

# **Curso de Programación Desde Cero**

## **Tema 1: Introducción a la Programación**

Duración estimada: 8 horas

### **1. ¿Qué es la Programación?**

La programación es el proceso de crear instrucciones para que un ordenador realice tareas específicas. Es como escribir una receta muy detallada que la máquina pueda seguir paso a paso.

### **2. Historia y Evolución**

Desde los primeros ordenadores mecánicos hasta los lenguajes modernos, la programación ha evolucionado drásticamente:

- 2 1940s: Primeros ordenadores programables
- 2 1950s: Aparecen los primeros lenguajes de alto nivel
- 2 1970s-80s: C, Pascal y programación estructurada
- 2 1990s: Internet y lenguajes web (HTML, JavaScript)
- 2 2000s: Programación orientada a objetos masiva
- 2 2010s-presente: Cloud computing y desarrollo Agile

### **3. Pensamiento Computacional**

Es la habilidad de resolver problemas de manera sistemática:

- 2 Análisis: Entender el problema completamente
- 2 Descomposición: Dividir en partes más pequeñas
- 2 Abstracción: Identificar patrones y conceptos clave
- 2 Algoritmos: Crear secuencias lógicas de pasos

## 4. Algoritmos: La Base de Todo Programa

Un algoritmo es una secuencia finita de instrucciones bien definidas que resuelven un problema. Características principales:

- 2 Finitud: Debe terminar en un número finito de pasos
- 2 Precisión: Cada paso debe estar claramente definido
- 2 Entrada: Puede tener cero o más datos de entrada
- 2 Salida: Debe producir al menos un resultado
- 2 Efectividad: Debe ser realizable en tiempo razonable

### Ejemplo Práctico: Algoritmo para Preparar Cafè

1. INICIO
2. Verificar si hay cafè, agua y electricidad
3. SI no hay alguno, IR a comprarlo
4. Llenar la cafetera con agua
5. Añadir café molido al filtro
6. Encender la cafetera
7. Esperar hasta que termine
8. Servir en taza
9. FIN

## 5. Diagramas de Flujo

Representación gráfica de algoritmos usando símbolos:

- 2 Óvalo: Inicio y fin del proceso
- 2 Rectángulo: Proceso o acción
- 2 Rombo: Decisión (sí/no)
- 2 Paralelogramo: Entrada o salida de datos
- 2 Flechas: Flujo de control

## 6. Pseudocódigo

Descripción de algoritmos en lenguaje natural estructurado:

```
ALGORITMO CalcularPromedio
VARIABLES: nota1, nota2, nota3, promedio
INICIO
    ESCRIBIR "Ingrese la primera nota: "
    LEER nota1
    ESCRIBIR "Ingrese la segunda nota: "
    LEER nota2
    ESCRIBIR "Ingrese la tercera nota: "
    LEER nota3
    promedio = (nota1 + nota2 + nota3) / 3
    ESCRIBIR "El promedio es: ", promedio
FIN
```

## 7. Tipos de Datos BÁSICOS

Los datos son la materia prima de los programas. Tipos principales:

### Números Enteros (int):

2 Representan cantidades completas: -5, 0, 42, 1000

2 Usos: contadores, edades, cantidades

### Números Decimales (float/double):

2 Representan cantidades con fracciones: 3.14, -2.5, 0.001

2 Usos: precios, medidas, cálculos científicos

### Texto (string):

2 Secuencias de caracteres: "Hola", "123abc", "usuario@email.com"

2 Usos: nombres, mensajes, URLs

### Booleanos (bool):

2 Solo dos valores: verdadero (true) o falso (false)

2 Usos: condiciones, estados, validaciones

## 8. Variables y Constantes

Las variables son contenedores que almacenan datos que pueden cambiar:

Ejemplos en pseudocódigo:

```
edad = 25           // Variable entera
nombre = "Ana"     // Variable de texto
precio = 19.99     // Variable decimal
activo = verdadero // Variable booleana
```

Las constantes son valores que NO cambian durante la ejecución:

```
PI = 3.14159        // Constante matemática
IVA = 0.21          // Constante fiscal
EMPRESA = "Mi Empresa" // Constante de texto
```

