

PROYECTO 2

202110509 – Mario Ernesto Marroquín Pérez

Resumen

El Laboratorio Avanzado de Física de la Universidad de San Carlos de Guatemala (LAF-USAC) ha desarrollado tecnología capaz de crear nuevos materiales, denominados compuestos, a partir de los elementos químicos básicos que se encuentran en la naturaleza.

La tecnología desarrollada por el LAF-USAC consiste en una máquina rectangular compuesta por “N” pines capaces de colocarse en “M” posiciones, donde cada posición contiene un elemento químico básico.

Para la solución a esta problemática, se desarrolló un software capaz de transmitir las instrucciones óptimas para que la máquina pueda elaborar nuevos compuestos. Se la creación de dicho software se utilizó el lenguaje de programación Python, así mismo; el uso de estructuras de datos, listas enlazadas y Tipos de Dato Abstracto (TDAs).

Palabras clave

TDAs: estructura de datos que define un conjunto de valores y las operaciones que se pueden realizar en esos valores.

XML: Extensible Markup Lenguaje por sus siglas en inglés, es un lenguaje de marcado que define un conjunto de reglas para la codificación de documentos.

Abstract

The Advanced Physics Laboratory of the University of San Carlos de Guatemala (LAF-USAC) has developed technology capable of creating new materials, called composites, from the basic chemical elements found in nature.

The technology developed by LAF-USAC consists of a rectangular machine composed of "N" pins capable of being placed in "M" positions, where each position contains a basic chemical element.

In order to solve this problem, a software capable of transmitting the optimal instructions for the machine to elaborate new compounds was developed. Python programming language was used to create this software, as well as the use of data structures, linked lists and Abstract Data Types (ADTs).

Keywords

TDAs: data structure that defines a set of values and the operations that can be performed on those values.

XML: Extensible Markup Language is a markup language that defines a set of rules for document encoding.

Introducción

Los TDAs son una forma de abstraer los detalles internos de una estructura de datos y proporcionar una interfaz clara y bien definida para trabajar con ella. Por ejemplo, un TDA puede definir una estructura de datos de cola, que permite agregar elementos al final y eliminar elementos del principio, pero oculta los detalles internos de cómo se implementa la cola.

Los TDAs son importantes porque permiten a los programadores trabajar con estructuras de datos complejas de manera más fácil y segura. Al proporcionar una interfaz bien definida y ocultar los detalles internos de la implementación, los TDAs pueden facilitar la escritura de código más claro y conciso, y reducir la posibilidad de errores y bugs.

Con ayuda de la Programación Orientada a Objetos se identificaron las características de cada elemento.

Desarrollo del tema

a. Descripción del Problema

El Laboratorio Avanzado de Física de la Universidad de San Carlos de Guatemala (LAF-USAC) ha desarrollado tecnología capaz de crear nuevos materiales, denominados compuestos, a partir de los elementos químicos básicos que se encuentran en la naturaleza.

La tabla periódica de los elementos químicos presenta todos los elementos químicos básicos que se encuentran en la naturaleza, en la figura No. 1, se presenta una muestra de la información que se maneja por cada elemento químico básico.

Elementos Químicos		
Número atómico	Símbolo	Elemento
1	H	Hidrógeno
2	He	Helio
3	Li	Litio
4	Be	Berilio
5	Be	Boro
6	C	Carbono
7	N	Nitrógeno

Figura 1. Ejemplo de una sección de la tabla periódica de elementos químicos.

Fuente: Enunciado Proyecto 2 IPC2, 2023.

La tecnología desarrollada por el LAF-USAC consiste en una máquina rectangular compuesta por “N” pines capaces de colocarse en “M” posiciones, donde cada posición contiene un elemento químico básico, la figura 2 ilustra una de estas máquinas.

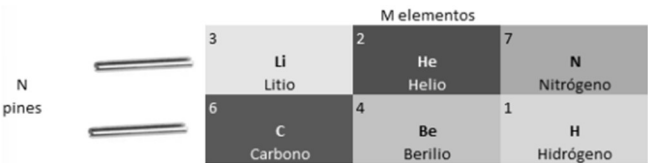


Figura 2. Máquina de N pines por M elementos para crear nuevos compuestos.

Fuente: Enunciado Proyecto 2 IPC2, 2023.

b. Conceptos Efectuados en la Creación del Programa.

