

## CASO PRÁCTICO 1

### LÓGICA DE PROGRAMACIÓN

#### Contexto

Tu jefe te ha comunicado que en pocas semanas comenzarás a codificar en Java, por lo que decides seguir practicando para dominar la lógica de programación combinando estructuras de control en algoritmos realizados con pseudocódigo y con la aplicación Pselnt.

Para ello, deberás realizar un algoritmo que pida un número entero entre 0 y 99.999. En el caso de que no esté en este rango se debe informar de que no es un rango correcto y volver a permitir que se introduzca un nuevo número.

Una vez el número introducido se ha validado como número que forma parte del rango indicado, se realizarán las acciones necesarias utilizando aquellas estructuras adecuadas para comprobar de cuántas cifras está compuesto dicho número. Por ejemplo: Si el número es 15, el mensaje mostrado será "El número tiene 2 cifras", si el número es 4500, sería "El número tiene 4 cifras".

Cuando se haya mostrado el número de cifras del número introducido, se le hará al usuario la pregunta "¿Quieres continuar? pulsa s ó cualquier tecla para finalizar". En caso de pulsar una tecla diferente a 's' la ejecución debe finalizar y si es una 's' volver a realizar todo el proceso.

#### Cuestiones a resolver

Resuelve las cuestiones planteadas utilizando la aplicación Pselnt y comentarios en el pseudocódigo para hacerlo más comprensible.

#### Objetivos

Dominar la lógica de programación combinando estructuras de control en algoritmos realizados con pseudocódigo y con la aplicación Pselnt.

---

## PRESENTACIÓN DEL CASO PRÁCTICO 1: LÓGICA DE PROGRAMACIÓN EN PSEINT

### 1. Introducción

Este trabajo consiste en desarrollar un algoritmo en Pselnt que permita mejorar la lógica de programación mediante el uso de estructuras de control. El objetivo principal es que el algoritmo solicite un número entero dentro de un rango específico, determine la cantidad de cifras que contiene y pregunte al usuario si desea continuar. Este proceso permitirá aplicar los conceptos de validación de datos, estructuras de repetición y selección.

### 2. Objetivos

- Solicitar al usuario un número entero entre 0 y 99.999, y verificar si está en el rango.
- Calcular y mostrar el número de cifras del valor introducido.

- Preguntar al usuario si desea realizar otra consulta. En caso afirmativo, el proceso se repetirá; de lo contrario, finalizará la ejecución.

### 3. Presentación del algoritmo 1

```

Algoritmo ContarCifras
    // Declaración de variables
    Definir numero Como Entero // Variable para almacenar el número
    ingresado por el usuario
    Definir respuesta Como Caracter // Variable para almacenar la decisión
    de continuar o finalizar del usuario
    respuesta <- "s" // Inicializamos "respuesta" con "s" para entrar en el
    ciclo la primera vez

    // Ciclo principal que se repite mientras el usuario desee continuar
    Mientras respuesta = "s" Hacer
        // Pedir al usuario que ingrese un número dentro del rango
        // permitido
        Escribir "Introduce un número entre 0 y 99,999:"
        Ler numero

        // Validar que el número esté dentro del rango 0-99,999
        Mientras numero < 0 O numero > 99999 Hacer
            Escribir "Número fuera del rango permitido. Inténtalo
            nuevamente."
            Ler numero // Leer un nuevo valor de número hasta que esté en
            el rango
        FinMientras

        // Determinación de la cantidad de cifras según el valor de
        // "numero"
        Si numero >= 0 Y numero < 10 Entonces
            Escribir "El número tiene 1 cifra." // Caso para números de 1
            cifra (0 a 9)
        Sino
            Si numero >= 10 Y numero < 100 Entonces
                Escribir "El número tiene 2 cifras." // Caso para números
                de 2 cifras (10 a 99)
            Sino
                Si numero >= 100 Y numero < 1000 Entonces
                    Escribir "El número tiene 3 cifras." // Caso para
                    números de 3 cifras (100 a 999)
                Sino
                    Si numero >= 1000 Y numero < 10000 Entonces
                        Escribir "El número tiene 4 cifras." // Caso para
                        números de 4 cifras (1000 a 9999)
                    Sino
                        Si numero >= 10000 Y numero <= 99999 Entonces
                            Escribir "El número tiene 5 cifras." // Caso
                            para números de 5 cifras (10000 a 99999)
                        FinSi
                    FinSi
                FinSi

```

```

FinSi
FinSi

    // Solicitar al usuario si desea realizar otra operación o
    finalizar
    Escribir "¿Quieres continuar? Pulta s para sí o cualquier otra
    tecla para finalizar."
    Leer respuesta // Leer la respuesta; si es 's', el ciclo se
    repetirá, de lo contrario finalizará
    FinMientras

    // Mensaje final al cerrar el programa
    Escribir "Programa finalizado."
FinAlgoritmo

