

# Paradigma de Programación Estructurada

Por:

- Humberto Alejandro Navarro Andujo
- Martin Francisco Ortiz Guerrero
- Raul Alejandro Diaz Gutierrez





# Orígenes

Estamos en 1966, En este año Böhm y Jacopin proponen el teorema del programa estructurado, concciones.de co este demuestran que cualquier programa puede ser escrito utilizando solamente tres instrucciones de control..

Dos años más tarde Edsger Dijkstra publica su celebre articulo ***Go To Statement Considered Harmful***. ¿Por qué es importante? Pues porque este científico holandés promovió activamente el uso de lenguajes de programación estructurada, fomentando la verificación formal de programas y la eliminación de la sentencia Goto.





# En qué consiste este paradigma?

Esta forma de programar (o paradigma) se basa en un dicha publicación, desarrollado por Edsger Dijkstra, que establece que todo programa puede escribirse utilizando únicamente las tres estructuras básicas de control:

- **Secuencia:** el bloque secuencial de instrucciones, ejecutadas sucesivamente, una detrás de otra.
- **Selección:** la instrucción condicional con doble alternativa, de la forma “*if* condición *then* instrucción-1 *else* instrucción 2”.
- **Iteración:** el bucle condicional “while condición do instrucción”, que ejecuta la instrucción repetidamente mientras la condición se cumpla.



# En qué consiste este paradigma?

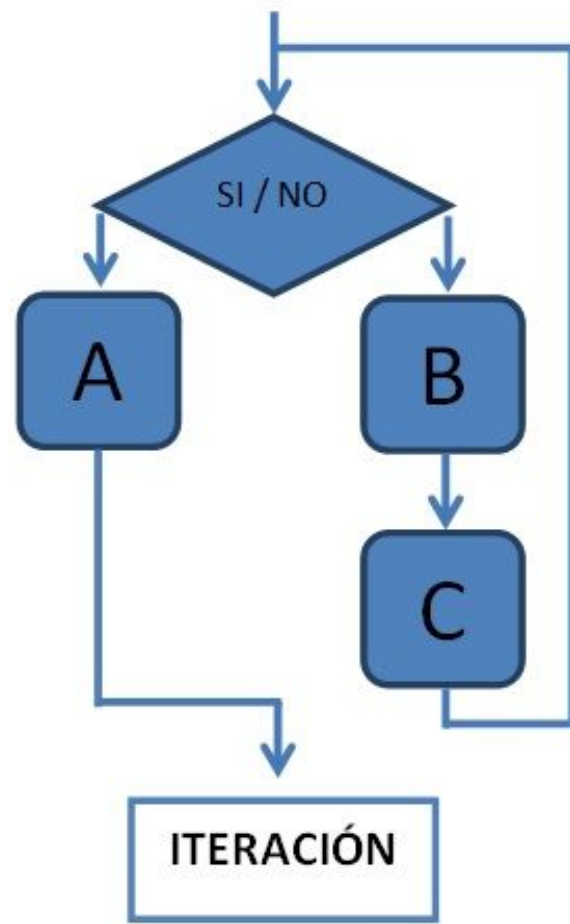
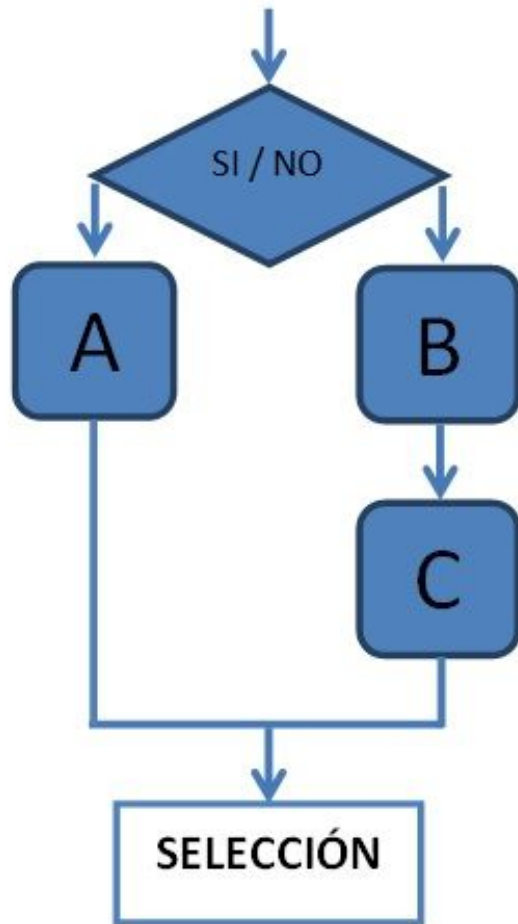
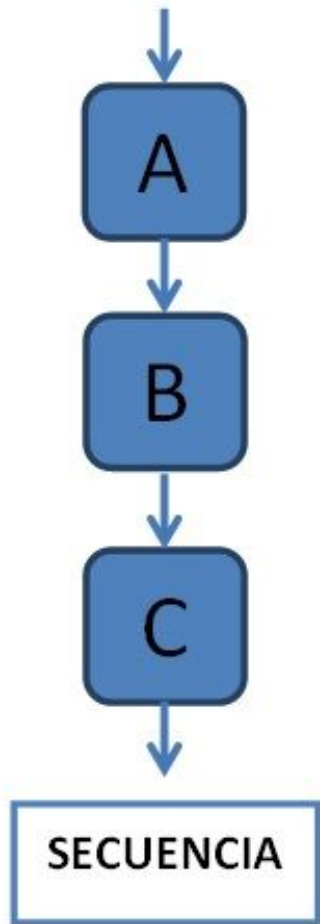
La estructura de secuencia es la que se da naturalmente en el lenguaje, ya que por defecto las sentencias son ejecutadas en el orden en que aparecen escritas en el programa.

