
Proyecto 2 – Introducción a la Programación y Computación 2

202103105 – Luis Daniel Castellanos Betancourt

Resumen

El siguiente proyecto consiste en la creación de un programa desarrollado en Python, el cual organiza una serie de Elementos, maquinas y compuestos químicos.

Los compuestos químicos pueden ser creados con las maquinas, las cuales toman un tiempo determinado para cada compuesto. El programa es capaz de ordenar los datos anteriores mencionados utilizando listas dobles.

Una vez se puedan ver el tiempo que tarda cada maquina en crear un compuesto, estos se pueden exportar utilizando Graphviz. También se puede generar una salida en XML que indica cada paso que se tomó para la creación del compuesto

Palabras clave

Python, Software de computadora, Elementos químicos, Compuestos químicos

Abstract

This project consists in the creation of a computer program developed in Python. This program organizes a series of elements, machines and chemical compounds

These compounds can be created using the machines. The machines take a determined number of seconds to create each compound. The program is capable of organizing the aforementioned data using doubly linked lists.

Once viewed the times of creation of each compound, these can be exported using Graphviz. There's also the option of an XML output that prints out each step during the creation

Keywords

Python, computer software, Chemical Elements, Chemical Compounds

Introducción

El principal objetivo de este proyecto es crear distintos tipos de funciones sin utilizar la arquitectura de datos que proporciona Python, como por ejemplo las listas, tuplas, diccionarios, etc.

También se desarrollaron varios algoritmos que ordenan los componentes químicos según como sea la máquina. El objetivo de estos algoritmos es que sean óptimos y que generen el tiempo mas corto para generar cada componente

En conjunto a esto se usó la solución de interfaz grafica de Tkinter, para elaborar todas las ventanas, información de los elementos, compuestos y maquinas

Uso del programa

a. Interfaz

La interfaz se desarrolló, como se mencionó anteriormente, con Tkinter. Es una interfaz sencilla que se creó con la facilidad de uso en mente.

Las tipografías que se usaron para la creación de la interfaz grafica son Courier New para los títulos, para los subtítulos Futura, y para las tablas Calibri

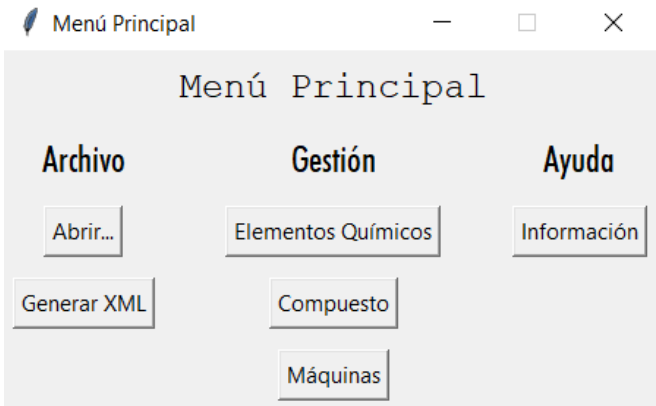


Figura 1. Menú principal del programa.

Fuente: elaboración propia.

La información múltiple se organizo en una serie de tablas, las cuales se abren en distintas ventanas para mantener un orden organizado

Número Atómico	Símbolo	Nombre del Elemento
1	H	Hidrogeno
2	He	Helio
3	Li	Litio
4	Be	Berilio
6	C	Carbono
7	N	Nitrogeno

Figura 2. Ejemplo de tablas

Fuente: elaboración propia.

b. Uso del programa

El programa toma por entrada un archivo XML el cual contiene la información de los elementos químicos, las Maquinas que contiene (con los pines que usa cada maquina), y los compuestos químicos.

El usuario puede añadir elementos químicos si lo desea utilizando la función incluida en la interfaz gráfica. Este debe ingresar los datos de forma correcta, de otro modo el programa arrojará un error



Figura 3. Añadiendo elementos químicos

Fuente: elaboración propia.

