

Universidad San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas
Introducción a la Programación y Computación 2

Ing. Claudia Liceth Rojas Morales
Ing. Marlon Antonio Pérez Türk
Ing. Byron Rodolfo Zepeta Arevalo
Ing. José Manuel Ruiz Juárez
Ing. Edwin Estuardo Zapeta Gómez

Tutores de curso:
Diego Pérez Alvarez
Marvin Daniel Rodríguez
Oscar Velásquez Leon
Daniel Arturo Alfaro
Sergio Felipe Zapeta



PROYECTO No. 2

OBJETIVO GENERAL

Modelar, documentar e implementar una solución al problema que se plantea utilizando las herramientas de desarrollo presentadas en clase y laboratorio.

OBJETIVOS ESPECÍFICOS

- Implementar una solución utilizando el lenguaje de programación Python.
- Utilizar estructuras de programación secuenciales, cíclicas y condicionales.
- Generar reportes con la herramienta Graphviz.
- Manipular archivos XML.
- Utilizar los conceptos de TDA y aplicarlos a memoria dinámica.
- Utilizar estructuras de programación propias.
- Utilizar el paradigma de programación orientada a objetos.

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

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.

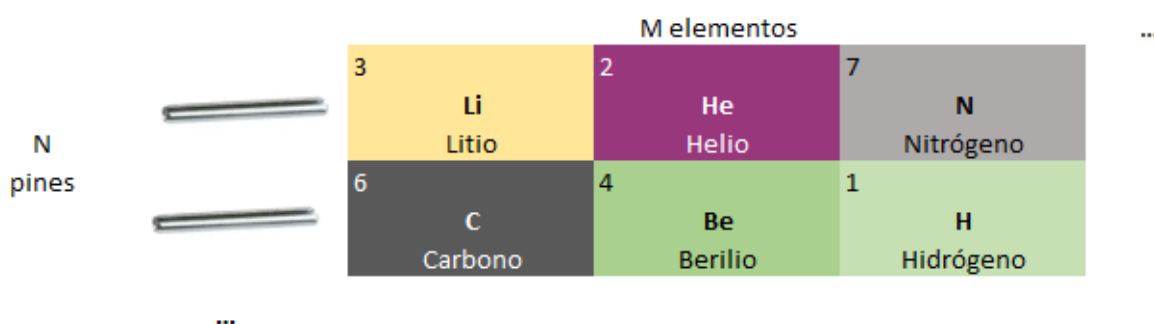


Figura No. 2 – Máquina de N pines x M elementos para crear nuevos compuestos

Cada pin de la máquina para crear compuestos puede realizar las siguientes funciones:

1. Moverse hacia adelante
2. Moverse hacia atrás
3. Esperar
4. Fusionar elemento químico

Las restricciones de cada máquina para crear compuestos son las siguientes:

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

Para crear un nuevo compuesto, simplemente se le transmite a la máquina una secuencia de elementos que debe fusionar en orden y se le da un nombre al compuesto, por ejemplo, el “Compuesto 1” podría estar constituido de la siguiente forma: Li, Li, Be, He, Be, Li.

La Facultad de Ingeniería de la USAC, desarrollará un software capaz de transmitir las instrucciones óptimas para que la máquina para crear nuevos compuestos pueda elaborarlos.

Por ejemplo, si tenemos la tabla de elementos químicos de la figura 1, y una máquina de 2 pines y 3 elementos como la que se muestra en la figura 2, al enviarse las instrucciones para crear el “Compuesto 1” (Li, Li, Be, He, Be, Li), el software desarrollado debe producir una salida como la que presenta la figura 3.

Tiempo óptimo para construir C1		7 segundos					
Instrucciones para construir el compuesto "C1"							
	Segundo 1	Segundo 2	Segundo 3	Segundo 4	Segundo 5	Segundo 6	Segundo 7
Pin 1	Mover adelante	Fusionar Li	Fusionar Li	Mover Adelante	Fusionar He	Mover Atrás	Fusionar Li
Pin 2	Mover adelante	Mover Adelante	Esperar	Fusionar Be	Esperar	Fusionar Be	Esperar

Figura 3 – Salida del software a construir para elaborar el “Compuesto 1” (Li, Li, Be, He, Be, Li)

En la figura 4 se puede observar gráficamente la forma en que la máquina trabaja para la creación del “Compuesto 1”.