Tipos de Dato Abstracto (TDAs):

Los TDAs son una técnica que permite simplificar el manejo de una estructura de datos compleja mediante la creación de una interfaz clara y precisa para trabajar con ella. Por ejemplo, un TDA de cola puede ser utilizado para agregar y eliminar elementos de

manera sencilla, sin necesidad de conocer los detalles técnicos de cómo se realiza la cola internamente.

Los TDAs se basan en el concepto de abstracción, que significa ocultar los detalles de implementación de una estructura de datos detrás de una interfaz clara y bien definida que permita trabajar con ella de manera sencilla y segura. La idea es proporcionar un conjunto de operaciones que se pueden realizar en los datos almacenados en la estructura, sin que el usuario tenga que preocuparse por cómo se realizan estas operaciones internamente.

Los TDAs pueden implementarse en cualquier lenguaje de programación y se utilizan comúnmente en la programación orientada a objetos. Por ejemplo, en Java, los TDAs se implementan mediante la definición de una clase que representa la estructura de datos y proporciona métodos para acceder y manipular los datos.

Además, los TDAs se pueden clasificar según el tipo de datos que almacenan y las operaciones que permiten. Por ejemplo, algunos ejemplos de TDAs comunes incluyen:

- Listas: una estructura de datos que permite agregar y eliminar elementos en cualquier posición.
- Pilas: una estructura de datos que permite agregar y eliminar elementos solo en el extremo superior.
- Colas: una estructura de datos que permite agregar elementos al final y eliminar elementos del principio.
- Árboles: una estructura de datos jerárquica que se utiliza para representar relaciones entre elementos.

- Grafos: una estructura de datos que representa relaciones entre elementos no necesariamente jerárquicas.

Los TDAs son importantes porque permiten a los programadores trabajar con estructuras de datos complejas de manera más fácil y segura. Al proporcionar una interfaz bien definida y ocultar los detalles internos de la implementación, los TDAs pueden facilitar la escritura de código más claro y conciso, y reducir la posibilidad de errores y bugs.

XML:

XML (eXtensible Markup Language) es un lenguaje de marcado de texto que permite representar y almacenar información estructurada de manera legible para las personas y las máquinas. XML se utiliza para definir y describir los datos, y su estructura se basa en etiquetas que describen los diferentes elementos de la información.

XML es un estándar abierto que permite a los desarrolladores crear sus propios formatos de archivo personalizados y es ampliamente utilizado para el intercambio de datos entre sistemas. Por ejemplo, se utiliza a menudo para representar documentos, datos de configuración, mensajes en aplicaciones de mensajería, entre otros.

XML es una herramienta versátil y valiosa para la representación y el almacenamiento de información estructurada en la programación y la tecnología de la información.

Programación Orientada a Objetos:

La programación orientada a objetos (POO) es un paradigma de programación que se basa en el

concepto de "objetos". Un objeto es una entidad que combina datos y comportamientos, y se puede pensar en un objeto como una instancia de una clase.

En POO, se modelan los problemas y las soluciones en términos de objetos y clases. Una clase es una plantilla o definición que describe los datos y los comportamientos comunes de un conjunto de objetos. Cada objeto se crea a partir de una clase y hereda todos los datos y comportamientos definidos en la clase.

La POO tiene varias ventajas, incluyendo una mayor legibilidad y mantenibilidad del código, una mejor organización y reutilización de código, y una mayor capacidad para trabajar con soluciones complejas y modulares.

Algunos de los conceptos clave de la POO incluyen la encapsulación, la herencia y el polimorfismo. La encapsulación permite proteger los datos y los comportamientos de un objeto, mientras que la herencia permite reutilizar y ampliar código existente. El polimorfismo permite que los objetos de diferentes clases respondan de manera diferente al mismo mensaje o acción.

c. Construcción del Programa

El programa se construyó en lenguaje Python, dicho programa es capaz de transmitir las instrucciones óptimas para que las máquinas puedan elaborar nuevos compuestos.

El software cuenta con una interfaz de usuario, la cual cuenta con las siguientes opciones:

- Inicialización.
- Cargar un XML
- Generar un XML de salida.
- Gestión de los elementos químicos.

- Gestión de compuestos.
- Gestión de máquinas.
- Ayuda.

