
PROYECTO 2

202106003 –Jhonatan Alexander Aguilar Reyes

Resumen

Dicho proyecto fue propuesto con la finalidad de crear un sistema de creación de nuevos materiales mediante la unificación de diferentes elementos químicos, dando lugar a compuestos nuevos

La aplicación sera capaz de crear nuevos compuestos químicos mediante la unificación de los elementos que conformaran dicho compuesto, todo este trabajo es realizado por un listado de maquinas las cuales tiene “n” cantidades de pines, los pines serán la rama que tendrán varios elementos químicos con los que la maquina podrá trabajar, los elementos químicos no pueden repetirse entre sus pines ya que esto arruinaría el sistema con el que opera la maquina y también la optimización con la que trabaja. Las listas de maquinas con sus respectivos pines serán recolectadas de un documento xml así también los compuestos químicos que estas maquinas deben procesar. Al finalizar los procesos se mostrar los pasos realizados por las maquinas para generar el compuesto y el tiempo en que lo realizaron, en caso la maquina sea capaz de operar dicho compuesto.

Palabras clave

Compuesto: Sustancia formada por la combinacion quimica de dos o mas elementos de la tabla periodica

Elemento: Son un tipo de materia formada por átomos de la misma clase

Unificación: Hacer que varios elementos se conviertan en la combinación de uno solo

Maquina: Mecanismos que se utilizan para facilitar o realizar un trabajo determinado

Abstract

This project was proposed with the purpose of creating a system for the creation of new materials through the unification of different chemical elements, giving rise to new compounds.

The application will be able to create new chemical compounds by unifying the elements that will make up said compound, all this work is done by a list of machines which have "n" amounts of pins, the pins will be the branch that will have various chemical elements with which the machine will be able to work, the chemical elements cannot be repeated between its pins since this would ruin the system with which the machine operates and also the optimization with which it works. The lists of machines with their respective pins will be collected from an xml document as well as the chemical compounds that these machines must process. At the end of the processes, the steps carried out by the machines to generate the compound and the time in

which they carried it out will be shown, if the machine is capable of operating said compound.

Keywords

Compound: *Substance formed by the chemical combination of two or more elements of the periodic table*

Machine: *Mechanisms that are used to facilitate or perform a specific job*

Unification: *Make several elements become the combination of one single*

Element: *They are a type of matter formed by atoms of the same class*

Introducción

El desarrollo de la aplicación se desarrollo con la ayuda de diferentes paradigmas de programación y el uso de estructuras de datos. Los paradigmas de programación implementados fueron programación orientada a objetos, programación funcional y modular para desarrollar de manera ordenada las diferentes funcionalidades las cuales en algunos casos eran dependientes de otras.

La carga de los datos fue establecida al formato XML (lenguaje de Marcado Extensible) donde almacenada los datos de los elementos, las listas de las maquinas las cuales contenían una lista de pines y esta lista de pines contenían una lista de elementos, también se recolecto una lista de compuestos la cual contenía otra lista donde se indicaban los elementos que se debían operar para

formar dicho compuesto. Se realizaron suficientes validaciones las cuales permiten ingresar x cantidad de archivos xml sin riesgo de romper el programa o tener datos repetidos.

Para la realizar la simulación, primero se muestra al cliente los compuestos disponibles, despues de que el cliente seleccione un compuesto se procede a analizarlo en cada maquina, para terminar el tiempo que tarda cada una o si esta no es compatible con dicho compuesto, teniendo esto listo se procede a llevar la informacion a la lista de pines en donde se crear listas auxiliares para poder guardar datos importantes para poder realizar las simulacion y lo mas importante poder utilizar correctamente la lista doblemente enlazada.

Desarrollo del tema

La aplicación fue desarrollada con el lenguaje de programacion “python” debido a su gran facilidad de sintaxis y manejo de variables, Ademas, implementando librerias integradas del propio lenguaje.

A. Directorios y Modulos

Cada proyecto y aplicación debe contener cierta jerarquia y organización para un manejo ordenado de la informacion, en este caso un buen manejo de la funcionalidad y datos globales.

- **Proyecto2:** El primer directorio de la jerarquia el cual contiene toda la aplicación
 - **TDA:** Es el primer directorio el cual contiene todas las listas de datos abstractos con las que el programa funciona.