## 9. Reglas para Nombrar Variables

2 Usar nombres descriptivos: "edad" en lugar de "e"

2 No espacios: usar guiones bajos o camelCase

2 No empezar con números: "usuario1" no "1usuario"

2 Evitar caracteres especiales: solo letras, números y \_

2 Ser consistente en el estilo a lo largo del programa

## 10. Operadores Aritméticos

Permiten realizar cálculos matemáticos con números:

- 2 Suma (+):  $5 + 3 = 8$
- 2 Resta (-):  $10 - 4 = 6$
- 2 Multiplicación (\*):  $6 * 7 = 42$
- 2 División (/):  $15 / 3 = 5$
- 2 Módulo (%):  $17 \% 5 = 2$  (resto de la división)

### Ejemplo Práctico - Calculadora de Propinas:

```
cuentaRestaurante = 45.50  
porcentajePropina = 15  
propina = cuentaRestaurante * porcentajePropina / 100  
totalPagar = cuentaRestaurante + propina  
// Resultado: propina = 6.83, total = 52.33
```

## 11. Operadores de Comparación

Comparan valores y devuelven verdadero o falso:

- 2 Igual a (==):  $5 == 5$  es verdadero
- 2 Diferente de (!=):  $5 != 3$  es verdadero
- 2 Mayor que (>):  $10 > 7$  es verdadero
- 2 Menor que (<):  $3 < 8$  es verdadero
- 2 Mayor o igual ( $\geq$ ):  $5 \geq 5$  es verdadero
- 2 Menor o igual ( $\leq$ ):  $4 \leq 9$  es verdadero

## 12. Operadores Lógicos

Combinan condiciones booleanas:

### AND (&&) - Y lógico:

- 2 verdadero AND verdadero = verdadero
- 2 Cualquier otra combinación = falso

### OR (||) - O lógico:

- 2 falso OR falso = falso
- 2 Cualquier otra combinación = verdadero

### NOT (!) - Negación:

- 2 NOT verdadero = falso
- 2 NOT falso = verdadero

### Ejemplo Práctico:

```
edad = 20  
tieneCarnet = verdadero  
puedeConducir = (edad  $\geq$  18) AND tieneCarnet  
// Resultado: verdadero
```

## 13. Ejercicios Prácticos

### Ejercicio 1: Conversor de Temperatura

Crear un algoritmo que convierta grados Celsius a Fahrenheit:

ALGORITMO ConvertirTemperatura

VARIABLES: celsius, fahrenheit

INICIO

    ESCRIBIR "Ingrese temperatura en Celsius: "

    LEER celsius

    fahrenheit = (celsius \* 9/5) + 32

    ESCRIBIR celsius, "°C son ", fahrenheit, "°F"

FIN

### Ejercicio 2: Calculadora de IMC

Crear un algoritmo que calcule el Índice de Masa Corporal:

ALGORITMO CalcularIMC

VARIABLES: peso, altura, imc

INICIO

    ESCRIBIR "Ingrese su peso en kg: "

    LEER peso

    ESCRIBIR "Ingrese su altura en metros: "

    LEER altura

    imc = peso / (altura \* altura)

    ESCRIBIR "Su IMC es: ", imc

    SI imc < 18.5 ENTONCES

        ESCRIBIR "Bajo peso"

    SINO SI imc < 25 ENTONCES

        ESCRIBIR "Peso normal"

    SINO SI imc < 30 ENTONCES

        ESCRIBIR "Sobrepeso"

    SINO

        ESCRIBIR "Obesidad"

    FIN SI

FIN

### Ejercicio 3: Verificador de Año Bisiesto

Un año es bisiesto si es divisible por 4, excepto los años

divisibles por 100, a menos que también sean divisibles por 400:

ALGORITMO VerificarBisiesto

VARIABLES: año, esBisiesto

INICIO

    ESCRIBIR "Ingrese un año: "

    LEER año

    esBisiesto = ((año % 4 == 0) AND (año % 100 != 0))

        OR (año % 400 == 0)

    SI esBisiesto ENTONCES

## 14. Buenas PrÆcticas en ProgramaciÓn

### Principio DRY (Don't Repeat Yourself)

Evitar duplicar cÓdigo. Si necesitas hacer algo mÆs de una vez, crØalo como una funciÓn reutilizable.