La aplicación permite elegir un archivo XML que contenga la lista de elementos, máquinas y compuestos.

Los datos iniciales son cargados en el programa, se identifican y almacenan los elementos, compuestos y máquinas para posteriormente ser utilizados.

El programa permite que el pin de cada máquina realice las siguientes funciones:

- Moverse hacia adelante.
- Moverse hacia atrás.
- Esperar.
- Fusionar elemento químico.

Cada máquina dentro del software tiene las siguientes restricciones:

- Solo se puede fusionar un elemento químico a la vez.
- Solo debe contener una vez cada elemento químico.

El software desarrollado le transmite las instrucciones óptimas para que la máquina pueda crear nuevos compuestos. Con dichas instrucciones se le transmite a la máquina una secuencia de elementos que debe fusionar en orden y se le asigna un nombre al compuesto.

La aplicación brinda una opción de inicialización, de esta manera se puede cargar un nuevo archivo desde cero, si se carga un archivo y existen organismos con muestras previamente existentes, éstas continúan en la aplicación, es decir; la carga de archivos incrementa la cantidad de información en el sistema.

d. Diagrama de Clase del Programa

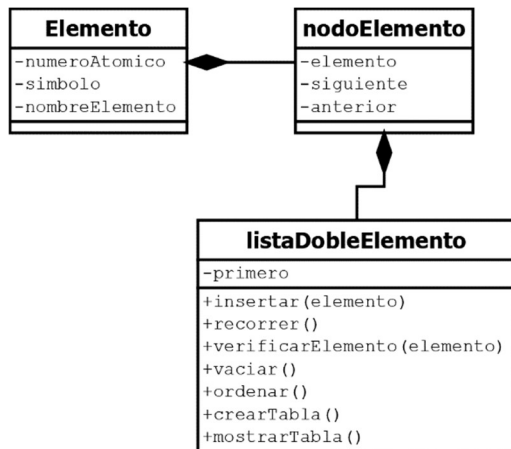


Figura 3. Diagrama de clase para Elemento.

Fuente: Elaboración propia, 2023.

La clase **Elemento** se encuentra como la clase “todo” para la clase **nodoElemento**, esto hace una relación de composición, ya que sin esta la clase **nodoElemento** no podría existir.

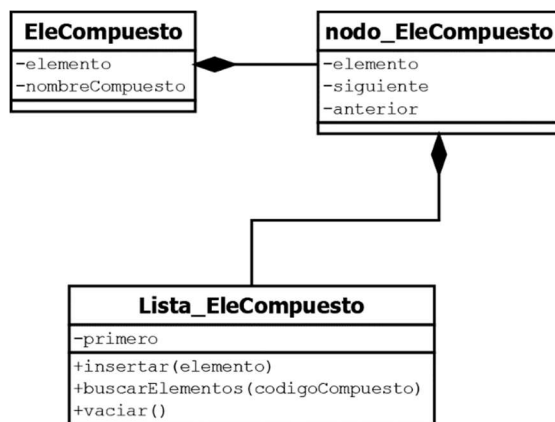


Figura 4. Diagrama de clase para Compuestos.

Fuente: Elaboración propia, 2023.

La clase **EleCompuesto** se encuentra como la clase “todo” para la clase **nodo_EleCompuesto**, esto hace una relación de composición, ya que sin esta la clase **nodo_EleCompuesto** no podría existir.

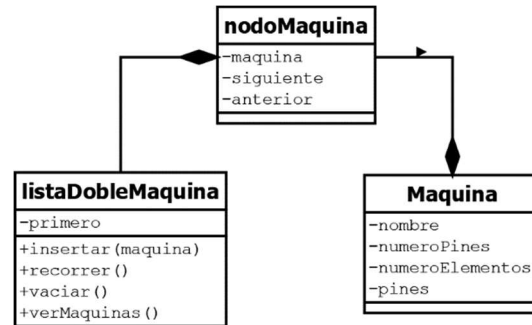


Figura 5. Diagrama de clase para Maquinas.

Fuente: Elaboración propia, 2023.

La clase **Maquina** se encuentra como la clase “todo” para la clase **nodoMaquina**, esto hace una relación de composición, ya que sin esta la clase **nodoMaquina** no podría existir.

