
PROGRAMACION DE BRAZO MECANICO ENSAMBLADOR DE PRODUCTOS PARA LA EMPRESA DIGITAL INTELLIGENCE S. A.

201807220 – Héctor Josué Ponsoy Ayala

Resumen

El proyecto consiste en la lectura de un archivo con extensión XML. Dicho acrónimo por sus siglas en ingles significa Extensible Markup Language, el cual es un lenguaje de marcado que define un conjunto de reglas para la codificación de documentos el cual posee en su interior las líneas de ensamblaje para las cuales el robot ensamblador ira leyendo una por una, así mismo con sus componentes. Los datos leídos se almacenan en memoria dinámica con el uso de nodos de TDA, específicamente haciendo uso de las listas enlazadas simples y de las listas enlazadas dobles. Las listas enlazadas son almacenadas en dichos TDA'S y luego son analizadas con distintos algoritmos para así ser capaz de encontrar el tiempo de ensamblaje de un producto en el menor tiempo posible, reduciendo así costos y tiempo el cual a nivel empresarial representa dinero.

Palabras clave

TDA (Tipo de dato abstracto)
nodos
Lista enlazada doble
Lista enlazada simple
Matriz

Abstract

Project consists of Reading a file with XML extension. These letters mean Extensible Markup Language, which one is a markup language that defines a group of rules for code documents, this same has inside lines of assembly so assembly robot will be reading one by one, as same with their components. Read datas are storage in dynamic memory with the use of nodes of ATD, specifically making use of simply linked list, and doubly linked list. Linked lists are storage in those ATD and then they are analyzed with distinct algorithms so then be able to find the minimum time to assembly of a product, reducing costs and time, which one in a company level represents a lot of money.

Keywords

*ATD (Abstract data type)
Nodes
Doubly linked list
Simply linked list
Matrix*

Introducción

Con el uso de TDA's para almacenar las líneas de ensamblaje y componentes de un producto, y así mismo usar otro TDA para guardar otras matrices de tamaño determinado por el archivo de entrada XML, el uso de memoria dinámica resulta ser muy favorable para poder tener un mejor manejo en la memoria utilizada durante la ejecución del programa, ya que no se desperdicia memoria como en los arreglos, los cuales son estáticos y por lo tanto no se puede cambiar su posición y si se hacen de posiciones grandes y no se usan, es ahí donde hay una pérdida de memoria, la cual puede ser fácilmente sustituida por el uso de TDA's, ya que los mismos son dinámicos y se pueden almacenar datos sin tener que haber definido un límite anteriormente, lo cual a largo plazo o en proyectos grandes ayuda a una reducción enorme de memoria, así mismo se hizo un proyecto el cual era capaz de generar archivos de salida con extensión XML, y extensión .dot, lo cual hace referencia a los grafos para así visualizar las matrices de mejor manera.

Desarrollo del tema

El uso de la memoria dinámica tiene muchas ventajas, ya que la misma nos permite introducir datos sin necesidad de establecer un tamaño, lo cual nos beneficia con espacio en memoria, así mismo esta se puede ir incrementando durante la ejecución del programa. Es por eso que en dicho proyecto se pensó en el uso de una lista enlazada de listas enlazadas, por lo que para una línea de producción se creo un nodo el cual tiene como atributos el numero, cantidad de componentes, tiempo de ensamblaje, y sus referencias a anterior y siguiente, y su respectiva lista enlazada la cual tiene como principal atributo en su constructor el apuntador a inicio. Luego para un producto se creo un nodo el cual contiene su nombre, cola de elaboración, y sus apuntadores, y su lista enlazada con su apuntador.

Luego se implementó una cola, la cual fue hecha con una lista simplemente enlazada. Y por último el nodo de la Maquina, el cual posee la cantidad de líneas de producción, la lista de líneas de producción, y la lista de productos, y su respectiva lista enlazada, la cual al final se podría denominar una matriz.