```

### 3.1 Explicación del Pseudocódigo

- **Declaración de variables:**
  - Se definen dos variables: **numero** (para almacenar el número ingresado) y **respuesta** (para decidir si el usuario quiere continuar o no).
- **Inicialización:**
  - La variable **respuesta** se inicializa con "s", lo que permite entrar en el ciclo principal.
- **Ciclo principal:**
  - El algoritmo entra en un bucle que se repetirá mientras **respuesta** sea igual a "s".
  - Dentro del ciclo, se solicita al usuario que ingrese un número entre 0 y 99,999.
- **Validación del número:**
  - Se verifica si el número está dentro del rango permitido (0 a 99,999). Si el número no está en ese rango, se muestra un mensaje de error y se vuelve a solicitar un nuevo número hasta que el usuario ingrese un valor válido.
- **Determinación de cifras:**
  - Dependiendo del valor de **numero**, se usa una serie de condicionales (**Si...Sino**) para determinar cuántas cifras tiene:
    - i. 1 cifra (0 a 9)
    - ii. 2 cifras (10 a 99)
    - iii. 3 cifras (100 a 999)
    - iv. 4 cifras (1000 a 9999)
    - v. 5 cifras (10000 a 99999)
  - Para cada caso, se imprime el número de cifras correspondiente.
- **Decisión para continuar o finalizar:**
  - Al final de cada iteración del ciclo, se pregunta al usuario si desea continuar. Si responde "s", el ciclo se repite; de lo contrario, se sale del ciclo.

- **Mensaje final:**
  - Una vez que el usuario decide no continuar, se imprime un mensaje indicando que el programa ha finalizado.

### 3.2 Ejemplo de Ejecución

Cuando se ejecuta el algoritmo, podría mostrarse de la siguiente manera:

```

Introduce un número entre 0 y 99,999:
4567
El programa interpreta que es un número en rango.
El número tiene 4 cifras.
¿Quieres continuar? Pulsa s para sí o cualquier otra tecla para
finalizar.
s
Introduce un número entre 0 y 99,999:
150
El programa interpreta que es un número en rango.
El número tiene 3 cifras.
¿Quieres continuar? Pulsa s para sí o cualquier otra tecla para
finalizar.
n
Programa finalizado al pulsar una tecla diferente a la 's'.

```

### 4. Presentación del algoritmo 2

```

// Algoritmo para contar cifras de un número en el rango especificado.
Algoritmo ContarCifras
    // Declaración de variables
    Definir numero Como Entero
    Definir continuar Como Caracter

    // Inicio del bucle principal
    Repetir
        Escribir "Introduce un número entre 0 y 99,999: "
        Leer numero

        // Validación del rango del número
        Mientras numero < 0 O numero > 99999 Hacer
            Escribir "El número no está en el rango correcto. Inténtalo
de nuevo."
            Escribir "Introduce un número entre 0 y 99,999: "
            Leer numero
        FinMientras

        // Determinar el número de cifras
        Si numero < 10 Entonces
            Escribir "El número tiene 1 cifra."
        Sino
            Si numero < 100 Entonces

```

```

        Escribir "El número tiene 2 cifras."
    Sino
        Si numero < 1000 Entonces
            Escribir "El número tiene 3 cifras."
        Sino
            Si numero < 10000 Entonces
                Escribir "El número tiene 4 cifras."
            Sino
                Escribir "El número tiene 5 cifras."
            FinSi
        FinSi
    FinSi

    // Preguntar si desea continuar
    Escribir "¿Quieres continuar? Pulsa 's' para continuar o
cualquier otra tecla para finalizar."
    Leer continuar
    Hasta Que continuar <> "s"
FinAlgoritmo

```

#### 4.1 Explicación del Pseudocódigo:

1. **Declaración de variables:**
  - Se declaran dos variables: `numero` (para almacenar el número ingresado por el usuario) y `continuar` (para decidir si el usuario desea repetir el proceso).
2. **Inicio del bucle principal:**
  - Se utiliza un bucle `Repetir...Hasta Que`, lo que implica que el bloque de código dentro de este bucle se ejecutará al menos una vez y continuará repitiéndose hasta que la condición especificada se cumpla.
3. **Ingreso del número:**
  - Se solicita al usuario que introduzca un número entre 0 y 99,999. Este número se almacena en la variable `numero`.
4. **Validación del número:**
  - Se emplea un bucle `Mientras` para comprobar si el número ingresado está dentro del rango permitido. Si el número es menor que 0 o mayor que 99,999, se muestra un mensaje de error y se vuelve a solicitar un nuevo número hasta que se introduzca un valor válido.
5. **Determinación del número de cifras:**
  - Se utiliza una serie de condicionales (`Si...Sino`) para determinar cuántas cifras tiene el número ingresado:
    - i. 1 cifra (0 a 9)
    - ii. 2 cifras (10 a 99)
    - iii. 3 cifras (100 a 999)
    - iv. 4 cifras (1000 a 9999)
    - v. 5 cifras (10000 a 99999)