Conclusiones

- Los TDAs son una herramienta importante para la programación, ya que permiten a los programadores trabajar con estructuras de datos complejas de manera fácil y segura, y también facilitan el mantenimiento del código a largo plazo.
- El uso de la Programación Orientada a Objetos facilita el almacenamiento de datos en memoria de las listas enlazadas al proporcionar una forma clara y organizada de representar y manipular los datos. Esto puede mejorar la eficiencia y la mantenibilidad del código, lo que a su vez puede aumentar la productividad y la calidad del software.
- El uso de XML es fundamental en muchos aspectos de la industria tecnológica, incluyendo la integración de sistemas, la transmisión de datos y la representación de información estructurada. Su facilidad de uso

y flexibilidad lo convierten en una herramienta valiosa para los desarrolladores y los profesionales de la tecnología.

Referencias bibliográficas

Amazon.(2018).*¿Qué es XML? - Explicación de XML - AWS*. Amazon Web Services, Inc.
<https://aws.amazon.com/es/what-is/xml/>

AppMaster. (2022). *¿Qué es un XML? AppMaster - ultimate all-in no-code platform*.
<https://appmaster.io/es/blog/que-es-xml>

Guerra, N. R. (2020, 5 marzo). Estructura de Datos: Lista Simplemente Enlazada. Technodyan.
<https://www.technodyan.com/lista-simplemente-enlazada/>

Programación, A. B. Y. (s. f.). TDA (Tipos de Datos Abstractos).
<http://blogalgoritmosyprogramacion.blogspot.com/2012/07/tda-tipos-de-datos-abstractos.html>

Anexos

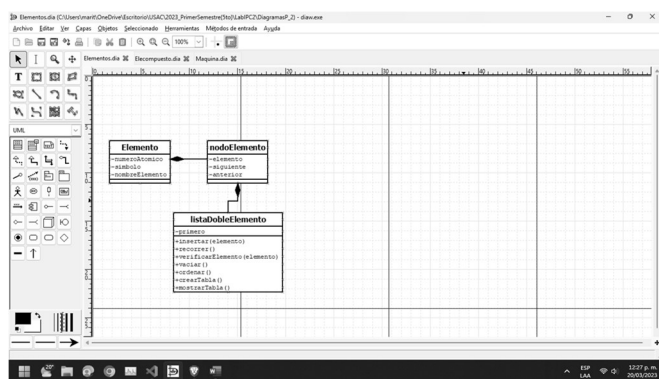


Figura 6. Creación de los diagramas de clases.

Fuente: Elaboración propia, 2023.

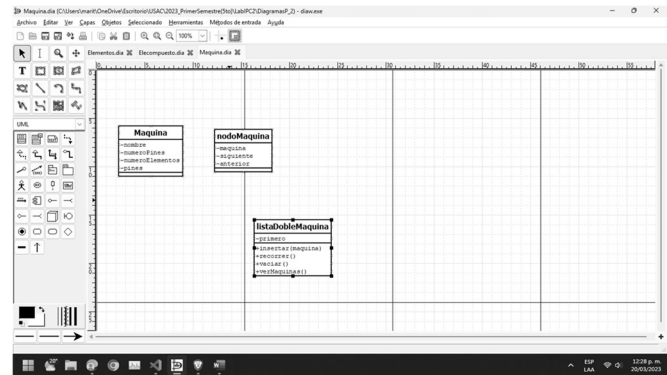


Figura 7. Creación de los diagramas de clases.

Fuente: Elaboración propia, 2023.

