Lógica de Programación: Fundamentos y Aplicaciones

Julian F. Latorre

8 de octubre de 2024

Contenido

¿Qué es la lógica de programación?

- Conjunto de habilidades y técnicas para resolver problemas
- Base fundamental para cualquier desarrollador
- Independiente del lenguaje o tecnología

Importancia de la lógica de programación

- Resolución eficiente de problemas
- Facilita el aprendizaje de nuevos lenguajes
- Mejora el pensamiento crítico y analítico
- Base para crear algoritmos efectivos
- Ayuda a escribir código limpio y mantenible

Algoritmos

- Serie de pasos ordenados y finitos
- Describen cómo resolver un problema o realizar una tarea

Algoritmos

- Serie de pasos ordenados y finitos
- Describen cómo resolver un problema o realizar una tarea

Ejemplo: Algoritmo para hacer una taza de té

1. Hervir agua en una tetera
2. Colocar una bolsa de té en una taza
3. Verter el agua caliente sobre la bolsa de té
4. Esperar 3-5 minutos
5. Retirar la bolsa de té
6. Agregar azúcar o leche al gusto (opcional)
7. Revolver y disfrutar

Pseudocódigo

- Lenguaje intermedio entre lenguaje natural y código de programación
- Útil para planificar y estructurar la lógica

Pseudocódigo

- Lenguaje intermedio entre lenguaje natural y código de programación
- Útil para planificar y estructurar la lógica

Ejemplo: Calcular el área de un triángulo

```
INICIO
    LEER base
    LEER altura
    area = (base * altura) / 2
    MOSTRAR "El área del triángulo es: " + area
FIN
```

Diagramas de flujo

- Representaciones gráficas de algoritmos
- Utilizan símbolos para mostrar pasos y decisiones

