

Universidad Sergio Arboleda
Facultad: Ingeniería y ciencias exactas
Pensamiento Algorítmico

Solución Computacional para el Cálculo del Tiempo de Impacto de proyectiles en Aetheria

David Santiago Ramos Duque
Rivera Gonzalez Samuel David

Universidad Sergio Arboleda



Fecha de presentación: 11/03/2025

1.1 Análisis del problema

Descripción del problema

En el mundo de Aetheria, un MMORPG (juego de rol multijugador masivo en línea). Los jugadores cuentan con la posibilidad de lanzar tres tipos de proyectiles mágicos: fuego, hielo y veneno. Cada proyectil cuenta con una velocidad distinta.

Fuego: 120 metros por segundo.

Hielo: 100 metros por segundo.

Veneno: 80 metros por segundo.

El objetivo principal del problema es crear una herramienta que le permita a los jugadores calcular el tiempo que tardará un proyectil en impactar a un objetivo, dada la distancia al mismo. El programa tiene que manejar los tres tipos de proyectiles y mostrar el tiempo de impacto con decimales (2 decimales). también se debe mostrar error si el tipo de proyectil no es válido.

Identificación de los requisitos

Funcionales

- El programa debe solicitar al usuario:
La distancia del objetivo en metros (decimal positivo)
- El programa debe calcular el tiempo de impacto utilizando la fórmula:

$$\text{tiempo} \longrightarrow t = \frac{d}{v}$$

t: tiempo d: distancia v: velocidad

- El programa tiene que mostrar el tiempo del impacto usando dos decimales
- Si el tipo de proyectil colocado no es válido, el programa tiene que mostrar un texto de error

No funcionales

- El programa debe funcionar en diferentes sistemas operativos (Windows, Linux, macOS).
- El programa debe ser fácil de usar, con mensajes claros y una interfaz sencilla.

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".

Caso de Uso 4(no funcional):

Entrada: Información del tipo de proyectil Rayo) y su distancia (200 metros.)

Proceso: Validar que el tipo de proyectil no es válido.

Salida: "Tipo de proyectil no valido"

Identificación de Entradas, Procesos y Salidas Esperadas**Entradas:**

Un número decimal positivo que representa la distancia al objetivo en metros.

Procesos:

- Verificar que el tipo de proyectil sea uno de los permitidos.
- Utilizar la fórmula ($t = d/v$), donde la velocidad depende del tipo de proyectil.

Mostrar el tiempo de impacto con dos decimales de precisión.

Salidas Esperadas:**Si el tipo de proyectil es válido:**

Un mensaje que indique el tiempo de impacto en segundos, con dos decimales de precisión.

Si el tipo de proyectil no es válido:

Un mensaje de error: "Tipo de proyectil no válido".

1.2 Justificación de la solución**Estrategia seleccionada:**

Al saber cosas como la velocidad del proyectil ya podíamos reconocer algo bastante evidente. La operación matemática necesaria para poder hallar el tiempo. Conocemos la velocidad. El usuario coloca la distancia. Lo que nos da como resultado el tiempo.

Justificación de las Estructuras de Datos y Algoritmos Seleccionados

Estructuras de Datos:

Variables Simples:

Se utilizan variables simples para almacenar el tipo de proyectil (string), la distancia (double) y la velocidad (double) (distinto cuando se usa el C++ ttype guide).

Algoritmos:

Validación con Condicionales:

Se utiliza una estructura if-else (distinto cuando se usa el C++ ttype guide) para validar el tipo de proyectil y asignar la velocidad correspondiente.

Cálculo Matemático:

El cálculo del tiempo de impacto se realiza mediante una simple división, lo que es eficiente y rápido

1.3 Diagrama de UML:

(Está adjunto en el repositorio)

2.2 Código en C++

```
main.cpp
4
5 int main() {
6     // Variable para almacenar la distancia al objetivo
7     double distancia;
8
9     // Solicitar al usuario la distancia al objetivo
10    cout << "Introduce la distancia al objetivo (metros): ";
11    cin >> distancia;
12
13    // Validar que la distancia sea un número positivo
14    if (distancia <= 0) {
15        cout << "La distancia debe ser un número positivo." << endl;
16        return 0; // Terminar el programa si la distancia no es válida
17    }
18
19    // Velocidades de los proyectiles
20    const double velocidad_fuego = 120; // Velocidad del proyectil de Fuego (m/s)
21    const double velocidad_hielo = 100; // Velocidad del proyectil de Hielo (m/s)
22    const double velocidad_veneno = 80; // Velocidad del proyectil de Veneno (m/s)
23
24    // Calcular el tiempo de impacto para cada proyectil
25    double tiempo_fuego = distancia / velocidad_fuego;
26    double tiempo_hielo = distancia / velocidad_hielo;
27    double tiempo_veneno = distancia / velocidad_veneno;
28
29    // Mostrar los resultados con dos decimales de precisión
30    cout << fixed << setprecision(2); // Formatear la salida a dos decimales
31
32    // Mostrar el tiempo de impacto para el proyectil de Fuego
33    cout << "Tiempo de impacto (Fuego): " << tiempo_fuego << " segundos" << endl;
34
35    // Mostrar el tiempo de impacto para el proyectil de Hielo
36    cout << "Tiempo de impacto (Hielo): " << tiempo_hielo << " segundos" << endl;
37
38    // Mostrar el tiempo de impacto para el proyectil de Veneno
39    cout << "Tiempo de impacto (Veneno): " << tiempo_veneno << " segundos" << endl;
40
41    return 0; // Indicar que el programa terminó correctamente
42 }
```

