

Interpretando la lógica de programación en ejercicios cotidianos

En la programación es importante saber interpretar lo que día a día los usuarios solicitan en un lenguaje común, para poder transformarlo en líneas de código y soluciones efectivas. Esta habilidad de interpretación no solo facilita la comunicación entre desarrolladores y usuarios, sino que también es crucial para garantizar que el producto final cumpla con las expectativas y necesidades reales. A lo largo de este documento, exploraremos cómo desglosar y comprender ejercicios de programación, lo que te permitirá abordar problemas de manera estructurada y eficiente, transformando requerimientos abstractos en implementaciones concretas y funcionales.

Comprensión del Enunciado

Lectura Atenta

Ejemplo: Supongamos que el enunciado es: "Escribe un programa que determine si un número es primo."

- Primera Lectura: Lee el enunciado completo para obtener una visión general.
- Segunda Lectura: Subraya palabras clave como "programa", "determinar", "número", y "primo".

Identificación de Palabras Clave

- Acciones: "Determinar" indica que necesitamos una función de verificación.
- Restricciones: No hay restricciones explícitas, pero implícitamente se espera eficiencia.

Preguntas Clave

- Clarificación: ¿Qué rango de números debemos considerar? ¿Cómo manejamos el número 1?
- Suposiciones: Suponemos que el número de entrada es un entero positivo.

Descomposición del Problema

Dividir en Partes

- Componentes:
 1. Leer el número de entrada.
 2. Verificar si el número es primo.
 3. Imprimir el resultado.

Identificación de Entradas y Salidas

- Entradas: Un número entero.
- Salidas: Un mensaje indicando si el número es primo o no.

Requisitos y Restricciones

- Condiciones: Un número primo es mayor que 1 y solo divisible por 1 y por sí mismo.
- Límites: Considerar eficiencia para números grandes.

Planificación de la Solución

Algoritmo Inicial

- Si el número es menor o igual a 1, no es primo.
- Verificar divisibilidad desde 2 hasta la raíz cuadrada del número.

Diagrama de Flujo

1. Inicio: Comienza el diagrama de flujo.
2. Entrada del Usuario:
 - Mostrar un mensaje: "Introduce un número".
 - Leer el número de entrada.
3. Verificar Conversión a Entero:
 - Decisión: ¿La entrada es un número entero válido?
 - Sí: Continuar al siguiente paso.
 - No: Mostrar "Por favor, introduce un número válido" y volver a la entrada del usuario.
4. Verificar si el Número es Menor o Igual a 1:
 - Decisión: ¿El número es menor o igual a 1?
 - Sí: Mostrar "El número no es primo" y terminar.
 - No: Continuar al siguiente paso.
5. Inicializar el Contador: Establecer $i = 2$.
6. Iteración:
 - Decisión: ¿ i es menor o igual a la raíz cuadrada del número?

- Sí: Continuar al siguiente paso.
- No: Mostrar "El número es primo" y terminar.

7. Verificar Divisibilidad:

- Decisión: ¿El número es divisible por i?
 - Sí: Mostrar "El número no es primo" y terminar.
 - No: Incrementar i en 1 y volver al paso de iteración.

8. Fin: Termina el diagrama de flujo.

Codificación

```
using System;
```

```
class Program
```

```
{
```

```
    static void Main()
```

```
    {
```

```
        Console.Write("Introduce un número: ");
```

```
        int numero;
```

```
        // Intentar convertir la entrada a un número entero
```

```
        if (int.TryParse(Console.ReadLine(), out numero))
```

```
        {
```

```
            if (EsPrimo(numero))
```

```
            {
```

```
                Console.WriteLine($"{numero} es primo.");
```

```
            }
```

```
        else
```

```
        {
```

```
            Console.WriteLine($"{numero} no es primo.");
```

```
    }  
}  
else  
{  
    Console.WriteLine("Por favor, introduce un número válido.");  
}  
}
```

```
static bool EsPrimo(int n)  
{  
    if (n <= 1)  
    {  
        return false;  
    }  
  
    for (int i = 2; i <= Math.Sqrt(n); i++)  
    {  
        if (n % i == 0)  
        {  
            return false;  
        }  
    }  
  
    return true;  
}  
}
```

Ejercicios de Practica

1. Calculadora de Propinas:

- Imagina que estás en un restaurante y quieres calcular la propina.
- Crea un programa que pida al usuario el total de la cuenta y el porcentaje de propina que desea dejar.
- Calcula la cantidad de propina y el total a pagar, y muestra ambos valores.

2. Conversión de Altura:

- Supón que estás comprando muebles y necesitas saber si caben en tu casa.
- Escribe un programa que solicite al usuario la altura de un mueble en metros.
- Convierte esa altura a centímetros para facilitar la comparación con las medidas de tu casa.

3. Recordatorio de Cumpleaños:

- Quieres asegurarte de no olvidar el cumpleaños de un amigo.
- Desarrolla un programa que pida al usuario el nombre de su amigo y su fecha de cumpleaños.
- Imprime un mensaje recordatorio, por ejemplo, "No olvides el cumpleaños de [nombre] el [fecha]".

4. Decisión de Ropa:

- Estás eligiendo qué ropa ponerte y quieres saber si hace frío o calor.
- Crea un programa que solicite al usuario la temperatura actual en grados Celsius.
- Indica si la temperatura es baja (frío), media (templado) o alta (calor).

5. Planificador de Compras:

- Estás planeando una fiesta y necesitas hacer una lista de compras.
- Escribe un programa que permita al usuario agregar artículos a una lista de compras.

- Muestra la lista completa para que el usuario pueda revisarla antes de ir al supermercado.

6. Repetidor de Ejercicios:

- Estás siguiendo una rutina de ejercicios y necesitas repetir ciertos movimientos.
- Desarrolla un programa que pida al usuario el nombre de un ejercicio y cuántas veces debe repetirlo.
- Imprime el nombre del ejercicio el número de veces indicado para ayudar a seguir la rutina.