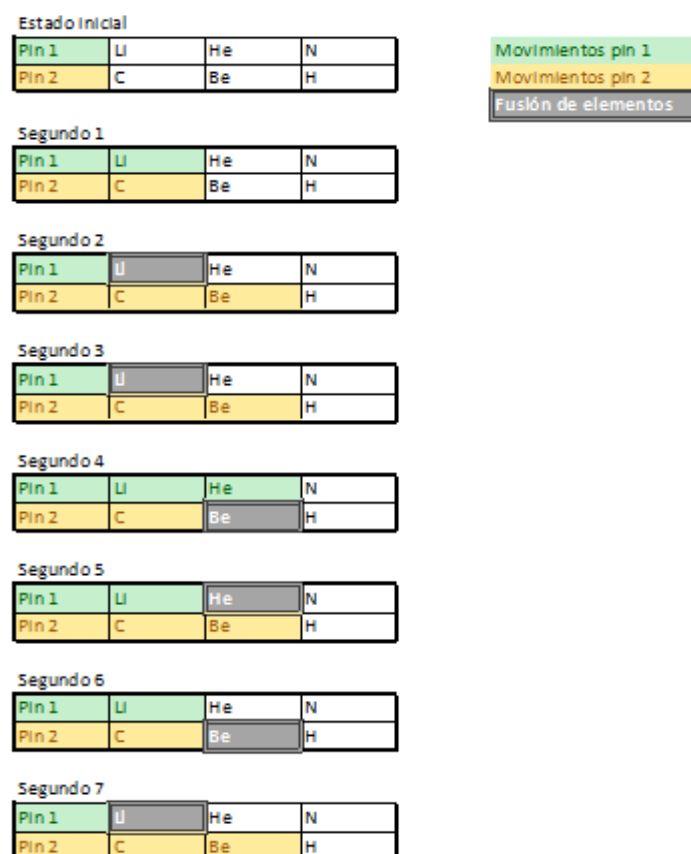


Figura 4 – Imagen de funcionamiento de la máquina para crear el “Compuesto 1”

ENTRADA

El archivo de entrada tendrá el siguiente formato XML

entrada.xml

```
<?xml version="1.0"?>
<CONFIG>
  <listaElementos>
    <elemento>
      <numeroAtómico>[valorNumerico]</numeroAtómico>
      <símbolo>[valorAlfanumerico]</símbolo>
      <nombreElemento>[valorAlfanumerico]</nombreElemento>
    </elemento>
    ...
  </listaElementos>
  <listaMaquinas>
    <Maquina>
      <nombre>[valorAlfanumerico]</nombre>
      <numeroPines>[valorNumerico]</numeroPines>
      <numeroElementos>[valorNumerico]</numeroElementos>
      <pin>
        <elementos>
          <elemento>[símboloDelElemento]</elemento>
          ...
        </elementos>
      </pin>
      ...
    </Maquina>
    ...
  </listaMaquinas>
  <listaCompuestos>
    <compuesto>
      <nombre>[valorAlfanumerico]</nombre>
      <elementos>
        <elemento>[símboloDelElemento]</elemento>
        ...
      </elementos>
    </compuesto>
    ...
  </listaCompuestos>
</CONFIG>
```

Tomar en cuenta este archivo es incremental, es decir, se pueden ingresar varios archivos de entrada, de tal manera que se pueden crear nuevos elementos químicos, nuevas máquinas y nuevos compuestos.

SALIDA

Se debe generar un archivo de salida que muestre los compuestos que posee el sistema, la máquina que podría elaborarlo de forma óptima, el tiempo óptimo y la secuencia de pasos para elaborar el compuesto.