### Nomenclatura Clara y Consistente

- 2 Usar nombres descriptivos: "calcularPrecioTotal" no "calc"
- 2 Ser consistente: si usas camelCase, usalo siempre
- 2 Evitar abreviaciones confusas: "usuario" no "usr"

### Comentarios y DocumentaciÓn

- 2 Explicar QUÉ hace el cÓdigo, no CÓMO
- 2 Comentar lÓgica compleja o decisiones de diseÑo
- 2 Mantener comentarios actualizados con el cÓdigo

### Manejo de Errores

- 2 Anticipar quØ puede salir mal
- 2 Validar datos de entrada
- 2 Proporcionar mensajes de error Ótiles

### Testing y VerificaciÓn

- 2 Probar con diferentes tipos de datos
- 2 Verificar casos extremos (valores muy grandes, muy pequeÑos)
- 2 Asegurarse de que el programa maneja errores graciosamente

## 15. IntroducciÓn a las Funciones

Las funciones son bloques de cÓdigo reutilizables que realizan una tarea específica. Beneficios principales:

- 2 ReutilizaciÓn: Escribe una vez, usa muchas veces
- 2 OrganizaciÓn: Divide problemas complejos en partes mÆs simples
- 2 Mantenimiento: Cambios en un lugar afectan todo el programa
- 2 Testing: MÆs fÆcil probar piezas pequeÑas independientes

### Ejemplo de FunciÓn:

```
FUNCION calcularAreaCirculo(radio)
    PI = 3.14159
    area = PI * radio * radio
    RETORNAR area
FIN FUNCION

// Uso de la funciÓn:
radioCirculo = 5
superficieCirculo = calcularAreaCirculo(radioCirculo)
ESCRIBIR "El Área es: ", superficieCirculo
```

# 16. Resumen y Próximos Pasos

## Conceptos Clave Aprendidos

- 2 La programación es resolver problemas paso a paso
- 2 Los algoritmos son secuencias lógicas de instrucciones
- 2 Los datos tienen tipos: números, texto, booleanos
- 2 Las variables almacenan datos que pueden cambiar
- 2 Los operadores permiten manipular y comparar datos
- 2 El pensamiento computacional es fundamental
- 2 Las buenas prácticas evitan problemas futuros

## Habilidades Desarrolladas

- 2 Analizar problemas y descomponerlos en partes
- 2 Crear algoritmos usando pseudocódigo
- 2 Diseñar diagramas de flujo
- 2 Entender tipos de datos y operadores
- 2 Aplicar lógica para resolver problemas reales

## Aplicaciones Prácticas

Los conceptos aprendidos se aplican en:

- 2 Desarrollo de aplicaciones móviles
- 2 Creación de sitios web interactivos
- 2 Automatización de tareas repetitivas
- 2 Análisis de datos y estadísticas
- 2 Inteligencia artificial y machine learning
- 2 Videojuegos y simulaciones

## Preparación para el Tema 2

En el próximo tema abordaremos:

- 2 HTML: Estructura de páginas web
- 2 CSS: Estilización y diseño visual
- 2 Creación de tu primera página web
- 2 Formularios interactivos
- 2 Diseño responsive para móviles

## Recomendaciones de Estudio

- 2 Practica los ejercicios hasta dominarlos completamente
- 2 Crea tus propios problemas y resuelve con algoritmos
- 2 Revisa los conceptos antes de avanzar al siguiente tema
- 2 Busca ejemplos adicionales en situaciones cotidianas
- 2 Forma grupos de estudio para discutir soluciones

Este material ha sido diseñado para proporcionar una base sólida en los fundamentos de la programación. La práctica constante y la aplicación de estos conceptos será clave para tu éxito como desarrollador. ¡Continúa con entusiasmo hacia el Tema 2!