Para las estructuras condicionales o de selección, la mayoría de los lenguajes de programación disponen de una sentencia if, que puede combinarse con sentencias else y/o else if, además de switch.

Para los bucles o iteraciones existen las estructuras while y for.

Cabe destacar que en un enfoque más modernizado, se agrega un punto muy importante: segmentación. La segmentación, o también denominado en algunos lenguajes como funciones o subrutinas, nos trae grandes ventajas como:

- Facilita el trabajo simultáneo en paralelo de distintos grupos de programadores.
- Posibilita en mayor grado la reutilización de código en futuras aplicaciones.





# Ventajas de este paradigma

- **Los programas son más fáciles de entender**, pueden ser leídos de forma secuencial y no hay necesidad de tener que rastrear saltos de líneas (GOTO) dentro de los bloques de código para intentar entender la lógica interna.
- **La estructura de los programas es clara**, puesto que las sentencias están más ligadas o relacionadas entre sí.
- Se optimiza el esfuerzo en las fases de pruebas y depuración. **El seguimiento de los fallos o errores del programa** (debugging), y con él **su detección y corrección, se facilita enormemente**.
- **Se reducen los costos de mantenimiento**. Durante la fase de mantenimiento, modificar o extender los programas resulta más fácil.
- **Los programas son más sencillos y más rápidos de confeccionar**.
- **Se incrementa el rendimiento de los programadores**.



# Ejemplos de programación estructurada

Casi cualquier fragmento de código de muestra en cualquier lenguaje de programación moderno será un ejemplo de programación estructurada.

Esta programación utiliza bucles, condicionales y subrutinas para poder controlar el flujo y modularidad para así facilitar la lectura de poder reutilizar el código.



## Secuencia

Las líneas o bloques de código se escriben y ejecutan en un orden secuencial. Por ejemplo, se tiene:

```
x= 6
```

```
y= 12
```

```
z= x + y
```

```
WriteLine(z)
```





## Repetición

Se repite un bloque de código mientras se cumple una condición. No hay límite para la cantidad de veces que se puede ejecutar el bloque. Ejemplo:

```
x= 2
```

```
Mientras x<100
```

```
WriteLine(x)
```

```
x= x * x
```

```
Fin
```



## Selección

Se ejecuta un bloque de código si una condición es verdadera. El bloque de código se ejecuta como máximo una vez. Ejemplo:

```
x= ReadLine()
```

```
Si x Mod 2= 0
```

```
WriteLine("El número es par")
```

```
Fin de Si
```



# Referencias

- Anónimo. (2021). Programación estructurada. Agosto 25, 2021, de EDIX Sitio web: <https://www.edix.com/es/instituto/programacion-estructurada/>
- Edgar Dijkstra. (March, 1968). Go To Statement Considered Harmful. Marzo, 1968 Sitio web: <https://homepages.cwi.nl/~storm/teaching/reader/Dijkstra68.pdf>
- Ivan Medina. (Feb 19, 2017). Programación Estructurada. Medium. Sitio web: <https://medium.com/laboratoria-how-to/programaci%C3%B3n-estructurada-7fe400bae43d>
- Helmut Sy Corvo. (8 de marzo de 2020). **Programación estructurada: características, ejemplos, ventajas, aplicaciones.** Lifeder. Recuperado de <https://www.lifeder.com/programacion-estructurada/>.