Salida.xml

```
<?xml version="1.0"?>
<RESPUESTA>
  <listaCompuestos>
    <compuesto>
      <nombre>[valorAlfanumerico]</nombre>
      <maquina>[valorAlfanumerico]</maquina>
      <tiempoOptimo>[valorNumerico]</tiempoOptimo>
      <instrucciones>
        <tiempo>
          <numeroSegundo>[valorNumerico]</numeroSegundo>
          <acciones>
            <accionPin>
              <numeroPin>[valorNumerico]</numeroPin>
              <accion>[valorAlfanumerico]</accion>
            </accionPin>
            ...
          </acciones>
        </tiempo>
        ...
      </instrucciones>
    </compuesto>
    ...
  </listaCompuestos>
</RESPUESTA>
```

INTERFAZ DE USUARIO

Se debe crear una interfaz de usuario fácil de utilizar e intuitiva que permita al usuario realizar las siguientes funciones:

- a. Inicialización – Para que el sistema pueda inicializarse sin ninguna información previa.
- b. Cargar un archivo XML de entrada
- c. Generar un archivo XML de salida
- d. Gestión de elementos químicos
 - a. Ver listado de elementos químicos ordenado por número atómico
 - b. Agregar un nuevo elemento químico (no deben haber elementos químicos con el mismo número atómico, el mismo símbolo o el mismo nombre)
- e. Gestión de Compuestos
 - a. Ver listado de compuestos y sus fórmulas
 - b. Analizar compuesto
 - i. Seleccionar un compuesto
 - ii. Ver listado de máquinas y tiempos necesarios para producir el compuesto
 - iii. Ver gráficamente (utilizando Graphiz) el listado de instrucciones con que una máquina puede producir el compuesto
- f. Gestión de máquinas
 - a. Listado de máquinas
- g. Ayuda – Mostrar la información del estudiante y un link hacia la documentación del proyecto.

CONSIDERACIONES

Se deberá realizar la implementación utilizando programación orientada a objetos, algoritmos desarrollados por el estudiante e implementación de estructuras a través de Tipos de Dato Abstracto (TDAs) propios del estudiante, es decir, no deben utilizar estructuras de datos propias de Python (list, dict, tuple, set). El estudiante deberá abstraer la información y definir qué estructuras implementar para construir la solución.

Debe utilizarse versionamiento para el desarrollo del proyecto. Se utilizará la plataforma **Github** en la cual se debe crear un repositorio en el que se gestionará el proyecto. Se deben realizar **4 releases** o versiones del proyecto (Se sugiere desarrollar un Release por semana). Se deberá agregar a sus respectivos auxiliares como colaboradores del repositorio. El último release será el release final y se deberá de realizar antes de entregar el proyecto en la fecha estipulada.

DOCUMENTACIÓN

Para que el proyecto sea calificado, el estudiante deberá entregar la documentación utilizando el formato de ensayo definido para el curso. En el caso del proyecto, el ensayo puede tener un mínimo de 4 y un máximo de 7 páginas de contenido, este máximo no incluye los apéndices o anexos donde se pueden mostrar modelos y diseños utilizados para construir la solución. Es obligatorio incluir el diagrama de clases del diseño del software desarrollado y se recomienda incluir el modelo conceptual y los diagramas de actividades que modelan la solución de software presentada por el estudiante.

RESTRICCIONES

- Solo se permitirá la utilización de los IDEs discutidos en el laboratorio.
- Uso de TDA implementados por el estudiante obligatorio; el uso de estructuras de Python (list, dict, tuple, set) resultará en penalización del 100% de la nota.
- Uso obligatorio de programación orientada a objetos (POO) desarrollada por completo por el estudiante. De no cumplir con la restricción, no se tendrá derecho a calificación.
- El nombre del repositorio debe de ser **IPC2_Proyecto2_#Carnet**.
- El estudiante debe entregar la documentación solicitada para poder optar a la calificación.
- Los archivos de entrada no podrán modificarse.
- Los archivos de salida deben llevar la estructura mostrada en el enunciado obligatoriamente.
- Deben existir 4 releases mínimo, podría ser uno por cada semana, de esta manera se corrobora el avance continuo del proyecto.
- Se calificará de los cambios realizados en el último release hasta la fecha de entrega. Los cambios realizados después de ese release no se tomarán en cuenta.
- Cualquier caso de copia parcial o total tendrá una nota de 0 y será reportada a escuela.
- Para dudas concernientes al proyecto se utilizarán los foros en UEDI de manera que todos los estudiantes puedan ver las preguntas y las posteriores respuestas.
- **NO HABRÁ PRÓRROGA.**

ENTREGA

- La entrega será el martes **04 de abril**, a más tardar a las 11:59 pm.
- La entrega será por medio de la UEDI.
- La documentación debe estar subida en el repositorio en una carpeta separada.
- Para entregar el proyecto en UEDI se deberá subir un archivo de texto con el link del repositorio.