De la misma forma que los elementos, el usuario puede visualizar de la misma forma las maquinas y sus pines, y los compuestos ordenados de forma lineal

En las secciones de ayuda se incluye el link del repositorio, y el nombre del Alumno.

Manual Técnico

Para la creación del programa se uso el paradigma de programación orientado a objetos. Se crearon clases para cada componente crucial del programa como Maquina, Compuesto, Analizador y también una clase que se dedica únicamente al análisis de la creación de compuestos

a. Lista doble

Se creo una lista doble, con métodos específicos para los objetos de tipo Elemento, Pin

b. Maquina, Pin, Elemento y Compuesto

Estos objetos contienen información sobre si mismos, como el numero de pines, elemento químico, etc y una lista doble la cual contiene el conjunto de elementos que lo componen

```
class Maquina:
    def __init__(self, nombre, numPines, numElementos, listaPines):
        self.nombre = nombre
        self.numPines = numPines
        self.numElementos = numElementos
        self.listaPines = listaPines
```

Figura 4. Variables de una clase

Fuente: elaboración propia.

c. Analizador

El analizador consiste en un algoritmo el cual averigua cual es el tiempo optimo para la optimización del tiempo de creacion de un compuesto dada una maquina y un compuesto

Este analizador puede retornar un número, y a su vez puede crear una tabla utilizando Graphviz, o una salida XML, la cual se utiliza posteriormente en otra función para la creación de elementos XML

```
def TiempoParaCrearCompuesto(listaMaquina, listaCompuesto, generarTabla, generarXML):
```

d. Menú Principal

El menú principal es donde se contiene la creación de Ventanas utilizando Tkinter. En esta clase es donde se realiza la ejecución principal del programa, la lectura de XML, la salida del XML, y el almacenamiento de la información temporal

e. Lectura XML

Para la lectura XML se utilizó la herramienta Minidom, la cual tiene una serie de métodos que facilita la lectura de los archivos utilizando NodeLists

a. Salida XML

Para la salida de datos en XML, se utiliza un archivo el cual contiene la información obtenida del método utilizado anteriormente para ver el tiempo óptimo de cada máquina para crear un componente.

El XML se generará en la raíz donde se encuentre el programa

```
<?xml version="1.0"?>
<RESPUESTA>
  <listaCompuestos>
    <compuesto>
      <nombreMaquina>Maquina 1</nombreMaquina>
      <tiempoOptimo>7</tiempoOptimo>
      <instrucciones>
        <tiempo>
          <numeroSegundo>1</numeroSegundo>
          <acciones>
            <accionPin>
              <numeroPin>1</numeroPin>
              <accion>Mover Adelante</accion>
            </accionPin>
            <accionPin>
              <numeroPin>2</numeroPin>
              <accion>Mover Adelante</accion>
            </accionPin>
          </acciones>
        </tiempo>
        <tiempo>
          <numeroSegundo>2</numeroSegundo>
          <acciones>
            <accionPin>
              <numeroPin>1</numeroPin>
              <accion>Fusionar Li</accion>
            </accionPin>
            <accionPin>
              <numeroPin>2</numeroPin>
              <accion>Mover Adelante</accion>
            </accionPin>
          </acciones>
        </tiempo>
      </instrucciones>
    </compuesto>
  </listaCompuestos>
</RESPUESTA>
```

Figura 5. Ejemplo de salida XML

Fuente: elaboración propia.

Conclusiones

Se logro utilizar exitosamente un método de listas doblemente enlazadas para poder generar una solución de software al problema dado

También se logro generar un algoritmo estable el cual genera el tiempo óptimo para la generación de compuestos, utilizando el sistema de reglas de cada maquina

Se implementó Graphviz exitosamente para crear tablas para ver los pasos que toma cada maquina en crear un compuesto

Referencias bibliográficas

Máximo 5 referencias en orden alfabético.

Python Software Foundation, (2023). *Minimal DOM implementation*.

<https://docs.python.org/3/library/xml.dom.minidom.html>

Anexos

Diagrama de Clases | Proyecto 2

