
PROYECTO 1 – IPC2

Estudiante

201115455 – José Ernesto Pajoc Raymundo.

Resumen

El proyecto está enfocado al análisis de matrices, manejo de archivos XML y tipo de datos abstracto como lo son las listas circulares simplemente enlazadas, todo ello utilizando el lenguaje de alto nivel Python para la creación del software, donde se aplica el paradigma de la programación orientada a objetos, modular y la funcional, además se crea grafos por medio de archivos con extensión DOT para representar las matrices de salida.

El usuario es el encargado de ingresar la ruta de los archivos tanto para cargar las matrices como también para exportarlas con la extensión adecuada. El programa mostrará en pantalla paso a paso el proceso que realiza en cada etapa, para que el usuario comprenda el trabajo que realiza el software en cada momento.

Palabras clave

- Nodo
- Lista
- Grafo
- Extensión
- Lenguaje

Abstract

The project is focused on the analysis of matrices, handling of XML files and abstract data types such as simply linked circular lists, all using the high-level language Python for the creation of the software, where the programming paradigm is applied. Object-oriented, modular and functional, in addition graphs are created by means of files with DOT extension to represent the output matrices.

The user is in charge of entering the path of the files both to load the matrices and also to export them with the appropriate extension. The program will display the process carried out in each stage step by step on the screen, so that the user understands the work carried out by the software at all times.

Keywords

- Node
- List
- Graph
- Extension
- Language

Introducción

El programa que corresponde al proyecto 1 es una aplicación desarrollada en el lenguaje de programación Python, la cual cuenta con un menú principal manejado en consola donde se puede cargar un archivo con extensión XML, el cual luego puede ser procesado y reducido según un patrón de frecuencia, como también la creación de su respectivo gráfico en formato PNG basado en un archivo DOT de Graphviz.

Desarrollo del software

El lenguaje utilizado para desarrollar el software es Python en su versión 3.8.1 y el editor de texto utilizado es Visual Studio Code 1.54.1

La finalidad del programa es crear una herramienta que permita minimizar el costo total de la transmisión de datos, para ello se tiene una matriz de frecuencia de acceso creada en un archivo XML, la cual debe ser cargada al programa y luego obtener su correspondiente matriz de patrones conformada por 1 y 0, de esa forma se podrá obtener que filas poseen los mismos patrones y así agruparlas realizando la suma de sus valores, para luego tener una matriz de frecuencia reducida la cual se podrá exportar en un nuevo archivo XML y obtener su respectivo grafo.

Para crear el software se utilizó tres archivos con extensión “.py” identificados como menú, manejoXML y claseListaCircular. Los paradigmas utilizados son modular, funcional, estructurado y orientado a objetos, el tipo de dato abstracto que maneja el control de la información es una lista circular simplemente enlazada y el programa encargado de generar el grafo de las matrices es Graphviz.

El archivo “menu.py” es el encargado de mostrar el menú principal, en donde se muestra seis opciones en la cual el usuario puede seleccionar una, ingresando el número correspondiente a la opción. Para ello se utilizó un ciclo while y sentencias if para controlar el flujo de datos, es indispensable primero cargar el archivo y luego procesarlo para poder hacer uso del resto de opciones, ya que en caso contrario volverá a cargar el menú principal.

A continuación, se muestra los módulos importados y funciones utilizadas en el archivo, para el funcionamiento adecuado:

Tabla I.

Módulos utilizados en menu.py

NOMBRE	FINALIDAD
archivoXML	Crear un objeto con la ruta del archivo XML y utilizar sus métodos para realizar los procesos solicitados por el usuario.
time	Realizar una pausa al programa, para que el usuario pueda observar cada proceso y la información de cada uno.

Fuente: elaboración propia.

Tabla II.

Funciones utilizadas en menu.py

NOMBRE	FINALIDAD
separador	Mostrar una línea formada por guiones para separar cada proceso.

Fuente: elaboración propia.

El segundo archivo se identifica con el nombre “manejoXML.py” este es el encargado de leer y procesar el archivo de entrada con las matrices de frecuencia de acceso, las estructuras de control utilizadas son ciclo for y sentencia if, en este archivo se creo una clase con métodos que permite la reducción de matrices para luego almacenarla en la lista circular simplemente enlazada.

A continuación, se muestra los módulos importados y métodos de la clase archivoXML:

Tabla III.

Módulos utilizados en manejoXML.py

NOMBRE	FINALIDAD
xml.etree.ElementTree	Realizar el parseo del archivo XML para poder ser legible dentro de Python, obtener la raíz de las matrices, etiquetas y atributos.
numpy	Crear matrices con dimensiones establecidas, tomando número de columnas y filas del archivo XML.
listaCircular	Crear una lista circular vacía con el puntero cabeza apuntando a nulo para luego agregar nodos con cada matriz reducida.
time	Realizar una pausa al programa, para que el usuario pueda observar cada proceso y la información de cada uno.

