

---

## BRAZO ROBÓTICO ENSAMBLADOR DE PRODUCTOS

---

201906558 – Eddy Fernando Díaz Galindo

### Resumen

Se realizó la implementación de un código óptimo para el brazo robótico, dicho brazo puede ensamblar componentes, para ello se utilizó interfaz gráfica para mejor experiencia de usuario y se analizó para armar el producto; se tuvo varias consideraciones que son el tiempo de movimiento que hay en cada línea de producción. Debido a que el robot solo se puede mover hacia arriba y hacia abajo se utilizó expresiones condicionales de Python y listas doblemente enlazadas.

Se utilizó programación orientada a objetos (POO), para modelar un programa que separa el problema en problemas más pequeños para tener mejores resultados, al utilizar este paradigma se analizó la mayoría de los conflictos que puede llegar a tener el brazo robótico.

También para que el usuario tenga mejor experiencia con el programa, se utilizó el software Graphviz para representar gráficos/grafos de la información de la secuencia de trabajo. Si el usuario quiere tener la simulación que tendrá el brazo debe utilizar el metalenguaje XML.

### Palabras clave

Python, XML, Graphviz, tkinter, Grafos, nodos.

### Abstract

*The implementation of an optimal code for the robotic arm was carried out, said arm can assemble components, for this a graphical interface was used for a better user experience and it was analyzed to assemble the product; There were several considerations that are the movement time that there is in each production line. Because the robot can only move up and down we used Python conditional expressions and doubly linked lists.*

*Object-oriented programming (OOP) was used to model a program that separates the problem into smaller problems to obtain better results. By using this paradigm, most of the conflicts that the robot may have were analyzed.*

*Also for the user to have a better experience with the program, the Graphviz software was used to represent graphs / graphs of the information of the work sequence. If the user wants to have the simulation that the arm will have, he must use the XML metalanguage.*

### Keywords

*Python, XML, Graphviz, tkinter, graphs, nodes.*

## Introducción

Al analizar matemáticamente las elaboraciones, se utilizó expresiones regulares para analizar las elaboraciones que tiene el archivo xml para luego establecer un algoritmo que al analizar dicha elaboración pueda llegar también a analizar elaboraciones de mayor tamaño, según este análisis se encontraron más problemas que al utilizar programación orientada a objetos se obtuvo que separar cada línea producción que hace referencia a cada componente utilizando listas de listas.

Los nodos fueron analizados según la posición en la que se encuentra en el componente, tomando en cuenta que el brazo al inicio esta en el primer componente de cada línea, se crearon dos apuntadores para analizar si el brazo debe bajar o subir o quedarse donde esta; este algoritmo se soluciono con desigualdades.

## Desarrollo del tema

Se utilizó archivos XML un metalenguaje que define la sintaxis utilizada para definir etiquetas estructurados. Para que la modificación y la eliminación del archivo sean relativamente simples mediante programación en Python, este tiene una biblioteca incorporada ElementTree, que tiene funciones para leer y manipular XML. Con esta librería se extraen los datos para luego introducirla a la respectiva lista que fueron creadas.

Una lista es una estructura dinámica de datos que contiene una colección de elementos homogéneos (del mismo tipo) de manera que se establece entre ellos un orden. Es decir, cada elemento, menos el primero, tiene un predecesor, y cada elemento, menos el último, tiene un sucesor.

Podemos distinguir, atendiendo a la organización de los nodos, entre:

Listas simplemente enlazadas: cada nodo tiene un apuntador que apunta al siguiente nodo.

Listas doblemente enlazadas: cada nodo dispone de un apuntador que apunta al siguiente nodo, y otro que apunta al nodo anterior.

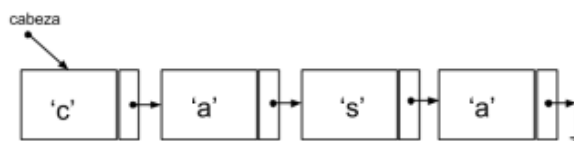


Figura 1. Lista simplemente enlazada.

Fuente: José Fager, W. Libardo Pantoja Yépez, 2014 y pag. 132.

El término estructura de datos se utiliza para referirse a una forma de organizar un conjunto de datos que se relacionan entre sí, sean estos simples o estructurados, con el objetivo de facilitar su manipulación y de operarlo como un todo.

