**Problema 1.**

**En esta unidad, se hace énfasis en lo importante que resulta no empezar el desarrollo de programas directamente tecleando código. ¿Estás de acuerdo con esta afirmación? ¿Qué otras cosas deberían hacerse antes? ¿En qué fases?**

Cuando uno intenta resolver un problema escribiendo directamente el código hace lugar a la perdida de información, tiempo y precisión. La situación es similar a intentar cocinar un platillo nuevo sin haber leído su receta con antelación, no sabes lo que va a pasar. Por tanto, existen ocho fases que mejoran las posibilidades de desarrollar un algoritmo óptimo para la solución de un problema.

1. Definición del problema: Establecer concretamente el problema a resolver

2. Análisis del problema: Entender qué pregunta y qué información podemos utilizar

3. Diseño de la solución del problema: Idear una manera de resolver el problema que se plantea, uno se puede apoyar de pseudocódigos y diagramas de flujo

4. Codificación: Escribir el código de la solución del problema

5. Compilación, prueba y depuración: Probar el código hasta intentar hallar su mejor versión

6. Documentación. Comentar las partes importantes del proceso dentro del código

7. Implantación del programa. Utilizar el programa para resolver el problema

8. Mantenimiento del programa. Estar al pendiente de mejoras o modificaciones

**Problema 2.**

**¿Qué relación existe entre los algoritmos y las estructuras de datos? ¿Cuál va primero? ¿Cuál es más importante en la resolución de problemas? ¿Qué papel juega cada uno?**

Las estructuras de datos son dos piezas fundamentales para la creación de programas organizados, y para que funcionen se necesitan de ambas partes, las cuales están estrechamente relacionadas. Por tanto, ninguna es más importante que la otra, sino que ambas son necesarias.

Los algoritmos son secuencias de pasos que transforman los datos que conocemos en información que podemos ocupar. Sin embargo, para que un algoritmo funcione de manera óptima. Los datos deben presentarse de forma ordenada, lo cual se logra con la implementación de estructuras de datos.

**Problema 3.**

**Enumera las propiedades fundamentales de un algoritmo. Ejemplifica con un código fuente sencillo, ¿se pueden demostrar formalmente las propiedades? ¿Y si es pseudocódigo?**

1. Simplicidad
2. Datos de Entrada
3. Información de salida
4. Proceso definido
5. Orden lógico
6. Precisión
7. Tener principio y fin
8. Efectividad

Programa para calcular el IVA de un producto

#include <stdio.h>

#include <stdlib.h>

int main ()

{

float neto, precio, iva;

printf ("Introduzca el precio sin iva de un producto:");

scanf ("%f", &precio);

iva=(16\*precio/100);

neto=precio + iva;

printf ("\n El precio sin IVA es de: %f\n", precio);

printf ("\n El precio con IVA es de: %f\n", neto);

return 0;

}

Pseudocódigo

**Flotante:** neto, precio, iva

**Inicio**

Escribir ("Introduzca el precio sin iva de un producto:");

leer(precio)

iva <- 16\*precio/100

neto <- precio + iva

escribir(" El precio sin IVA es de: , precio)

escribir(" El precio con IVA es de: , neto)

fin

En ambas presentaciones del algoritmo es posible identificar sus características con facilidad. Se presenta la siguiente tabla para ejemplificar.

|  |  |
| --- | --- |
| Simplicidad | El proceso se muestra en código y pseudocódigos breves |
| Datos de entrada | Precio del producto sin iva |
| Información de salida | Precio del producto sin iva y neto |
| Proceso definido | Se calcula el iva de un producto y se muestra en pantalla |
| Orden lógico | 1. Se conoce el precio del producto 2. Se calcula el iva del producto respectivo 3. Se suma el iva al precio del producto   En otro orden no es posible calcular |
| Tener principio y fin | Inicio: definición de variables  Fin: Precio mostrado en pantalla |
| Efectividad | Resultados precisos |

**Problema 4.**

**Diseñe y elabore un algoritmo en diagrama de flujo y pseudocódigo que visualice y calcule el factorial de un número entero.**

Diagrama de Flujo Pseudocódigo

Diagrama

Descripción generada automáticamente

**entero:** factorial

**inicio**

escribir(“Ingresa el numero cuyo factorial deseas conocer”)

leer(n)

i<- 1

factorial <- 1

**mientras**( i<=n)

factorial \*=1

i+=1

**fin mientras**

escribir(“El factorial de ”, n, “ es ”, factorial)

**fin**

**Problema 5.**

**Diseñe un algoritmo en diagrama de flujo y pseudocódigo que solicite la edad de 2 hermanos y muestre un mensaje indicando la edad del mayor y cuantos años de diferencia tiene con el menor.**

Diagrama de Flujo Pseudocódigo

Diagrama

Descripción generada automáticamente

**entero:** Edad1, Edad2, Diferencia

**inicio**

escribir(“Ingresa la edad del hermano 2”)

leer(Edad1)

escribir(“Ingresa la edad del hermano 1”)

leer(Edad2)

diferencia <- abs(Edad1-Edad2)

**si**(Edad1>Edad2)

escribir( “El hermano 1 de edad es mayor, Edad: ”, Edad)

**fin si**

escribir( “El hermano 2 es mayor. Edad: ”, Edad2)

escribir(“Años de diferencia”, Diferencia)

**fin**