Fuente: elaboración propia.

Tabla IV.

Métodos utilizados en manejoXML.py

NOMBRE	FINALIDAD
init	Utilizar el método constructor realizar el parseo del archivo XML con la ruta ingresada por el usuario.
ProcesarArchivo	Obtener el nombre y las dimensiones de cada matriz de entrada, luego recorrer cada elemento para obtener los datos, crear la matriz de frecuencia, realizar la reducción de matrices y luego enviar la información a la lista circular como un nuevo nodo.
escribirXML	Solicitar la ruta de salida para el archivo XML, realizar el llamado del método perteneciente a la lista circular para poder recorrer los nodos almacenados y exportarlos, luego mostrar en consola los nodos de la lista.
generarGrafica	Realizar el llamado de un método perteneciente a la lista circular para crear el grafo.

Fuente: elaboración propia.

Para finalizar se tiene el archivo identificado como “claseListaCircular.py” la cual se encarga de crear y recorrer los nodos de la lista circular, este es la base de todo el software ya que en esta área se encuentra la información de todas las matrices reducidas, la creación directa del archivo XML como también la creación y renderización del archivo DOT necesario para el grafo.

A continuación, se muestra los módulos importados, clases y métodos del archivo:

Tabla V.

Módulos utilizados en claseListaCircular.py

NOMBRE	FINALIDAD
io	Crear un archivo en el sistema para las matrices en el XML.
graphviz	Renderizar el archivo escrito en DOT.
time	Realizar una pausa al programa, para que el usuario pueda observar cada proceso y la información de cada uno.
Fuente: elaboración propia.	

Tabla VI.

Clases utilizados en claseListaCircular.py

NOMBRE	FINALIDAD
nodo	Crear objetos de tipo nodo con los atributos de nombre, matriz y filas únicas.
listaCircular	Definir una lista circular vacía, en la cual se agregarán los nodos correspondientes a cada matriz, estableciendo los punteros.
Fuente: elaboración propia.	

Tabla VII.

Métodos utilizados en la clase listaCircular

NOMBRE	FINALIDAD
init	Crear un objeto llamado cabeza con valor nulo.
verVacio	Retornar un valor booleano que depende de que cabeza continúe con valor nulo.
agregarFinalLista	Crear un objeto de tipo nodo, evaluar si la lista está vacía y dependiendo de ese resultado puede asignarse al objeto cabeza el valor de este nuevo nodo o se recorre toda la lista hasta encontrar al último para que apunte este elemento al nuevo nodo y el nodo apunte a la cabeza para cerrar el círculo.
recorrerLista	Mostrar todos los nodos de la lista circular con su respectivo nombre, matriz y filas repetidas.
tamanoLista	Mostrar la cantidad de nodos existentes en la lista.
crearXML	Recorrer todos los nodos y escribir un archivo plano con extensión XML que contenga la información de cada nodo en la ruta que ingrese el usuario.
crearGrafo	Recorrer todos los nodos y escribir un archivo plano con extensión DOT que contenga la información de cada nodo, para luego renderizar el archivo creado y obtener una imagen con extensión PNG.

Fuente: elaboración propia.

De esa forma es que se logra cargar, procesar y exportar la información de las matrices.

El sistema operativo que se utilizó es Windows 10 y el instalador de paquetes de Python fue pip para poder hacer uso de Graphviz y numpy.

Conclusiones

El lenguaje de Python es muy sencillo de utilizar y entender comparado con otros lenguajes de programación de alto nivel, lo cual permite enfocarse en la solución del problema.

Existe muchas fuentes de información que facilitan el uso del lenguaje Python, pero respecto a tipos de datos abstractos la mayor parte de fuentes se encuentran en inglés y no es mucha, ya que la mayoría utiliza la memoria dinámica que gestiona el mismo lenguaje, reduciendo las fuentes de información.

El lenguaje posee muchos paquetes que abarcan desde herramientas sencillas hasta complejas, lo cual ayuda a uno como desarrollador para facilitar la manipulación de archivos XML y GRAPHVIZ, con lo cual queda la inquietud de profundizar en este lenguaje de programación.

Referencias bibliográficas

- (s.f.). Obtenido de
http://facundoq.github.io/courses/images/res/03_numpy.html
- Programador clic.* (s.f.). Obtenido de
<https://programmerclick.com/article/8282763126/>
- Severance, C. (2009). *Python para informáticos*. Michigan.