

Universidad Sergio Arboleda  
Facultad: Ingeniería y ciencias exactas

Pensamiento Algorítmico



Proyectiles en Aetheria

David Santiago Ramos Duque  
Samuel David Rivera Gonzalez

11/03/2025

# Informe

## Introducción:

El siguiente proyecto busca como objetivo principal desarrollar un código funcional para calcular el tiempo de impacto de proyectiles mágicos en el mundo de Aetheria. Un juego MMORPG (juego de rol multijugador masivo en línea). Los principales proyectiles existentes son de fuego, hielo y veneno, cada uno con una velocidad única. Dado el tipo de proyectil y la distancia al objetivo en metros, la solución al problema es resolver cuanto es el tiempo que tarda en impactar al objetivo en segundos.

## 1.1 Análisis del problema

1. El problema consiste en calcular el tiempo que tarda un proyectil en impactar un objetivo, dada una distancia en metros y un tipo de proyectil (fuego, hielo y veneno) elegido por el jugador. Para ello, se deben considerar las diferentes velocidades de los proyectiles disponibles.

2. Entradas: Los proyectiles que son:

**Fuego:** 120 m/s

**Hielo:** 100 m/s

**Veneno:** 80 m/s

3. Proceso: Es el cálculo del tiempo de impacto  
Se calcula con la fórmula:  $\text{tiempo} = \text{distancia} / \text{velocidad} = \text{Segundos}$   
Se redondea a 2 decimales.
4. Salida:
  - Si el proyectil es válido:  
Se muestra su velocidad en m/s.  
Se muestra su tiempo de impacto en s.
  - Si el proyectil no es válido:  
Se indica que no es un proyectil válido.  
No se puede calcular el tiempo de impacto.
5. Algunos requisitos funcional y no funcionales son:  
Funcionales: es un proyectil válido.  
No funcionales: es un proyectil no válido.

6. Análisis de casos de uso

Casos de uso de los 3 proyectiles que tenemos:

### Caso de Uso 1:

Entrada: Información del tipo de proyectil (Hielo) y su distancia (150 metros)

Proceso: Calcular el tiempo usando la velocidad del proyectil de Fuego (100 m/s).

Salida: "Tiempo de impacto: 1.50 segundos".

### **Caso de Uso 2:**

Entrada: Información del tipo de proyectil (fuego) y su distancia (240 metros.)

Proceso: Calcular el tiempo usando la velocidad del proyectil de Fuego (120 m/s).

Salida: "Tiempo de impacto: 2.00 segundos".

### **Caso de Uso 3:**

Entrada: Información del tipo de proyectil (veneno) y su distancia (80 metros.)

Proceso: Calcular el tiempo usando la velocidad del proyectil de Fuego (80 m/s).

Salida: "Tiempo de impacto: 1.00 segundos".

## **1.2. Justificación de la solución**

1. La estrategia utilizada fue seguir algunos pasos para la interpretación del problema al programa, el primer paso sería definir las variables de los proyectiles con su respectivas velocidades, el segundo paso sería definir la distancia al objetivo en metros, el tercer paso es colocar si los proyectiles son válidos o inválidos, el cuarto paso sería calcular la velocidad y el tiempo de impacto del proyectil y el último paso será ejecutar los proyectiles.

### **2. Estructuras de datos:**

Una estructura de datos es una forma de organizar, almacenar y gestionar datos para que se pueda definir una variable.

```
proyectiles = { "Fuego":120, "Hielo": 100, "Veneno": 80}
```

Se usa esta estructura de dato para definir cada variable con su respectivo dato, por ejemplo Fuego = 120.

```
d = int(input("Ingrese la distancia al objetivo en metros: "))
```

Se usa para definir en este caso la distancia que es d con un dato que es int, son números enteros para saber la distancia que se encuentra el proyectil.