input

```
Introduce la distancia al objetivo (metros): 7
Tiempo de impacto (Fuego): 0.06 segundos
Tiempo de impacto (Hielo): 0.07 segundos
Tiempo de impacto (Veneno): 0.09 segundos

...Program finished with exit code 0
Press ENTER to exit console.
```

Documentación (C++)

Instalación

Compilador de C++: Asegúrate de tener instalado un compilador de C++ como ****g++*** (GNU Compiler Collection)

Editor de Texto o IDE: Puedes usar cualquier editor de texto o IDE que admita C++, como Visual Studio Code, Code::Blocks o Dev-C++.

Caso de MacOS o Linux:

Descarga o copia el archivo `proyectiles.cpp` en tu computadora.

- Abre una terminal y navega hasta la carpeta donde está el archivo.
- Compila el programa con el siguiente comando:

```
g++ proyectiles.cpp -o proyectiles
```

3. Ejecutar el Programa:

- Ejecuta el programa con el siguiente comando:

```
./proyectiles
```

4. Guía de uso

Un número decimal positivo que representa la distancia en metros.

Una vez que el usuario ingresa este dato, el programa calcula el tiempo de impacto y muestra el resultado en segundos, con dos decimales de precisión. Si el tipo de proyectil no es válido, el programa muestra un mensaje de error.

Ejemplos de Uso

Ejemplo 1: Proyectoil de Fuego

- Entrada:

Introduce el tipo de proyectil (Fuego, Hielo, Veneno): Fuego

Introduce la distancia al objetivo (metros): 240

- Salida:

Tiempo de impacto: 2.00 segundos

*Ejemplo 2: Proyectoil de Hielo

- Entrada:

Introduce el tipo de proyectil (Fuego, Hielo, Veneno): Hielo

Introduce la distancia al objetivo (metros): 150

- Salida:

Tiempo de impacto: 1.50 segundos

Ejemplo 3: Tipo de Proyectoil No Válido*

- Entrada:

Introduce el tipo de proyectil (Fuego, Hielo, Veneno): Rayo

Introduce la distancia al objetivo (metros): 200

- Salida:

Tipo de proyectil no válido.

5. Documentación Técnica

El programa está compuesto por un único archivo (proyectiles.cpp) que contiene el código fuente en C++. A continuación, se describe la estructura del código:

. Inclusión de Bibliotecas

- include <iostream>: Para manejar la entrada y salida estándar.
- include <iomanip>: Para formatear la salida con dos decimales.

El código es un programa estructurado que se encuentra en la función main().

Constantes:

kVelocidadFuego: Velocidad del proyectil de fuego (120 m/s).

kVelocidadHielo: Velocidad del proyectil de hielo (100 m/s).

kVelocidadVeneno: Velocidad del proyectil de veneno (80 m/s).

Funciones:

CalcularTiempoImpacto(double distancia, double velocidad): Calcula el tiempo de impacto con base en la distancia y velocidad.

Ejemplo de código

Entrada esperada: Indica qué datos debe ingresar el usuario (en este caso, la distancia al objetivo en metros).

Salida esperada: Muestra el resultado que debería imprimir el programa en función de la entrada dada.

Implementación con funciones: Presenta el código en C++ que realiza los cálculos, usando constantes y una función (CalcularTiempoImpacto) para hacer el cálculo del tiempo de impacto.

Errores comunes

Error: La distancia debe ser positiva.

Ocurre cuando el usuario ingresa un número negativo o cero.

Solución: Ingresar un valor mayor a 0.

Formato incorrecto de entrada

Ocurre cuando el usuario ingresa texto en lugar de un número.

Solución: Asegurarse de ingresar solo valores numéricos.

Tiempo incorrecto

Puede ser causado por un error en la ecuación de cálculo.

Solución: Verificar que $\text{tiempo} = \text{distancia} / \text{velocidad}$ esté implementado correctamente.

Licencia

Este proyecto se distribuye bajo la licencia MIT, lo que permite su uso, modificación y distribución con restricciones mínimas (La Licencia MIT es una licencia de software permisiva que permite a los desarrolladores utilizar, copiar, modificar y distribuir el software con pocas restricciones).

