

U. SERGIO ARBOLEDA

PARCIAL N°1

Pensamiento algorítmico

Andres Felipe Velasquez Salinas

¡RADIACION AL LIMITE!

En el mundo de Dr. Stone, el conocimiento sobre la radiactividad es crucial para la supervivencia y el avance de la ciencia. Senku y su equipo han encontrado varios isótopos radiactivos y necesitan clasificarlos según su estabilidad para usarlos de manera segura en sus experimentos.

Para ello, deben determinar la categoría de un isótopo en función de su vida media T (el tiempo que tarda en desintegrarse la mitad de su masa). Un isótopo con una vida media corta es altamente radiactivo, mientras que uno con vida media larga es más estable y seguro.

ANALISIS DEL PROBLEMA.

Descripción del problema: Se necesitan clasificar los isotopos de forma que podamos identificar que isotopos son mas radioactivos que otros, Para esto nos basamos en la vida media del isotopo, que representa el tiempo en el cual la mitad de la masa se desintegra.

Identificación de los requisitos funcionales y no funcionales:

Funcionales:

- Categoría del isótopo (Alta radioactividad, Radioactividad moderada, Baja radioactividad)
- Vida media del isótopo
- Tiempo en el cual la mitad de la masa del isótopo se desintegra

No funcionales:

-

Análisis de casos de uso:

- Caso 1:** Un isótopo es muy radioactivo (La mitad de su masa se desintegra en 0.5 segundos (5×10^8 ns))
- Caso 2:** Un isótopo es de radioactividad moderada (La mitad de su masa se desintegrará en 5000 segundos (5×10^{12} ns))
- Caso 3:** Un isótopo es de radioactividad baja (La mitad de su masa se desintegra en 100.000 segundos (1×10^{14} ns))

Identificación de las entradas, proceso y salidas

Entrada	Proceso	Salida
5e8	Se identifica el numero en nanosegundos para saber cuanto es su valor en segundos, si es menor a 1 tendrá la siguiente salida:	Alta radioactividad
5e12	Se identifica el numero en nanosegundos para saber cuánto es su valor en segundos, si es entre 1 y 86400 tendrá la siguiente salida:	Radioactividad moderada
1e14	Se identifica el numero en nanosegundos para saber cuánto es su valor en segundos, si supera los 86400 segundos tendrá esta salida:	Baja radioactividad

JUSTIFICACION DE LA SOLUCION.

Explicación detallada de la estrategia elegida para resolver el problema:

Para esto lo dividiremos en tres partes, primero tendremos en cuenta los datos que nos dará el usuario que será la vida media del isotopo, como segunda parte este dato se divide en $1e+9$ (1.000.000.000)

El dato que resulte de esta división se compara con las condiciones establecidas, y por último se definirá cual de las tres salidas tendrá que tomar este dato

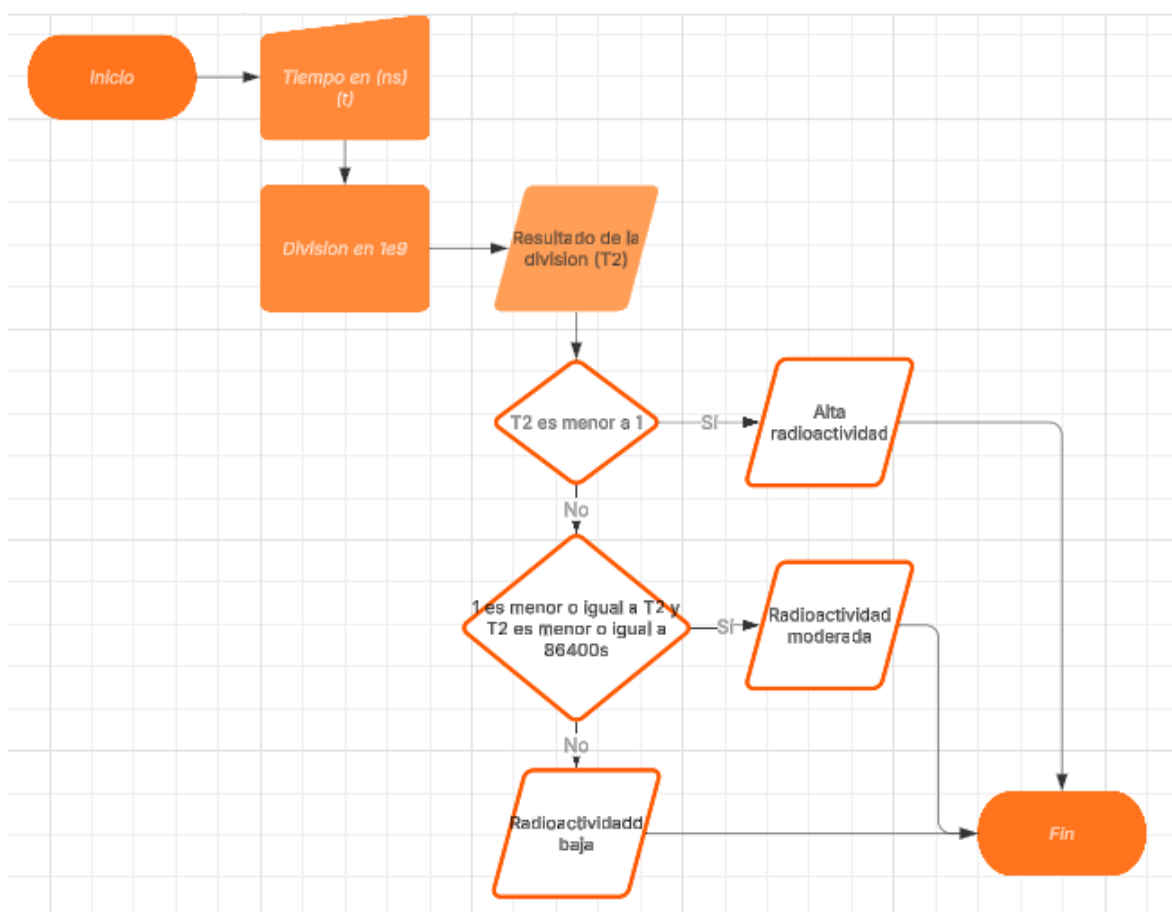
Justificación de las estructuras de datos y algoritmos seleccionados:

Uso de condicionales: Son la mejor opción ya que al utilizarles podemos delimitar hasta que momento los isotopos tienen su nivel de radioactividad

Conversion a nanosegundos: Se desarrolla una división para tener este dato en (ns) y de forma en el que el mundo del Dr.Stone podrá saber con mas claridad el nivel de radioactividad

Comparación con posibles soluciones alternativas y razones de la elección final: Una opción podría ser poder todas las variables junto con sus llaves que serian los tiempos en lo que se identifica que tan radioactivo son los isotopos.

DIAGRAMA UML



IMPLEMENTACION

Código en Python:

```
1
2 #Se pide el tiempo al usuario
3 T = float(input("Digite los nanosegundos: "))
4
5 #Division en 1e9 (1.000.000.000)
6 T2 = T / 1e9
7
8 #T2= Es la variable del tiempo ingresado representada en segundos
9 print( T2 )
10
11 #Incluimos las condiciones
12 if (T2 <= 1):
13     print ("Alta radioactividad")
14 elif 1 < T2 <= 86400:
15     print ("Radioactividad moderada")
16 else:
17     print ("Radioactividad baja")
18
19
```

Código en C++

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      //Definimos las variables que utilizaremos
8      double T,T2;
9
10     //Se le pide al usuario el tiempo medio de la vida del isotopo
11     cout << ("Ingrese los nanosegundos: ");
12     cin >> T;
13
14     //El valor T se dividira en 1e9 (1.000.000.000)
15     T2 = (T) / 1e9;
16
17     //Concluimos las condiciones para imprimir el nivel de radiacion
18     if(T2<=1){
19         cout<<("Radiacion alta");
20     }
21     else{
22         if(1<T2 && T2 < 84600){
23             cout<<("Radiacion moderada");}
24         else{
25             if(T2>84600){
26                 cout<<("Radiacion baja");
27             }
28         }
29     }
30 }
31
32 }
```

DOCUMENTACION DEL CODIGO

Python

Instrucciones de ejecución:

Al ejecutar el código se pedirá que se inserte un valor en nanosegundos (ns) el cual será la vida media del isotopo (T)

Un ejemplo del valor en nanosegundos podría ser (5e8) o (1e14)

Ejemplos de uso:

Windows: En este caso se podría utilizar Visual Studio Code o instalar la misma aplicación de Python para ejecutarlo

Linux y Mac: Para estos sistemas operativos usualmente Python viene ya incluido para verificarlo debes abrir una terminal y ejecutar:

```
python3 --version
```

Entrada: 5e12

Salida: Radioactividad Moderada

C++

Instrucciones de ejecución:

El programa debe de tener un compilador que sea compatible con C++

Windows: Para este sistema operativo podría utilizarse Visual Studio code para la ejecución del programa

Para Linux o Mac: En este caso sistemas como g++ servirán para ejecutar este

Ejemplos de uso:

Al ejecutar el código se pedirá que se inserte un valor en nanosegundos (ns) el cual será la vida media del isotopo (T)

Entrada: 5e8

Salida: Alta radioactividad