- **Data.py**
 - **ListaCompuestos.py**
 - **ListaElementos.py**
 - **listaElementostrabajo.py**
 - **ListaInstrucciones.py**
 - **ListaMaquinas.py**
 - **ListaPines.py**
 - **ListaElementCompuesto.py**
 - **Nodo.py**
- **Documentacion:** Es el directorio en donde se almacenan los documentos xml y los documentos que creara el programa durante su ejecucion como la grafica de instrucciones y la grafica de las maquinas disponibles cargadas en la aplicación.
 - **Main.py:** En esta clase se encuentra todo lo necesario para iniciar el programa, la interfaz creada con “tkinter” y todas las funciones que le dan vida a la aplicación, todas estas funciones dependen de las listas TDA ya que de aquí es donde obtienen todos los datos necesarios.

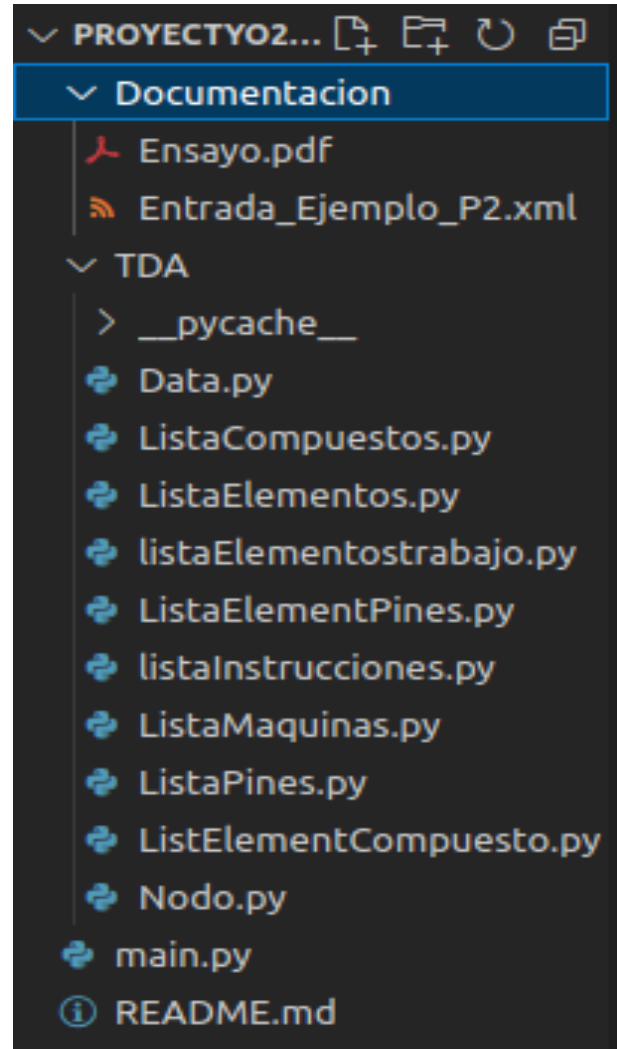


Figura 1. Directorios y Modulos.

Fuente: elaboración propia, 2022.

B. Clases

Al implementar distintos paradigmas de programación como programación orientada a objetos, se implementaron distintas clases para crear objetos que se complementaran entre si y fueran almacenados en instancias de listas simples enlazadas o doblemente enlazadas. Ver diagrama de clases en apartado “Anexos”.

C. Lista simple enlazadas

Para la simulación de una cola, se utilizó una lista simple enlazada la cual contenía diferentes métodos para darle la funcionalidad necesaria al programa, un ejemplo de estos métodos sería el método que retornaba el primer elemento de la lista, o el método que retornaba la lista de pines dependiendo de la máquina que reciba como parámetro, el método con el que se creaba el dibujo de las máquinas disponibles, el método donde se ordenaban los elementos por número atómico, el método donde se generaba la lista de instrucciones, el método donde se genera la gráfica del listado de instrucciones, entre muchos otros. La mayor parte del programa se estuvo manipulando con listas simples, pero hay una parte en donde se necesitó la implementación de una lista doblemente enlazada para poder retroceder o avanzar entre nodos según la necesidad del programa.

D. Módulos

La aplicación está dividida en diferentes directorios que cuentan con diferentes módulos que corresponden a una tarea específica. Como primer punto tenemos al módulo principal que controla los demás módulos.