### 3. Algoritmos:

Es un conjunto de pasos organizados para resolver un problema.

```
def obtener_velocidad(proyectil):  
    return proyectiles.get(proyectil, None)
```

Este algoritmo sirve para resolver la velocidad de cada proyectil, también puede decir si es un proyectil o no es un proyectil

```
def calcular_tiempo(proyectil, distancia):  
    Este algoritmo sirve para resolver el tiempo en que tarda en impactar cada proyectil
```

```
proyectiles_prueba = ["Fuego", "Hielo", "Veneno", "Rayo"]:  
    Este algoritmo sirve para resolver cada proyectil con su respectiva velocidad en m/s y tiempo en segundos.
```

```
if velocidad is not None:  
    Este algoritmo sirve para resolver si es o no es un proyectil.
```

En conclusión este código final se diseñó para ser eficiente y fácil de entender y simplificar el código para no hacerlo muy largo.

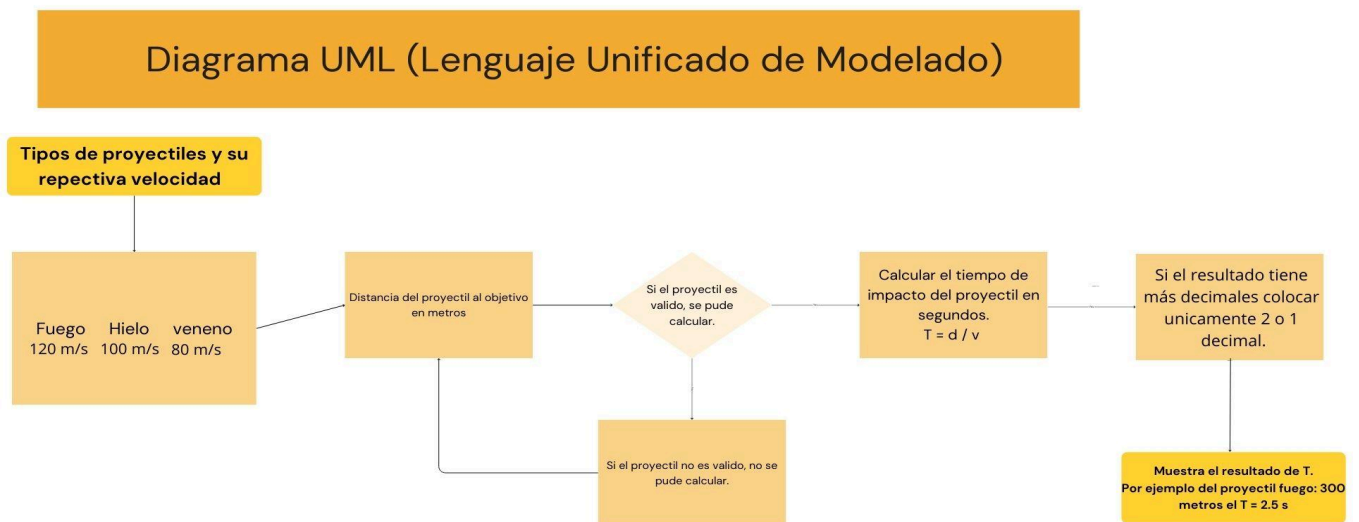
**1.5** Posibles soluciones alternativas para el programa utilizando if-elif y else será más largo, por ejemplo:

```
def obtener_velocidad(proyectil):  
    if proyectil == "Fuego":  
        return 120  
    elif proyectil == "Hielo":  
        return 100  
    elif proyectil == "Veneno":  
        return 80  
    else:  
        return None
```

```
def calcular_tiempo(proyectil, distancia):  
  
    velocidad = obtener_velocidad(proyectil)  
    return round(distancia / velocidad, 2)  
except TypeError:  
    return "No se puede calcular"
```

Este sería otra alternativa para el código mostrando un código más largo y más o menos fácil de entender.

## 1.3 Diagrama UML



## 2. Implementación

### 2.1 Código en Python

```
1 # Definir las velocidades de los proyectiles
2 proyectiles = { "Fuego":120, "Hielo": 100, "Veneno": 80}
3 # Definir la distancia al objetivo
4 d = int(input("Ingrese la distancia al objetivo en metros: "))
5 # Si los proyectiles son validos o no son validos y obtener la
   velocidad en m/s
6 def obtener_velocidad(proyectil):
7     return proyectiles.get(proyectil, None)
8 # calcular la velocidad y el tiempo de impacto del proyectil
9 def calcular_tiempo(proyectil, distancia):
10    velocidad = obtener_velocidad(proyectil)
11    if velocidad is not None: # Verifica si la velocidad es válida
12        return round(distancia / velocidad,2)
13    return "No se puede calcular"
14 # Resultado final
15 proyectiles_prueba = ["Fuego", "Hielo", "Veneno", "Rayo"]
16 for p in proyectiles_prueba:
17     tiempo = calcular_tiempo(p, d)
18     velocidad = obtener_velocidad(p)
19     print(f"{p}: {velocidad if velocidad is not None else 'Proyectil
       no válido'} m/s, Tiempo de impacto: {tiempo} s")
```