Todo esto con el fin de crear la ya antes mencionada lista de lista de manera que pudiese ser fácil crear métodos para crear un XML de su ensamblaje, un método para graficar la cola de elaboración.

a. ¿Qué es una matriz?

Es un conjunto de números los cuales poseen una coordenada dando referencia a su posición, de tal manera que se forman filas y columnas.

b. ¿Qué es una lista enlazada?

Es un tipo de dato abstracto que nos permite introducir datos de forma dinámica, lo que quiere decir que esta no posee un tamaño en declarado.

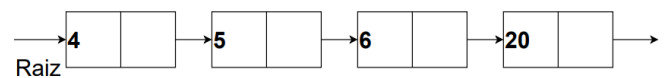


Figura 1. Forma gráfica de una lista enlazada

Fuente: elaboración propia

c. ¿Qué es un nodo?

Es un objeto creado desde una clase que esta auto referenciada. Puede ser el fragmento de un valor anterior, o de su valor siguiente, o incluso puede apuntar a None.

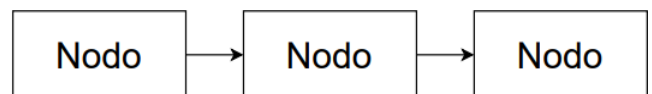
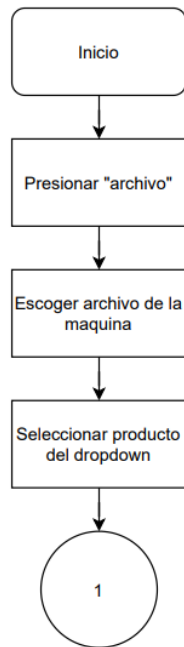


Figura 2. Forma gráfica de nodos

Fuente: elaboración propia

d. Diagramas de flujo del programa



Flujo para generar la cola de elaboración

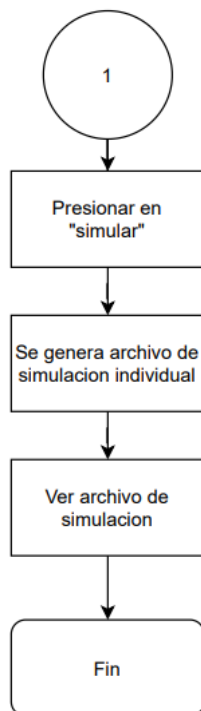
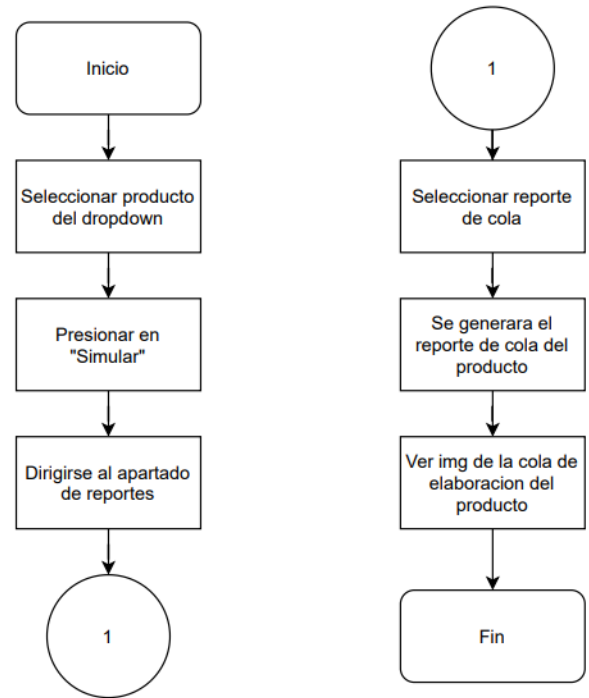


Figura 3. Diagrama de flujo del programa

Fuente: elaboración propia.