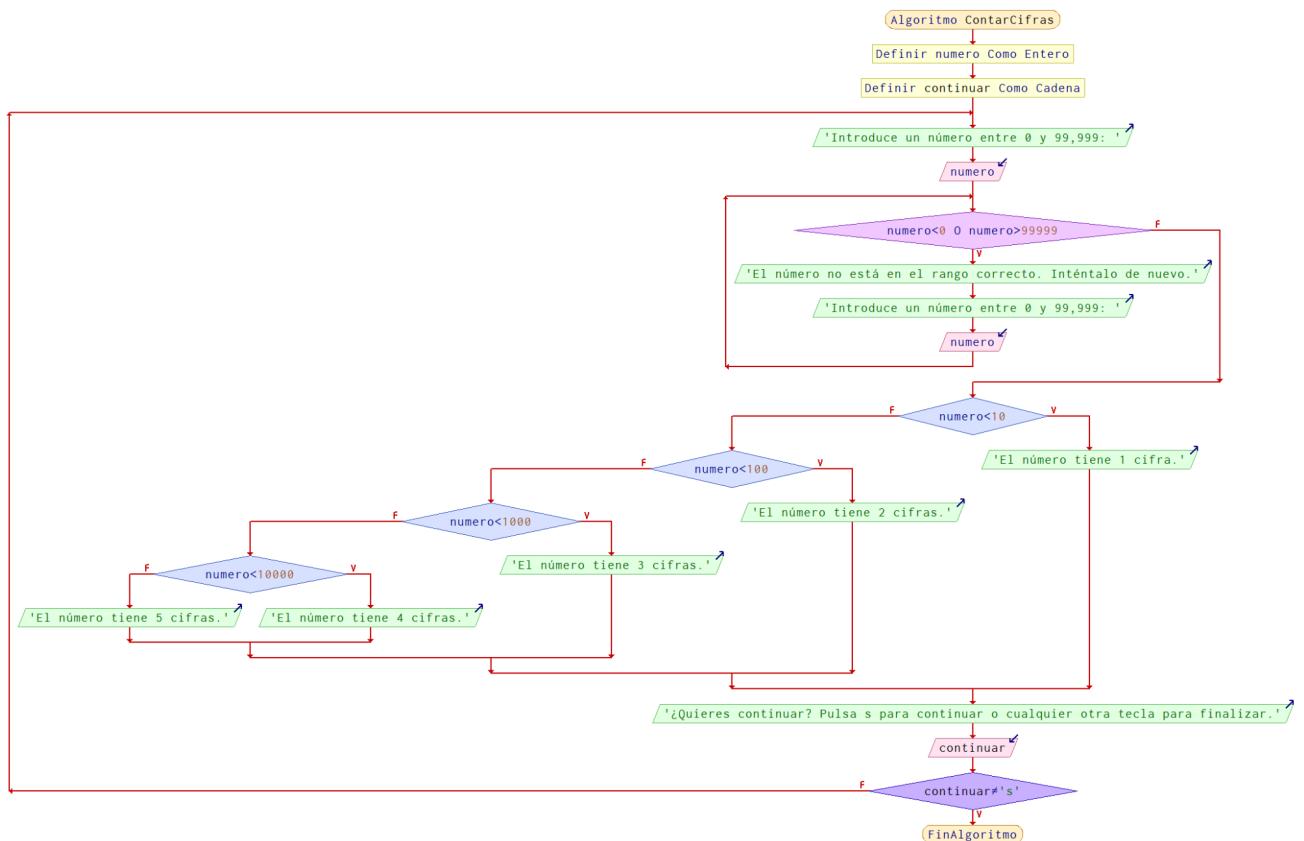
6. **Consulta para continuar o finalizar:**
  - Al final de cada iteración del bucle, se pregunta al usuario si desea continuar. Si el usuario responde con 's', el bucle se repetirá; si responde con cualquier otra tecla, el bucle se detendrá.
7. **Condición de finalización:**
  - La condición del bucle **Hasta Que** se verifica al final de cada iteración, lo que permite salir del bucle cuando el usuario decide no continuar.

## 4.2 Ejemplo de Ejecución

Cuando se ejecuta el algoritmo, podría mostrarse de la siguiente manera:

```
Introduce un número entre 0 y 99,999:  
125  
El programa interpreta que es un número en rango.  
El número tiene 3 cifras.  
¿Quieres continuar? Pulsa 's' para continuar o cualquier otra tecla para  
finalizar.  
s  
Introduce un número entre 0 y 99,999:  
45678  
El programa interpreta que es un número en rango.  
El número tiene 5 cifras.  
¿Quieres continuar? Pulsa 's' para continuar o cualquier otra tecla para  
finalizar.  
x  
Programa finalizado al pulsar una tecla diferente a la 's'.
```

## 4.3 Diagrama de Flujo



## 5. Conclusión

Este ejercicio ha sido útil para practicar la lógica de programación, permitiendo combinar estructuras de control para resolver un problema común de validación y procesamiento de datos en Pselnt. La práctica refuerza el uso adecuado de pseudocódigo en el diseño de algoritmos y facilita la transición a la programación en Java, tal como se requiere en el entorno laboral.

## Recursos Utilizados

- Herramienta Pselnt para el desarrollo de pseudocódigo.
- Contenidos teóricos de la unidad de programación.
- Udemy
- Youtube