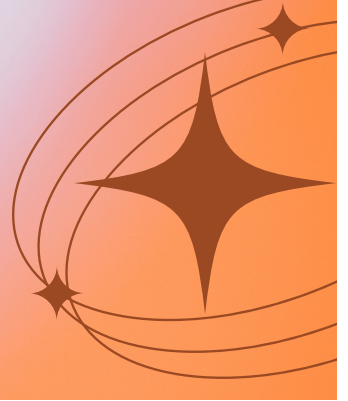




TRABAJO PRACTICO N°1

VALENTINA
R.A.BEJARANO C.



Problema N° 1

El primer ejemplo que se nos dio fue la resolución de un problema de programación siendo la incógnita encontrar la distancia entre Mario y Koopa.

Si tuviéramos que definir el rol que cumple cada parte entenderíamos que los datos de entrada serian las posiciones iniciales de Mario y Koopa, el dato de salida seria la distancia a recorrer para que el objeto 1 llegue al objeto 2, siendo el proceso $\text{posicionKoopa} - \text{posicionMario} = \text{distanciaMK}$ sabiendo que el resultado será la distancia recorrida

Entonces:

- Datos de entrada: posicionMario (inicial) posicionKoopa (inicial)
- Datos de salida: distanciaMK
- Proceso: $\text{posicionMario} - \text{posicionKoopa} = \text{distanciaMK}$

Ahora veamos como seria todo este problema resuelto

¿Quién tiene que resolver esto? Sabemos que los que tendrían que hacer la resolución a la hora de tener un producto terminado seria el usuario, aunque esto puede llegar a variar

¿Cuáles son las variables dentro de este problema? Para saber cuales son y como se definen sabemos que estamos trabajando con dos elementos, la posicionMario y posicionKoopa contienen los datos que debemos restar para obtener la distanciaMK que contiene el resultado final de la resta

¿Que algoritmo podriamos usar para resolver esto? Para que el programa (processing) es colocando directamente lo que queremos realizar, calcular la distancia (calcularDistancia) siendo que el programa primeramente leyendo las posiciones iniciales, resolviendo la resta entre estas dando en la consola el resultado final, terminando con el problema

Entidad que resuelve el problema: Usuario

Variables: posicionMario (inicial) posicionKoopa (inicial)

contienen lo numeros a restar.

distanciaMK

resultado final de la resta

Nombre algoritmo: CalcularDistancia

1. Se leen las posiciones iniciales
2. $\text{posicionMario} - \text{posicionKoopa}$ //Restar las distancias entre si
3. Obtener el resultado
4. Leer la distancia final