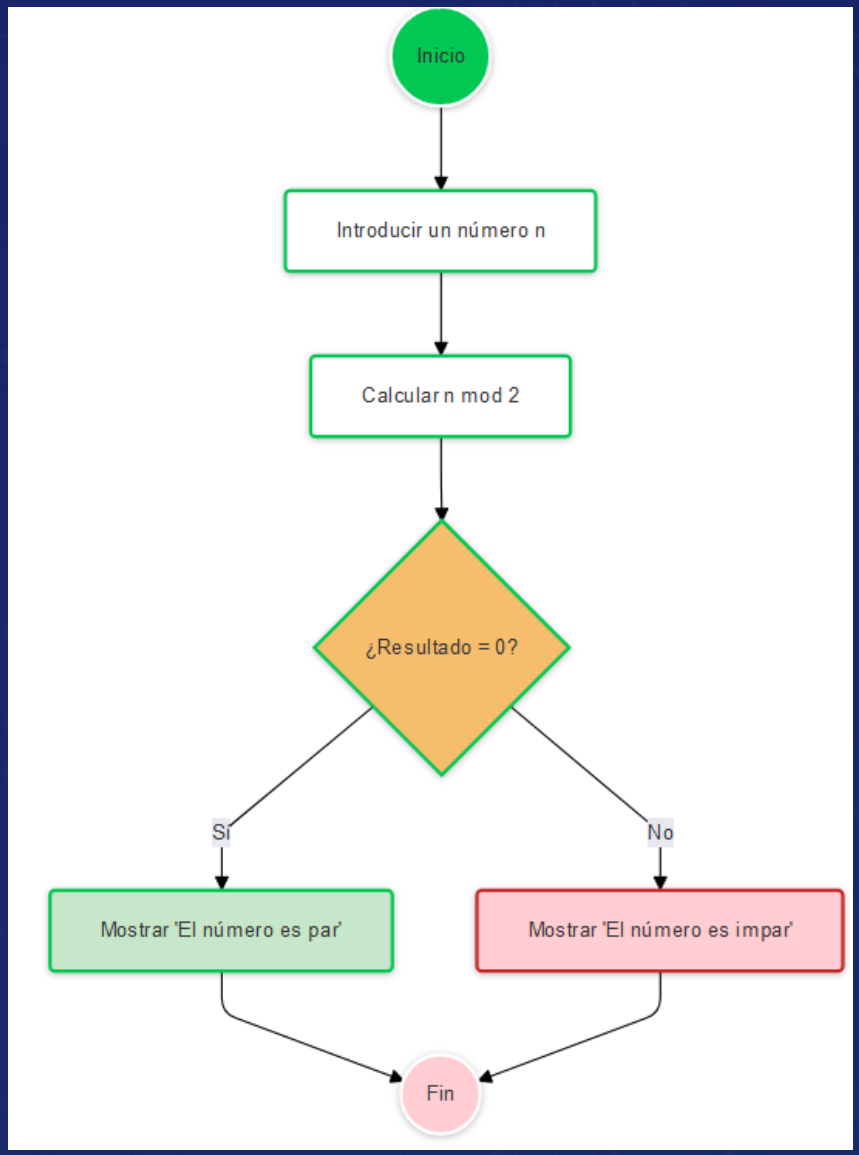

Figura: Ejemplo: Determinar si un número es par o impar

Tipos de datos

- Enteros (int): 1, -5, 100
- Flotantes (float): 3.14, -0.5, 2.0
- Cadenas (string): "Hola mundo"
- Booleanos (bool): true, false

Variables

- Contenedores que almacenan datos en la memoria
- Se utilizan para guardar y manipular información

Variables

- Contenedores que almacenan datos en la memoria
- Se utilizan para guardar y manipular información

Ejemplo: Declaración de variables

```
edad = 25
nombre = "Juan"
altura = 1.75
esMayorDeEdad = true
```

Operadores

- Aritméticos: +, -, *, /, % (módulo)
- De comparación: ==, !=, ¡, ¿, ¡=, ¿=
- Lógicos: AND, OR, NOT

Operadores

- Aritméticos: +, -, *, /, % (módulo)
- De comparación: ==, !=, ¡, ¿, ¡=, ¿=
- Lógicos: AND, OR, NOT

Ejemplo: Uso de operadores

```
suma = 5 + 3    suma = 8
esMayor = edad > 18    esMayor = true (si edad = 25)
esValido = (x > 0) AND (y < 10)
```

Secuencia

- Estructura más básica
- Ejecuta instrucciones una tras otra

Secuencia

- Estructura más básica
- Ejecuta instrucciones una tras otra

Ejemplo: Secuencia en pseudocódigo

```
INICIO
    LEER nombre
    LEER edad
    MOSTRAR "Hola, " + nombre
    MOSTRAR "Tienes " + edad + " años"
FIN
```

Selección (condicionales)

- Permiten ejecutar diferentes bloques de código según condiciones

Selección (condicionales)

- Permiten ejecutar diferentes bloques de código según condiciones

Ejemplo: Estructura if-else

```
SI edad >= 18 ENTONCES
    MOSTRAR "Eres mayor de edad"
SINO
    MOSTRAR "Eres menor de edad"
FIN SI
```

Repetición (bucles)

- Permiten ejecutar un bloque de código múltiples veces

Repetición (bucles)

- Permiten ejecutar un bloque de código múltiples veces

Ejemplo: Bucle while

```
contador = 1
MIENTRAS contador <= 5 HACER
    MOSTRAR contador
    contador = contador + 1
FIN MIENTRAS
```

Análisis del problema

- Identificar los datos de entrada
- Determinar el resultado esperado
- Reconocer las restricciones o condiciones especiales

Diseño de la solución

- Dividir el problema en subproblemas más pequeños
- Crear un algoritmo o pseudocódigo para cada subproblema
- Combinar las soluciones de los subproblemas

Implementación y pruebas

- Implementar la solución en un lenguaje de programación
- Realizar pruebas para asegurar su correcto funcionamiento

Ejercicio 1: Calculadora de IMC

Pseudocódigo

```
INICIO
    LEER peso
    LEER altura
    IMC = peso / (altura * altura)
    MOSTRAR "Tu IMC es: " + IMC

    SI IMC < 18.5 ENTONCES
        MOSTRAR "Bajo peso"
    SINO SI IMC >= 18.5 Y IMC < 25 ENTONCES
        MOSTRAR "Peso normal"
    SINO SI IMC >= 25 Y IMC < 30 ENTONCES
        MOSTRAR "Sobrepeso"
    SINO
        MOSTRAR "Obesidad"
    FIN SI
FIN
```

Ejercicio 2: Suma de números pares

Pseudocódigo

```
INICIO
    suma = 0
    PARA i DESDE 1 HASTA 100 HACER
        SI i % 2 == 0 ENTONCES
            suma = suma + i
        FIN SI
    FIN PARA
    MOSTRAR "La suma de los números pares del 1 al 100 es: " + suma
FIN
```

Conclusión

- La lógica de programación es fundamental para el desarrollo de software
- Dominar estos conceptos prepara el camino para aprender lenguajes específicos
- La práctica constante es clave para mejorar tus habilidades
- Continúa resolviendo problemas y creando algoritmos