|  |
| --- |
| **PROYECTO 1 – IPC 2** |
| **201800665 – Samuel Alejandro Pajoc Raymundo** |

**Resumen**

Se realizó un programa capaz de leer y procesar información contenida en un archivo de tipo xml, además, esta información fue analizada por distintos procesos, en los cuales, se especificaba que los datos de las matrices, dentro del archivo xml, fueran transformadas a matrices de patrones de acceso, las cuales contenían únicamente 1´s y 0´s. Después se debía escribir un archivo de salida de tipo xml, con la nueva matriz reducida, y a su vez se debía validar la creación de distintos grafos que representaran el flujo de los datos dentro de la matriz de salida. Estas especificaciones se encuentran dentro del archivo llamado proyecto 1.

**Palabras clave**

* Datos
* Matriz
* Ciclo
* Archivo
* Proceso

***Abstract***

*It was created a program capable of Reading and processing information conteined in a xml file, furthermore, this information was analyzed for several processes, those ones, were specified that the matrix data, into the xml file, were changed into a pattern access matrix, these ones just contained 1´s and 0’s. After that, it will be nescessary to write an xml output file, with the new reduced matrix, and moreover, it should confirm the creation of several graph´s that would represent the data flow into the output matrix. These specifications are expressed into the document called project 1.*

***Keywords***

* *Data*
* *Matrix*
* *Cycle*
* *File*
* *Process*

**Introducción**

La importancia de analizar el comportamiento de un tipo de dato especifico de distintos tamaños, como lo son las matrices, permiten que el programador pueda desarrollar un criterio muy detallado al momento de evaluar y validar cada proceso al momento de recibir el conjunto de datos contenidos en las matrices. Se tomo en cuenta que este debía ser creado con Orientación a Objetos, y se observó que cuando se trabaja con una matriz de patrones, esta presenta nuevos retos, ya que los pasos a seguir para su funcionamiento correcto, posee ciertas características que pueden llegar a confundir a un programador principiante, debido a que, si no se posee un buen control de validaciones y una visualización de los posibles errores a cometer por recorrer las filas de datos de una manera no eficiente, esto puede ocasionar fallos en el resto del programa.

Así mismo se comprendido la gran utilidad que presenta el manejo de datos dentro de archivos xml, ya que, de esta forma, cualquier base de datos comprenderá que debe hacer cuando se le presente un conjunto de datos con atributos y con especificaciones del tipo de dato al que pertenece. También se debe reconocer el valor que posee el manejo de estructuras graficas como lo fue al utilizar el lenguaje de tipo .dot, junto al entorno de graphviz, ya que por este medio se pudo generar y visualizar el flujo de los datos en el archivo de salida de la matriz reducida.

**Desarrollo del tema**

Al iniciar con el programa el usuario podrá visualizar un menú llamado “menú principal”, el cual contiene seis opciones, las cuales son: cargar archivo, procesar archivo, escribir archivo de salida, mostrar datos del estudiante, generar gráfica y salir.

Dentro de la primer opción llamada “cargar archivo”, se le solicitará que ingrese una ruta, en la cual se encuentre el archivo xml a utilizar, si se ingresa una ruta la cual no contenga un archivo del tipo xml, automáticamente se le indicará que no existe tal archivo y será devuelto al menú principal.

Ya que se estableció que se utilizaría el paradigma de objetos, esta primer opción fue creada con una clase, la cual recibe como parámetro la ruta del archivo a cargar, para luego devolver la estructura principal que será de utilidad en la segunda opción.

Tomar en cuenta que para poder ingresar a las demás opciones a excepción de la opción cinco, es necesario haber cargado y procesado el archivo, para que de esta forma pueda tener acceso hacia las demás opciones.

Después de haber cargado el archivo, es posible ingresar en la segunda opción llamada “procesar archivo”, la función de esta opción es leer y almacenar toda la información contenida dentro del archivo xml especificado en la opción anterior. Dentro de esta opción se realizan distintos procesos, los cuales transforman la matriz de entrada en una matriz de patrón, para luego buscar esos patrones, y así de esta forma conocer que filas se deben sumar, para luego poder generar una matriz reducida.

Dentro de la tercer opción llamada “escribir archivo de salida”, se encuentra una clase encargada de recibir por parámetro una lista con toda la información previamente procesada en la opción anterior, junto a la cantidad de matrices que fueron procesadas, en esta opción se le solicita al usuario que especifique una nueva ruta en la cual desea almacenar o crear un nuevo archivo xml con la nueva información. Si la ruta no existe, se le mostrará al usuario un mensaje indicándole que su ruta no existe, o se le indicará que debe especificar el formato del archivo a crear el cual debe finalizar con .xml.

Luego procede la cuarta opción llamada “datos del estudiante”, en la cual como su nombre lo indica, dentro se encuentran los datos del estudiante, esta es la única opción que no necesita validación de ningún tipo, ya que únicamente posee información del desarrollador.

A continuación prosigue la opción cinco, la cual es llamada “generar grafica”, dentro de esta opción no se le solicitará ningún dato al usuario más que la validación de que se haya procesado el archivo, luego se le mostrará de forma automática todas las graficas creadas por esta opción, cada grafica se encuentra dentro de un único archivo, el formato de estos archivos es de tipo pdf, se optó por este formato ya que posee una mayor calidad al momento de almacenar las graficas de salida generadas por el lenguaje tipo .dot,

Por ultimo se tiene la opción seis, la cual es llamada “salir”, esta opción como su nombre lo indica, finaliza con la ejecución del programa.

**Conclusiones**

Se demostró que es muy útil plantear soluciones con orientación a objetos, ya que de esta forma es más eficiente y fácil de crear nuevos elementos de tipo objeto.

Se comprendió el funcionamiento de un archivo de tipo xml al momento de haber creado la clase que permitió leerlo, así mismo se asimiló su estructuración básica al momento de escribir un archivo nuevo.

Se logró implementar la interfaz grafica de graphviz, utilizando como lenguaje principal Python, con declaraciones de tipo .dot, además se comprendió lo que es un nodo y una arista dentro de un grafo.

Se demostró nuevos métodos en los que se puede trabajar con listas enlazadas simples, las cuales pueden ayudar a crear matrices grandes y complejas, pero se aconseja descartar este método ya que es demasiado laborioso para la creación de una matriz simple, y se recomienda trabajar con las listas que permite implementar python.

**Referencias bibliográficas**

Charles Severance. Date (2013). Python para informáticos*.* Addison-Wesley Publishing Company, Inc.