



EMILIO GÓMEZ BRESCHI
FÁTIMA LISETTE HERNÁNDEZ BARAJAS
RUTH REGINA MENDOZA OCHOA

Paradigma procedimental



EQUIPO 5

¿Que es?

Es un paradigma de Programación y su origen parte como una evolución de la programación estructurada (que se basa principalmente en dividir un problema en partes pequeñas).

Como aborda problemas

Este paradigma descompone el problema en subproblemas y organiza el código en procedimientos también conocidos como subrutinas las cuales contienen una secuencia de instrucciones con orden lógico para realizar tareas específicas.

Se centra en mejorar la claridad y tiempo de desarrollo de los programas.

Taza de cafe

Ejemplo

1. Poner agua a calentar.
2. Colocar café en una taza.
3. Cuando el agua hierve, verterla en la taza.
4. Revolver con una cuchara.
5. Agregar azúcar o leche (opcional).
6. Servir y beber.



Cracteristicas

1

MODULARIDAD

Los programas se dividen en módulos o procedimientos, lo que facilita la comprensión y el mantenimiento del código.

2

REUTILIZACIÓN DE CÓDIGO

Los procedimientos pueden ser reutilizados en diferentes partes del programa o incluso en otros programas, lo que reduce la duplicación de código.

3

SECUENCIALIDAD

La ejecución del programa sigue una secuencia de pasos, lo que facilita el seguimiento del flujo de control.

4

ABSTRACCIÓN

Los detalles de implementación de un procedimiento se ocultan detrás de su interfaz, lo que permite a los programadores utilizar procedimientos sin necesidad de comprender su funcionamiento interno.

5

CONTROL DE FLUJO

El paradigma procedimental proporciona estructuras de control que permiten a los programadores dirigir el flujo de ejecución del programa de manera efectiva.