Utilizando este tipo de estructura de datos se necesitó utilizar tres listas simplemente enlazadas, que contienen:

- Lista Doblemente enlazada de componentes: Contiene la información de cada componente que tiene las líneas de producción, en esta lista esta el algoritmo principal del simula.
- Lista Simplemente enlazada de contenido: Contiene la información que se utiliza para llenar la tabla que se muestra en la interfaz gráfica.
- Lista Simplemente enlazada de elaboraciones: Contiene la información de

todas las elaboraciones que tiene cada producto.

- **Lista Simplemente enlazada líneas:** Contiene la información de todas las líneas de producción que tendrá el brazo.
- **Lista Simplemente enlazada nombre de producciones:** Contiene la información de todos los nombres de los productos que se mostraran en el combobox.

Sobre una estructura de datos se puede efectuar diferentes tipos de operaciones, entre las mas importantes y que fue utilizada en la realización de la solución

- **Insertión:** Es aquella mediante la cual se incluye un nuevo elemento en la estructura.
- **Modificación:** Permite variar parcial o totalmente el contenido de la información de los elementos de la estructura.
- **Eliminación:** Como su nombre lo indica, es la que permite suprimir elementos de la estructura
- **Búsqueda:** Permite determinar si un elemento se encuentra o no en la estructura.
- **Consulta de la información:** Permite obtener información de uno o más elementos de la estructura.

Algunas de estas listas además están enlazadas entre sí, la lista de componentes depende de la lista de líneas; para que funcionara correctamente el algoritmo de la simulación, primero el usuario debe de ingresar el nombre del producto para tener las líneas de producción, con este dato se muestran las líneas de producción que hace referencia a cada componente.

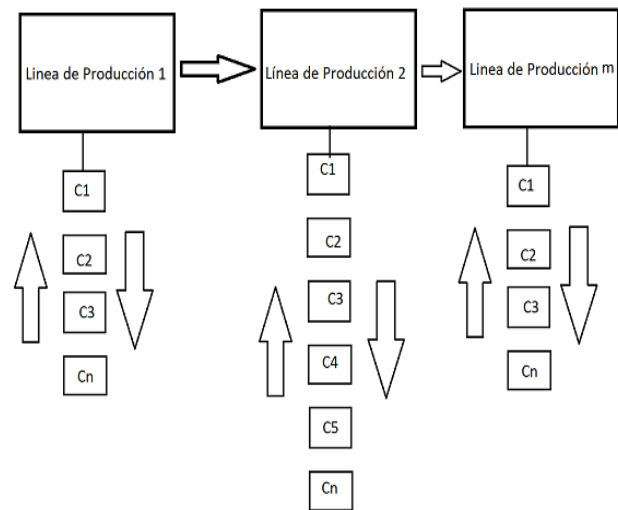


Figura 11. Listas dependiente de otra lista.

Fuente: Elaboración propia, 2021

El análisis que tuvo la lista doblemente enlazada de componentes tuvo como prioridad buscar primeramente la posición inicial del donde se encontraba el brazo robotico, pero, la mayor prioridad que se puede notar si se lee el código, fueron los métodos de recorrer la lista que contiene el componente para crear un método que pueda armar el brazo.

Tkinter es un binding de la biblioteca gráfica Tcl/Tk para el lenguaje de programación Python, con estos queremos decir que Tk se encuentra disponible para varios lenguajes de programación entre los cuales se encuentra Python con el nombre de Tkinter. Este no es mas que una capa de esta librería para el lenguaje Python con lo cual usar Tk en otro lenguaje no nos supondrá un inconveniente.

Se considera un estándar para la interfaz gráfica de usuario (GUI) para Python y es el que viene por defecto con la instalación para Microsoft Windows y preinstalado en la mayoría de las distribuciones de GNU/Linux. Con Tkinter podremos conseguir resultados casi tan buenos

como con otras librerías gráficas siempre teniendo en cuenta que quizás con otras herramientas podamos realizar trabajos mas complejos donde necesitemos una plataforma mas robusta, pero como herramienta didáctica e interfaces sencillas nos sobra, dándonos una perspectiva de lo que se trata el desarrollo de una parte muy importante de una aplicación si deseamos distribuirla. Gracias a Tkinter veremos como interactuar con el usuario pidiéndole el ingreso de datos, capturando la pulsación de teclas, movimientos del mouse, entre algunas de las cosas que podremos lograr.

**El Hardware.** El procesador es lo que más se toma en cuenta ya que el programa tiene muchas sentencias de control y también la frecuencia de trabajo, memoria, discos, etc.

**El Software.** Puede variar dependiendo del sistema operativo y hasta lenguaje de programación, pero tenemos que recordar que Python es un lenguaje de alto nivel.