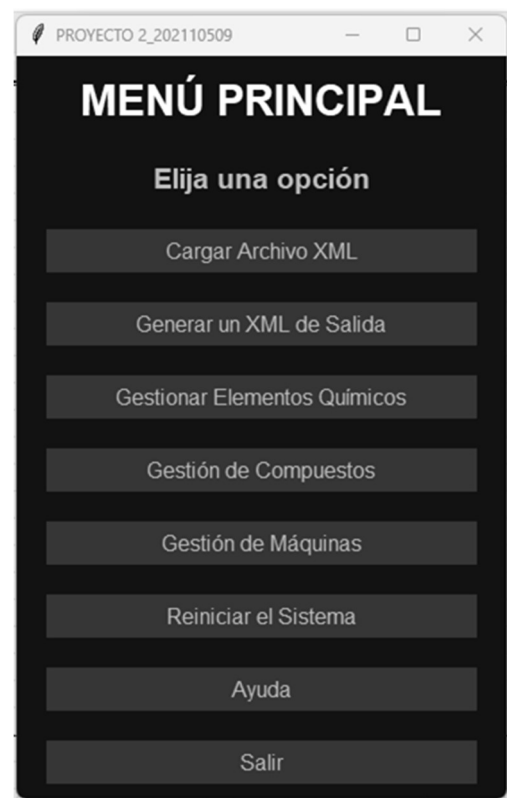
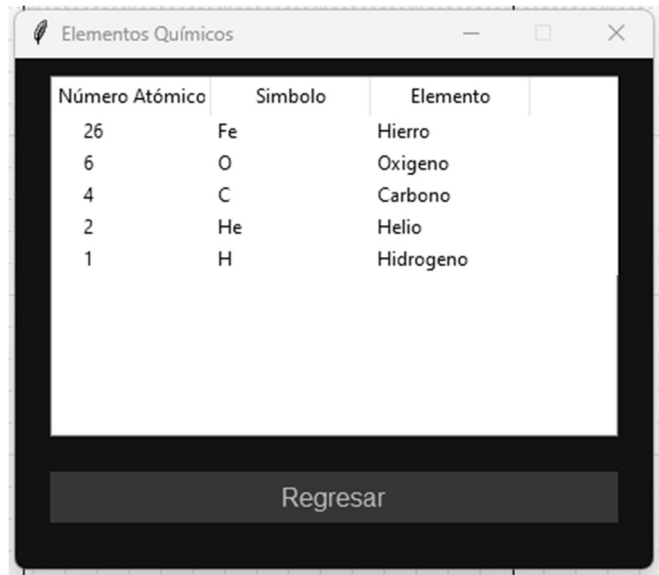


Figura 8. Interfaz gráfica.

Fuente: Elaboración propia, 2023.



Número Atómico	Símbolo	Elemento
26	Fe	Hierro
6	O	Oxígeno
4	C	Carbono
2	He	Helio
1	H	Hidrogeno

Regresar

Figura 9. Tabla de elementos.

Fuente: Elaboración propia, 2023.



Desarrollado por:

Mario Ernesto Marroquín Pérez

Carné:

202110509

Contacto:

2815806340401@ingenieria.usac.edu.gt

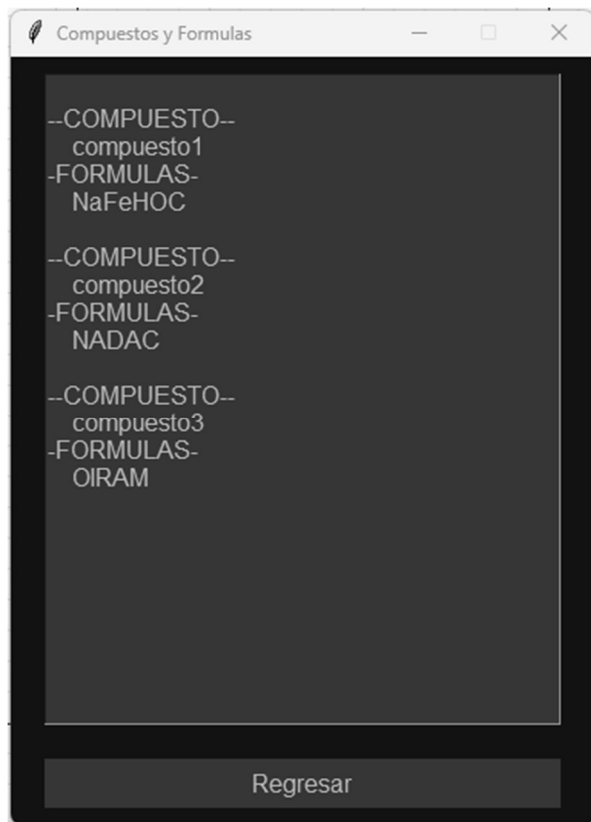
Manual de Usuario

Documentación Oficial

Regresar

Figura 11. Ventana de ayuda.

Fuente: Elaboración propia, 2023.



--COMPUESTO--
compuesto1
-FORMULAS-
NaFeHOC

--COMPUESTO--
compuesto2
-FORMULAS-
NADAC

--COMPUESTO--
compuesto3
-FORMULAS-
OIRAM

Regresar

Figura 10. Compuestos y formulas listados.