Ventajas

**MUCHOS LENGUAJES
DE PROGRAMACIÓN
DE PROPÓSITO
GENERAL LO ADMITEN.**

**SIMPLIFICA EL
CÓDIGO, LO HACE
MÁS COMPRENSIBLE.**

**IMPLEMENTAR
ALGORITMOS ES FÁCIL.**

**EL PROGRAMA FLUYE
EN DIRECCIÓN FINAL
LO QUE LO HACE FÁCIL
DE SEGUIR.**

**AUMENTA LA
REUTILIZACIÓN DE
CÓDIGO**

Desventajas

**DIFICULTAD PARA
MANEJAR LA
COMPLEJIDAD**

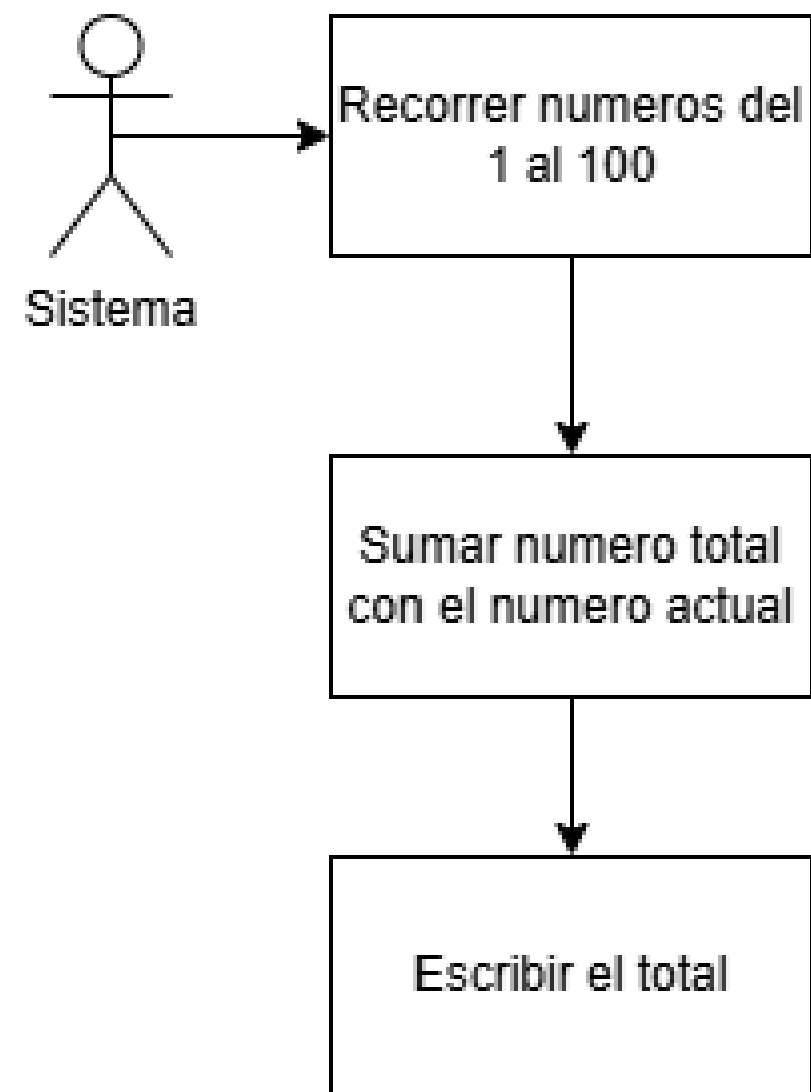
**DEPENDENCIA DE
ESTADO GLOBAL**

**.FALTA DE
ENCAPSULAMIENTO**

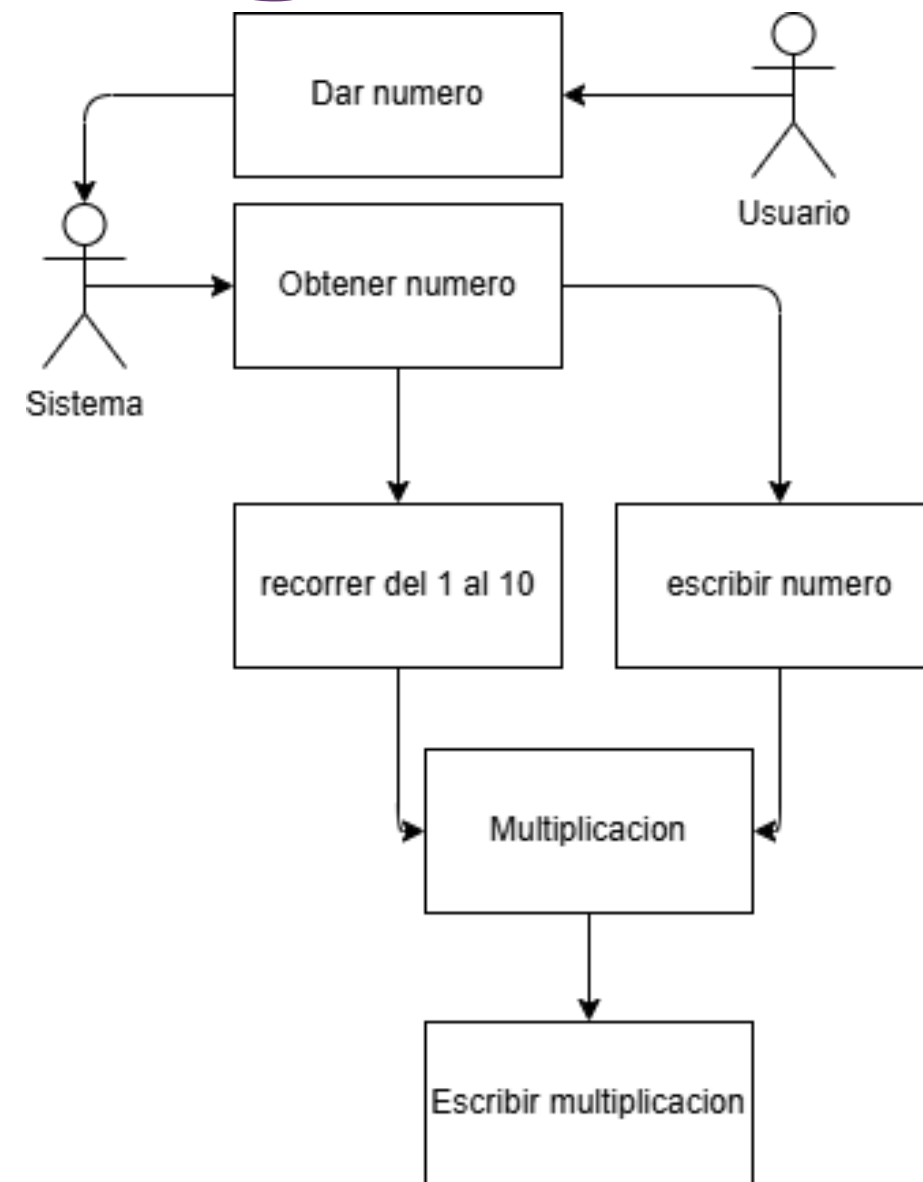
**DIFICULTAD PARA LA
PRUEBA Y
VALIDACIÓN**

RIGIDEZ

Programa 1



Programa 2



Como usar abstraccion

En los dos problemas aplicamos la abstracción al crear los procedimientos `multiplicar()` y `suma()` dentro de los cuales se desarrolla la lógica utilizando ciclos `for` en ambos casos para obtener los resultados, cuando se necesite hacer cualquiera de estas dos operaciones no será necesario volver a escribir toda la lógica si no que bastara con llamar al procedimiento correspondiente.

Conclusion

El paradigma procedimental ha sido fundamental en la evolución de la programación, proporcionando un enfoque claro y estructurado para el desarrollo de software. A pesar de sus desventajas, como la dificultad para manejar la complejidad y la falta de encapsulamiento, sigue siendo ampliamente utilizado en la práctica. La comprensión de este paradigma es esencial para los programadores, ya que sienta las bases para el aprendizaje de enfoques más avanzados, como la programación orientada a objetos.