

Parcial Pensamiento Algorítmico

Alejandro Moreno Lacharme

Juan Sebastián García Contreras

INFORME

Análisis del problema

Descripción del problema

El programa calcula el daño recibido por un personaje en función del ataque mágico infligido y su resistencia mágica. La resistencia se expresa en porcentaje y afecta la cantidad de daño total recibido. Dependiendo del valor de la resistencia, el daño se reduce en distintos niveles.

Requisitos Funcionales

Solicitar al usuario la cantidad de daño del ataque mágico (D).

Solicitar al usuario la resistencia mágica del personaje en porcentaje (r).

Validar que la resistencia mágica esté dentro del rango de 0 a 100.

Calcular el daño total recibido en función de la resistencia mágica:

Si la resistencia es menor a 50%, se reduce el daño en un 30%.

Si la resistencia está entre $50\% \leq R < 75\%$, se reduce el daño en función del porcentaje ingresado.

Si la resistencia es 75% o superior, el daño se reduce en un 90%. Mostrar el resultado del daño recibido.

Mostrar el resultado del daño recibido.

Requisitos No Funcionales

El programa debe ejecutarse en un entorno de consola.

Debe manejar errores de entrada (valores fuera del rango permitido).

La salida debe ser clara y fácil de entender para el usuario.

El cálculo del daño debe ser eficiente y preciso.

Análisis de casos de uso principales

El programa se utiliza para conocer la probabilidad de reducción de daño según la resistencia mágica que tiene personaje del WoW (World of Warcraft) al momento de ser atacado.

Si $R \geq 75\%$ → El personaje solo recibe 10% del daño total.

Si $50\% \leq R < 75\%$ → El daño se reduce exactamente en un $R\%$.

Si $R < 50\%$ → El daño se reduce en un 30%, sin importar el valor exacto de R .

Identificación de Entradas, Procesos y Salidas esperadas

Entrada: En ambos lenguajes se definen las variables (d y r)

Proceso: En ambos lenguajes se calcula la reducción de daño según la resistencia mágica:

1. Si $r < 50$, se aplica una reducción del 30%.
2. Si $50 \leq r < 75$, se aplica una reducción del $r\%$.
3. Si $r \geq 75$, se aplica una reducción del 90%.

Salida: En ambos lenguajes se evidenciará el daño recibido por el personaje o en su defecto mensaje de "Error: La resistencia está fuera del alcance permitido".

Justificación de la Solución

Estrategia para resolver el problema

Se utilizó el **if-else** para que en ambos lenguajes se pudiese determinar el daño final recibido según el porcentaje de resistencia mágica del personaje. Se implementa esta solución ya que es la forma más óptima en la cual se puede evidenciar el resultado.

Estructuras de datos y algoritmos seleccionados

Se definen variables como enteros (**int**) para representar valores de daño y resistencia. Se utilizan estructuras condicionales **if-else** para clasificar la resistencia mágica y calcular la reducción del daño.

Se prioriza la validación de entrada para evitar errores por valores fuera del rango permitido.

Comparación con posibles soluciones alternativas y razones elección final

La solución utilizada es la más efectiva debido al uso de condicionales (**if else**) fáciles de entender y aplicar en los diferentes lenguajes. Debido a que se pudo haber realizado con bucles (**loops**), sin embargo, la implementación de este tipo de proceso es largo y complicado porque tiene más líneas de código, esto lo hace difícil de leer, también con el propósito de evitar errores se evita su uso.

Diagrama UML