- Módulo: main.py

Como primer Módulo se encuentra el que controla toda la funcionalidad de nuestro programa ya que en este módulo están todas las funciones que se llaman al momento de presionar cualquier botón de la interfaz gráfica

```
> def AbrirArchivo(): ...  
  
> def GenerarXML(): ...  
#MENU GESTION ELEMENTOS  
> def ListElementos(): ...  
  
> def AgregarElemento(): ...  
> def VentanaAgregarElemento(): ...  
  
#MENU GESTION COMPUESTOS  
> def ListaCompuestosVentana(): ...  
  
> def VentanaAnalizarCompuesto(): ...  
  
> def VentanaMquina(): ...  
  
v3 = pdf.ShowPdf()  
v4 = pdf.ShowPdf()  
  
> def VerInstrucciones(): ...  
  
#MENU GESTION DE MAQUINAS  
v1 = pdf.ShowPdf()  
v2 = pdf.ShowPdf()  
  
> def graficaMaquinas(): ...  
  
#TEMAS DE AYUDA  
> def ayuda(): ...
```

Figura 2. Funciones para Interfaz.

Fuente: elaboración propia, 2022.

- En la primera función (Abrir xml) se encuentra toda la lógica que se implementó para poder recolectar la información del documento xml. Toda esta información se almacena en variables globales para poder acceder a ellas desde cualquier función o parte del programa.
- Una de las funciones más complicadas de desarrollar fue la llamada “Ver Instrucciones” en esta función es donde se recolecta el

compuesto a analizar y se le pasa a todas las maquinas para que estas lleven esa informacion a sus pines y realicen todos los procesos necesarios con sus listas simples y la lista doblemente enlazada y luego con los datos obtenidos que realice la grafica de los procesos por cada segundo de proceso.

Conclusiones

El propósito de este proyecto es leer, generar e interpretar de manera correcta archivos con el formato XML, cumplir la necesidad que el programa solicita mediante el uso de todas las herramientas que la programación ofrece como los paradigmas de programación.

Dicho proyecto busca fortalecer los conocimientos de como las estructuras de datos funcionan ya que estas son la base de como se construyen las diferentes listas nativas de un lenguaje de programación y aplicar conceptos matemáticos y químicos para poder adaptar una herramienta a cualquier necesidad

Por ultimo, hacer uso de la herramienta graphviz para la generacion de grafos o en este caso listas enlazadas y sus derivados

Referencias bibliográficas

The Graphviz Authors. (2022).
Documentation. Graphviz. Recuperado 6 de septiembre de 2022, de
<https://graphviz.org/documentation/>

A linked list program using one class. (2020, 19 septiembre). Stack Overflow. Recuperado 6 de septiembre de 2022, de

<https://stackoverflow.com/questions/63973503/a-linked-list-program-using-one-class>

xml.etree.ElementTree — The ElementTree XML API — Python 3.10.6 documentation. (s. f.). Recuperado 6 de septiembre de 2022, de
<https://docs.python.org/3/library/xml.etree.elementtree.html>

ANEXOS

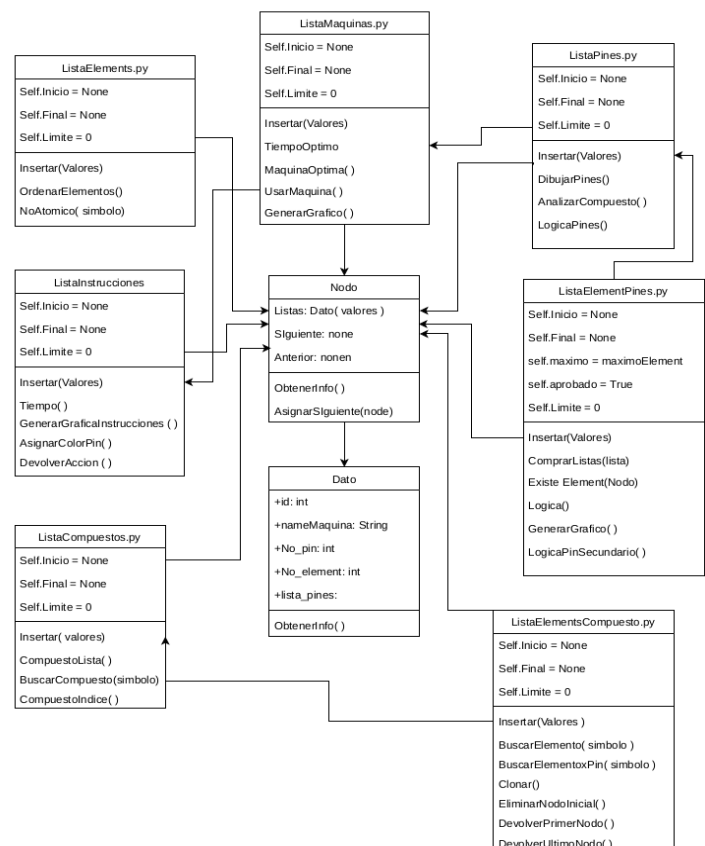


Figura 3. Diagrama de clases.

Fuente: elaboración propia, 2022.