



## Fundamentos de Programación

### Tema 2 Introducción a la programación

#### A2.2 Introducción a la programación

Nombre del estudiante: **Fabian Cruz Hernández**  
Grupo: **AED-1285\_NUE\_SVC\_VET\_CDV\_HCG\_UZA\_TL1\_2025C**  
Nombre del Docente: **Cupertino Luna Trejo**  
Fecha de Entrega: **17 – diciembre –2025**

1. Para los siguientes ejercicios realiza el análisis y su correspondiente algoritmo, así como la codificación en Lenguaje C

### Problema 1

En las olimpiadas de invierno el tiempo que realizan los participantes en la competencia de velocidad en pista se mide en minutos, segundos y centésimas. La distancia que recorren se expresa en metros. Calcule la velocidad de los participantes en kilómetros por hora de las diferentes competencias.

Consideraciones:

El tiempo debemos expresarlo en segundos, y para hacerlo aplicamos la siguiente fórmula:

$$TSE = MIN \times 60 + SEG + \frac{CEN}{100}$$

Luego podemos calcular la velocidad expresada en metros sobre segundos (m/s):

$$VMS = \frac{DIS \text{ (Metros)}}{TSE \text{ (Segundos)}}$$

Para obtener la velocidad en kilómetros por hora (K/h), aplicamos la siguiente fórmula:

$$VKH = \frac{VMS \times 3600 \text{ (Kilómetros)}}{1000 \text{ (hora)}}$$

#### a. Análisis del problema

(Realiza el análisis del problema: identifica entradas, proceso y salidas)



<b>Entradas</b>	Minutos, segundos, centésimas y la distancia
<b>Proceso</b>	Convertir el tiempo total a segundos, calcular la velocidad en m/s y después convertir a km/h
<b>Salidas</b>	Velocidad en Km/h

### b. Solución en pseudocódigo

(Estructura de la solución de cada problema mediante un algoritmo. Elaborado a mano o con alguna herramienta digital como PSeInt)

```

1  Algoritmo VelocidadOlimpiadas
2      Definir minutos, segundos, centesimas, distancia, tseg, vms, vkh como Real
3
4      Escribir "Ingrese el tiempo del participante:"
5      Escribir "Minutos: "
6      Leer minutos
7      Escribir "Segundos: "
8      Leer segundos
9      Escribir "Centesimas de segundo: "
10     Leer centesimas
11     Escribir "Ingrese la distancia recorrida (metros): "
12     Leer distancia
13
14     tseg = (minutos * 60) + segundos + (centesimas / 100)
15     vms = distancia / tseg
16     vkh = vms * 3.6
17
18     Escribir "Velocidad: ", vkh, " km/h"
19
20 FinAlgoritmo

```

### c. Codificación de la solución

(Elaborar un programa en Lenguaje C para codificar la solución en DevC++).



```
1  #include <stdio.h>
2
3  int main()
4  {
5      float minutos, segundos, centesimas;
6      float distancia, t_segundos, v_ms, v_kmh;
7
8      printf("Ingrese el tiempo del participante:\n");
9      printf("  Minutos: ");
10     scanf("%f", &minutos);
11
12     printf("  Segundos: ");
13     scanf("%f", &segundos);
14
15     printf("  Centesimas de segundo: ");
16     scanf("%f", &centesimas);
17
18     printf("\nIngrese la distancia recorrida (metros): ");
19     scanf("%f", &distancia);
20
21     t_segundos = (minutos * 60) + segundos + (centesimas / 100);
22     v_ms = distancia / t_segundos;
23     v_kmh = v_ms * 3.6;
24
25     printf("  Velocidad:      %.2f km/h\n", v_kmh);
26
27     return 0;
28 }
```

● **fabian@fchDev:**~/fundamentosDeProgramacion/Tema2/Act2/output\$ ./"1"

Ingrese el tiempo del participante:

Minutos: 1

Segundos: 23

Centesimas de segundo: 45

Ingrese la distancia recorrida (metros): 500

Velocidad: 21.57 km/h

○ **fabian@fchDev:**~/fundamentosDeProgramacion/Tema2/Act2/output\$

## Problema 2

En una miscelánea se requiere que dado el peso en gramos (máximo 10000) se determine el número de pesas que hay que poner en una balanza cuyos pesos sean de 1000g, 500g, 200g, 100g, 50g, 10g, 5g, 2g y 1g, y que permitan equilibrar el peso...

