”, ya podemos poner en marcha lo más importante, y es que las listas enlazadas están vacías y necesitan llenarse de objetos de nombre Terreno, cada objeto tendrá como atributo un nombre, dimensiones, posiciones iniciales y finales, además su “grid”, conocido como la rótula o la maya del terreno.

Con listas enlazadas nuestra información siempre estará conectada y eso es fundamental para todas las validaciones que tendremos que realizar en todo momento, por ejemplo, validar que el terreno que se quiere procesar exista, si no existe el terreno, la aplicación mostrará un aviso y se le solicitará uno nuevo hasta que se valide mediante la lista enlazada y cada uno de sus nodos que el terreno solicitado exista.

b. Procesar Archivo y Listas Enlazadas

En esta sección del menú principal se mostrará una lista de todos los terrenos recolectados del archivo de entrada, donde el usuario podrá primero ver el listado de los terrenos, luego mediante escribir el número del listado del terreno correspondiente, escoger uno y empezar el análisis del mismo, al momento se empezará a deducir la ruta más óptima para el traslado del robot dentro del terreno, y así poder gastar la gasolina mínima durante el recorrido.

Recordando que el listado inicial donde se podrá escoger el terreno a evaluar son todos los terrenos anteriormente almacenados en la lista enlazada, por lo cual, si el usuario introduce un número incorrecto que no pertenece a ninguno el sistema deberá validarlo y repetir el proceso de pedir otro número al usuario.

Por último, se generarán los reportes necesarios o indicados por el usuario de una manera eficiente por el sistema, y se le mostrará el menú de nuevo para que con la información recolectada se pueda escribir un archivo de salida.

Mediante este proceso se puede reflexionar y ver la importancia de las validaciones mediante las listas enlazadas, ya que la información siempre persiste y puede encontrarse de una manera eficiente, aunque en muchos casos las listas enlazadas no son la manera más rápida de buscar información y es necesario otro tipo de estructuras de datos, pero para el efecto de este proyecto las listas enlazadas simples son las implementadas dentro del mismo.

Una ventaja impresionante de usar estos métodos como las listas enlazadas, es que no te acostumbrarás a usar las funciones que ya vienen establecidas en los lenguajes (Python, en este proyecto), como métodos ya predefinidos como append, sort, pop, remove, etc.

c. Graphviz

Al tratarse de estructuras de datos es una necesidad ver de una manera gráfica y detallada lo que ocurre en el flujo de la aplicación y en ocasiones la consola o terminal no es suficiente para cumplir los objetivos, por lo cual, es necesario implementar herramientas gráficas, una de ellas es graphviz, una herramienta que mediante código y lógica podemos generar grandes ilustraciones del flujo de nuestras operaciones.

Graphviz, para serles sincero no tenía idea de que era exactamente, y hasta la fecha no he encontrado mucha información donde corroborar los mejores métodos o librerías para implementarlas, debido a esto mediante la librería de Python “os” logré implementar la herramienta al proyecto, ya que, mediante el “cmd” del computador en Windows, se puede ejecutar y abrir los archivos deseados.

d. Datos de Estudiante

La cuarta opción del menú, donde mediante un pequeño apartado se pueden corroborar los datos del estudiante que diseño la aplicación, y también se indica el curso y sección al cual pertenece.

```
=====
MasterMind:
Erwin Fernando Vásquez Peñate
202001534
Introducción a la programación y computación 2 Sección 'E'
Ingeniería en ciencias y sistemas
4to semestre
=====
```

Figura 2. Datos estudiantiles.

Fuente: elaboración propia, 2021.

e. Ventajas y desventajas de las listas enlazadas

Se ha discutido a fondo cada tema expuesto en este proyecto, pero en esta sección se enfatizará más en conocer algunas de las ventajas y desventajas de las listas enlazadas y su aplicación en la programación actualmente.

Empezando por las ventajas, podemos destacar algunos aspectos muy específicos, tomando en cuenta que el tema es un tanto abstracto, puede aplicarse de distintas formas, como por ejemplo la implementación de las mismas en la programación orientada a objetos, ya que podríamos guardar los objetos con sus atributos y poder acceder a ellos en cualquier momento, podemos eliminarlos en cualquier momento por medio de un parámetro e incluso es muy fácil su recorrido mediante un ciclo for o mayormente usado el famoso ciclo while, que puede mostrar los nodos de la lista hasta que no encuentre ninguno.