e. Uso de XML

```

import xml.etree.ElementTree as ET

root = ET.fromstring(countrydata)

# Top-Level elements
root.findall(".")

# ALL 'neighbor' grand-children of 'country' children of the top-level
# elements
root.findall("./country/neighbor")

# Nodes with name='Singapore' that have a 'year' child
root.findall("./year/..[@name='Singapore']")

# 'year' nodes that are children of nodes with name='Singapore'
root.findall("./*[@name='Singapore']/year")

# ALL 'neighbor' nodes that are the second child of their parent
root.findall("./neighbor[2]")
  
```

Figura 4. Estructura básica de eTree.

Fuente: Python Software Foundation

Conclusiones

- El uso de archivos XML, simplifica y separa datos de manera que al compartir información e intercambiar datos, hace que los datos estén disponibles más fácilmente.
- La visualización de las colas de elaboración por medio de graphviz hace que se interprete de mejor manera los objetos que se almacena en la lista, así como los atributos del objeto.
- El uso de listas enlazadas en el proyecto hace que el uso de memoria se adapte dinámicamente al número de datos almacenados en la lista en cada momento.

Referencias bibliográficas

- Documentation. (s. f.). graphviz.org. Recuperado 7 de marzo de 2021, de <https://graphviz.org/documentation/> Robinson, S. (s. f.). Reading and Writing XML Files in
- Python. Stack Abuse. Recuperado 7 de marzo de 2021, de <https://stackabuse.com/reading-and-writing-xmlfilesin-python/>

Anexos

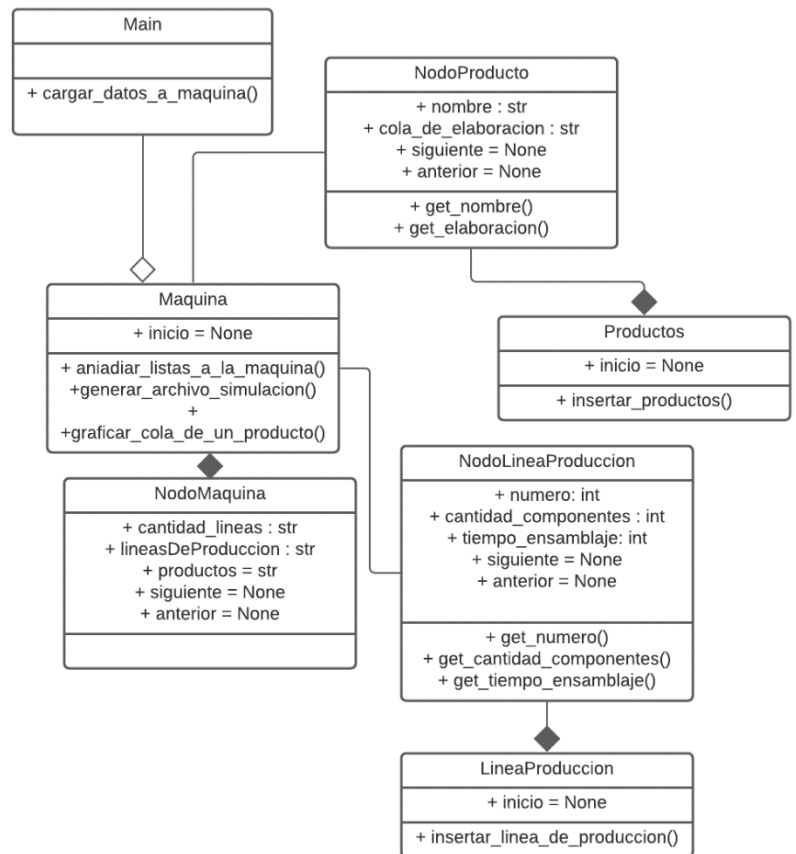


Figura 5. Diagrama de clases

Fuente: elaboración propia.