El XML de salida que fue creado contiene el nombre del terreno que se está evaluando en la lista de terrenos, este contiene la suma del combustible de la ruta óptima.

### **Graphviz:**

La intención del diseño original es realizar un diseño automático de gráficos dirigidos / gráficos no dirigidos para este caso se utiliza una tabla que simula un terreno, al utilizar esta herramienta debemos tener bien definidos como funciona este sub-lenguaje

### **EXPRESIONES REGULARES**

Debido a que la estructura de las elaboraciones tiene que ser analizado se opto por utilizar expresiones regulares.

Las expresiones regulares permiten comprobar si una cadena de texto se ajusta a un determinado

tipo de estructura o patrón. Utilizan un lenguaje de signos propio, y permiten también buscar o remplazar texto, no sólo un texto concreto sino un tipo de texto que se ajuste a un determinado patrón.

Este algoritmo comparado con otros algoritmos puede tener mejores o peores factores que influyen en la eficiencia de un algoritmo, pero tomas en cuenta factores que sean externos al algoritmo como la computadora donde se ejecuta

### **Caracteres especiales**

Podemos incluir también una serie de caracteres especiales que indican si la cadena tiene un determinado tipo de caracteres.

`^w/` : Contiene caracteres alfanuméricos

`^W/` : devuelve false si sólo contiene caracteres alfanuméricos.

`^d/` : Contiene dígitos o cifras

`^D/` : devuelve false si sólo contiene dígitos.

`^s/` : Contiene alguno de los caracteres que definen espacios en blanco (espacio, tabulador, retorno de carro ... )

`^S/` devuelve false si sólo contiene caracteres que definen espacios en blanco.

### **Conclusiones**

Los simuladores son recursos tecnológicos que permiten aprender de manera práctica, rápida y visual de lo que hace una máquina, recrea entornos reales en virtuales, para el brazo robótico este simulador puede ser utilizado cuando este a la venta.

### **Referencias bibliográficas**

José Fager, (2014). *Estructura de datos*. Proyecto LATIn.

## Extensión:

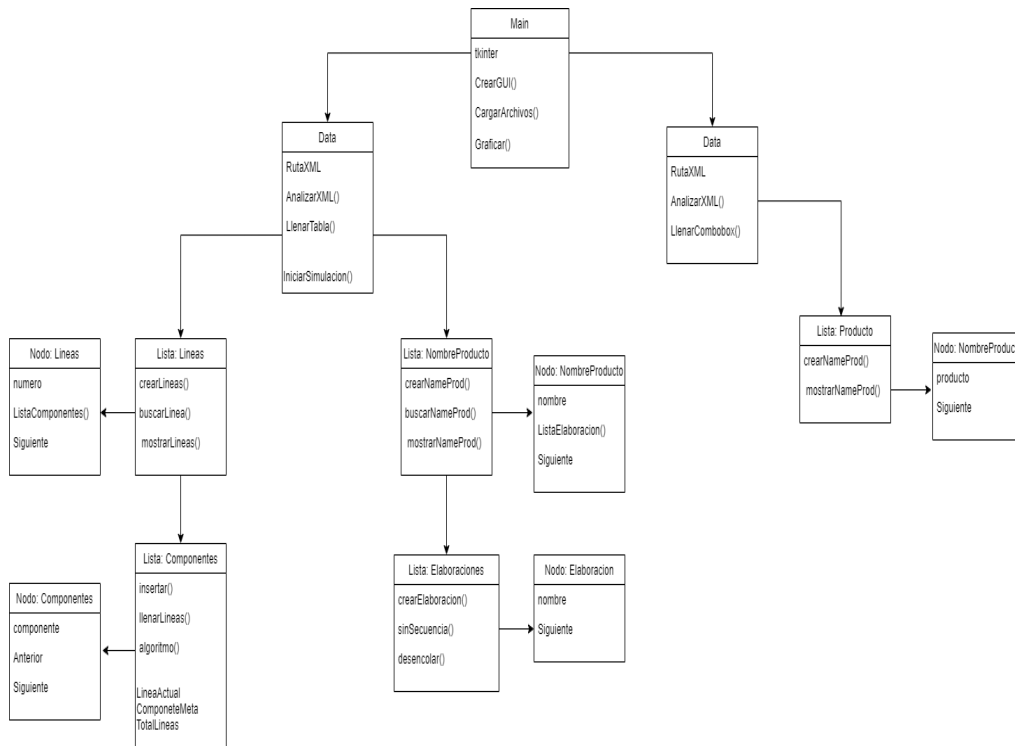


Figura III. Diagrama de Clases.

Fuente: Elaboración propia, 2021