### a. Análisis del problema

(Realiza el análisis del problema: identifica entradas, proceso y salidas)

<b>Entradas</b>	El peso en gramos
<b>Proceso</b>	Usar un algoritmo de descomposición de divisiones sucesivas, por cada tipo de pesa calcular cuantas veces cabe en el peso restante.
<b>Salidas</b>	La cantidad necesaria de pesas para equilibrar los pesos.

### b. Solución en pseudocódigo



```
1  Algoritmo DistribucionPesas
2      Definir peso_total, num, i Como Entero
3      Definir pesas Como Entero
4      Dimensionar pesas[9]
5
6      Definir nombres Como Cadena
7      Dimensionar nombres[9]
8
9      // Inicializar valores y nombres de pesas
10     pesas[1] = 1000
11     pesas[2] = 500
12     pesas[3] = 200
13     pesas[4] = 100
14     pesas[5] = 50
15     pesas[6] = 10
16     pesas[7] = 5
17     pesas[8] = 2
18     pesas[9] = 1
19
20     nombres[1] = "1000 g"
21     nombres[2] = "500 g"
22     nombres[3] = "200 g"
23     nombres[4] = "100 g"
24     nombres[5] = "50 g"
25     nombres[6] = "10 g"
26     nombres[7] = "5 g"
27     nombres[8] = "2 g"
28     nombres[9] = "1 g"
29
30     // Entrada del usuario
31     Escribir "Ingrese el peso en gramos (max 10000):"
32     Leer peso_total
33
34     Si (peso_total ≤ 0 o peso_total > 10000 ) Entonces
35         Escribir "Peso fuera de rango"
36     FinSi
37
38     // Calcular distribución de pesas
39     Para i = 1 Hasta 9 Hacer
40         num = Trunc(peso_total / pesas[i])
41         Escribir nombres[i], ": ", num
42         peso_total = peso_total MOD pesas[i]
43     FinPara
44     FinAlgoritmo
45
```

### c. Codificación de la solución

(Elaborar un programa en Lenguaje C para codificar la solución en DevC++).





```
6 #include <stdio.h>
7
8 int main()
9 {
10     int peso_total;
11
12     // Entrada
13     printf("Ingrese el peso en gramos (max 10000): ");
14     scanf("%d", &peso_total);
15
16     // Validación
17     if (peso_total < 0 || peso_total > 10000)
18     {
19         printf("Peso fuera de rango.\n");
20         return 1;
21     }
22
23     // Arreglo de pesas disponibles
24     int pesas[] = {1000, 500, 200, 100, 50, 10, 5, 2, 1};
25     int n = sizeof(pesas) / sizeof(pesas[0]);
26
27     printf("Distribución de pesas:\n");
28
29     // Cálculo de cuántas pesas de cada tipo
30     for (int i = 0; i < n; i++)
31     {
32         int num = peso_total / pesas[i]; // cuántas pesas de este tipo
33         printf("%d g: %d\n", pesas[i], num);
34         peso_total = peso_total % pesas[i]; // residuo para la siguiente pesa
35     }
36
37     return 0;
38 }
```

● **fabian@fchDev:~/fundamentosDeProgramacion\$** cd "/home/fabian/funda

● **fabian@fchDev:~/fundamentosDeProgramacion/Tema2/Act2/output\$** ./"2

Ingrese el peso en gramos (max 10000): 3680

Distribución de pesas:

1000 g: 3

500 g: 1

200 g: 0

100 g: 1

50 g: 1

10 g: 3

5 g: 0

2 g: 0

1 g: 0

○ **fabian@fchDev:~/fundamentosDeProgramacion/Tema2/Act2/output\$**

Adjunto mi repositorio;

<https://github.com/FabianCruz14/Fundamentos-de-Programacion-TecNM-Huau/tree/7570dc4128c0b06b70b4ec0545947e5f10510125/Tema2/Act2>