Resultado del programa:

```
Ingrese la distancia al objetivo en metros: 300
Fuego: 120 m/s, Tiempo de impacto: 2.5 s
Hielo: 100 m/s, Tiempo de impacto: 3.0 s
Veneno: 80 m/s, Tiempo de impacto: 3.75 s
Rayo: Proyectoil no válido m/s, Tiempo de impacto: No se puede calcular s
```

### 3. Documentación del código

#### 3.1. Documentación Python

#### 3.2 Los comentarios explicativos son los siguientes:

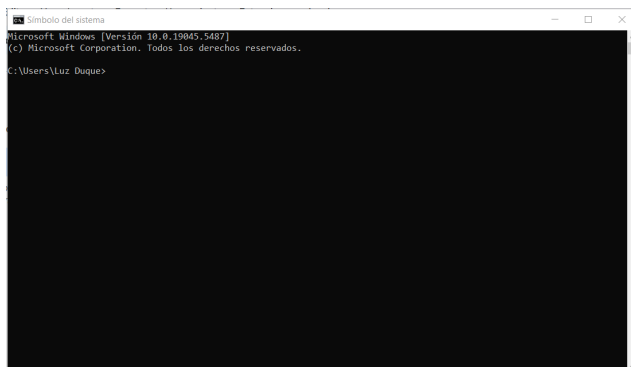
- # Definir las velocidades de los proyectiles
- # Definir la distancia al objetivo
- # Si los proyectiles son válidos o no son válidos y obtener la velocidad en m/s
- # calcular la velocidad y el tiempo de impacto del proyectil
- # Resultado final

#### 3.3 Las instrucciones para ejecutar el programa en diferentes sistemas operativos por ejemplo en Windows son:



Windows:

1. Abrir el símbolo del sistema (Cmd) o PowerShell, en el computador abrir en la parte inferior Windows y en el buscador escribir “símbolo de sistema”



2. Navegar hasta la carpeta donde se encuentra el archivo .py usando el comando 'cd'.

```
Administrador: Símbolo del sistema
Microsoft Windows [Versión 10.0.19045.5487]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Windows\system32>CD..
C:\Windows>CD..
C:\>DIR
El volumen de la unidad C no tiene etiqueta.
El número de serie del volumen es: F22E-1447

Directorio de C:\

10/02/2025  18:01 p. m.  <DIR>          .Py
09/03/2025  18:17 a. m.  <DIR>          Intel
07/12/2019  04:14 a. m.  <DIR>          Perflogs
10/03/2025  08:58 p. m.  <DIR>          Program Files
12/02/2025  09:09 p. m.  <DIR>          Program Files (x86)
14/01/2025  10:15 p. m.  <DIR>          Users
09/03/2025  07:05 p. m.  <DIR>          Windows
                0 archivos            0 bytes
                7 dirs   60.895.754.240 bytes libres

C:\>CD .Py
C:\.Py>
```

### 3. Ejecutar el programa con: python nombre\_del\_archivo.py

```
Administrador: Símbolo del sistema
Microsoft Windows [Versión 10.0.19045.5487]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Windows\system32>cd..
C:\Windows>cd..
C:\Windows>cd..
C:\>cd .Py
```

**3.4** Los ejemplos de uso, es como se vería el programa y los ejemplos de uso es la entrada y salida del programa:

La entrada:

Ingrese la distancia al objetivo en metros: 200

La salida:

Fuego: 120 m/s, Tiempo de impacto: 1.67 s

Hielo: 100 m/s, Tiempo de impacto: 2.0 s

Veneno: 80 m/s, Tiempo de impacto: 2.5 s

Rayo: Proyectil no válido