Fuente: Elaboración propia, 2023.

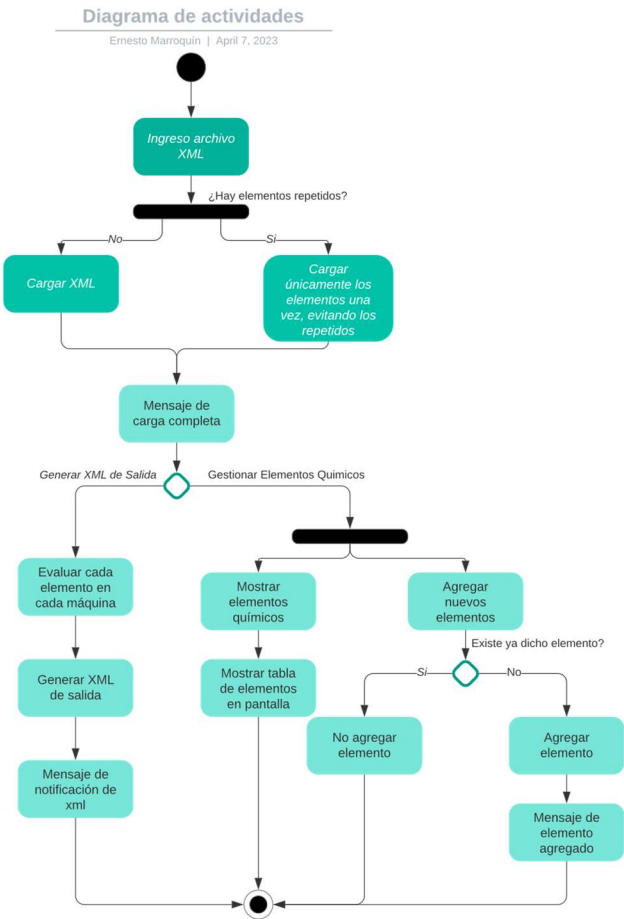


Figura 12. Diagrama de actividades.
Fuente: Elaboración propia, 2023.

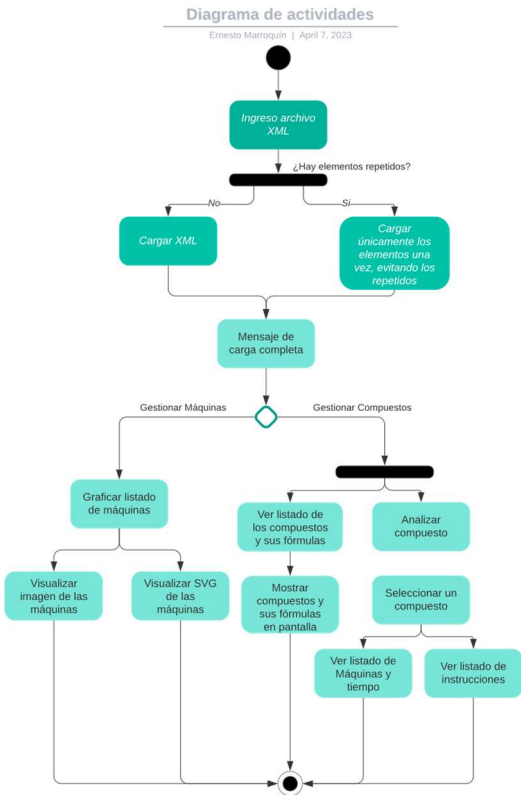


Figura 13. Diagrama de actividades.
Fuente: Elaboración propia, 2023.

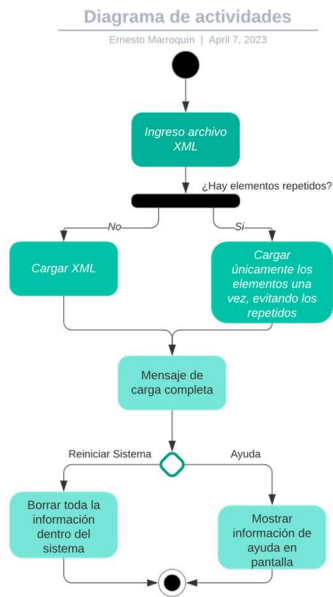


Figura 14. Diagrama de actividades.
Fuente: Elaboración propia, 2023.