Pero no todo puede ser colores de rosa, y es que todo tiene un lado negativo o menos eficaz, existen otros tipos de estructuras más efectivas a la hora de hacer ordenamientos, resultando que es más fácil ordenar una pila o una cola que ordenar una lista enlazada, y eso se debe a los nodos, además las listas enlazadas ocupan una memoria bastante considerable y a gran escala podría afectar en cuánto a rendimiento de la aplicación, esto se debe a que cada nodo aparte de guardar información arbitraria tiene un apuntador, que como su nombre lo indica, “apunta” a otro nodo, excepto el último que ya no apunta a nada. Conociendo estas ventajas y desventajas que pueden resultar subjetivas tal vez de mi parte con mi experiencia trabajando con las mismas, por lo cuál todo lo expuesto en este ensayo no es una verdad

absoluta, puede abordarse y entenderse de diferentes formas y no es hasta que se llega a la práctica dónde ya podemos experimentar y generar nuestras propias opiniones y sensaciones con las cuales podamos contar para futuras ocasiones.

Conclusiones

- En la actualidad es muy necesario incluir en las aplicaciones herramientas que permitan ver de otra forma el flujo, en este caso hablamos de una forma gráfica, y por eso graphviz es una muy buena opción.

- Los archivos xml son universales dentro de la programación, por lo mismo los lenguajes se ven en la necesidad de implementar librerías para su lectura y escritura.

- Las listas enlazadas es una muy buena forma de almacenar información de manera ordenada, y donde cada uno de sus nodos va conectado y gracias a distintos ciclos se pueden agregar, eliminar o actualizar datos de los distintos nodos.

- ¿Las listas enlazadas son la estructura de datos más utilizada?, y si la respuesta fuera sí, ¿en que aplicaciones es donde más se utilizan?, esto es un tema para otro ensayo, pero con el contenido de este podemos darnos una idea de lo que acontece actualmente.

Referencias bibliográficas

- M.E.S College of Engineering, (2006), Principles of data Structures using C and C++.
- School of Computer Science, University of Birmingham, UK, (2018), Data Structures and Algorithms.

Anexos

- **Diagrama de clases:** Diagrama que muestra el flujo de la aplicación.

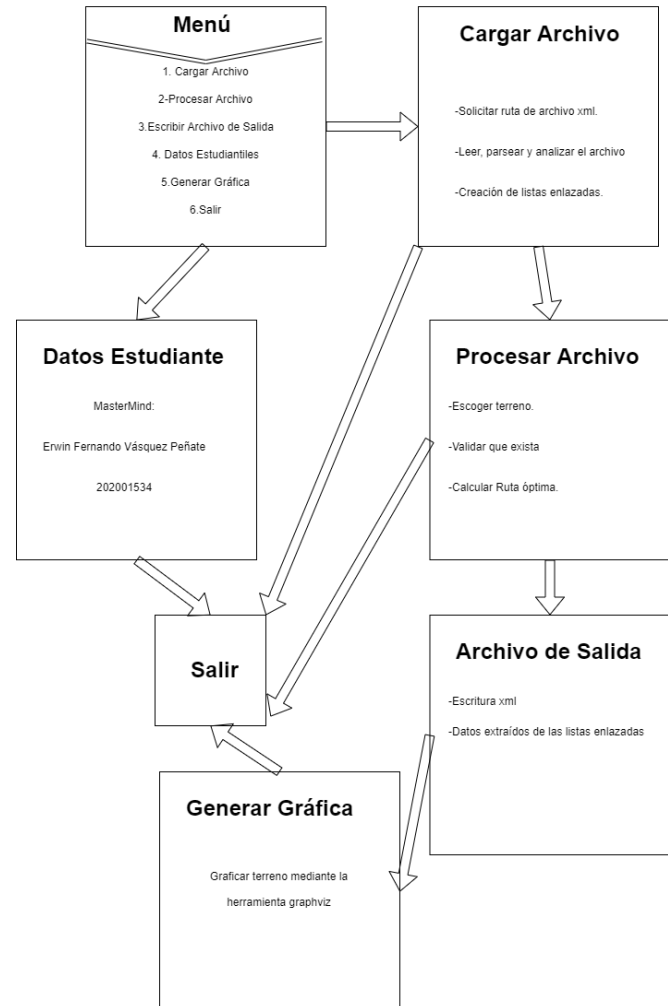


Figura 3. Diagrama de clases.

Fuente: elaboración propia, 2021.