

Algoritmia y Programación

Estructuras Repetitivas

Estructuras Repetitivas

- Es muy común encontrar en la práctica algoritmos cuyas operaciones se deben ejecutar un número repetido de veces.
- El conjunto de instrucciones que se ejecuta repetidamente se llama ciclo o iteración.
- Todo ciclo debe terminar de ejecutarse luego de un número finito de veces, por lo que es necesario en cada iteración del mismo, evaluar las condiciones necesarias para decidir si se debe seguir ejecutando o si debe detenerse.
- En todo ciclo, siempre debe existir una condición de parada o fin de ciclo.

Estructuras Repetitivas

- Las estructuras lógicas repetitivas se utilizan cuando necesitamos realizar un algoritmo en el cual haya que repetir un proceso un número N de veces, las estructuras repetitivas se pueden mezclar con las estructuras selectivas si el problema lo necesita, también se hará uso de expresiones lógicas y operadores relacionales.

Contadores

- La construcción de un contador es una de las técnicas más comunes en algoritmos repetitivos.
- Un contador es una variable que se incrementará en una unidad cada vez que se ejecute el proceso.
- El contador se utiliza para llevar la cuenta de determinadas acciones que se pueden solicitar durante la resolución de un problema. En las instrucciones de preparación se realiza la inicialización del contador o contadores. La inicialización consiste en poner el valor inicial de la variable que representa al contador. Generalmente se inicializa con el valor 0.

Contadores

- Un contador es una variable utilizada para contar ocurrencias, es decir, llevar registro de cuántas veces ocurre algo.

Ejemplos:

- Aplicaciones de Redes Sociales: Los contadores se utilizan para seguir la cantidad de "me gusta", comentarios o seguidores de un usuario. Cada vez que un usuario recibe un nuevo "me gusta", el contador correspondiente se incrementa en uno.
- Sistemas de Comercio Electrónico: En una tienda en línea, un contador puede llevar la cuenta de los artículos disponibles en stock. Cada vez que se realiza una venta, el contador disminuye en función de la cantidad de artículos vendidos.
- Videojuegos: Los contadores se emplean para rastrear la puntuación de los jugadores, vidas restantes, o el número de enemigos derrotados. Estos contadores se actualizan a lo largo del juego según las acciones del jugador.

Acumuladores

- Un acumulador es una variable en la memoria cuya misión es almacenar cantidades variables, a diferencia del contador que almacena incrementos fijos.
- Se utiliza para efectuar sumas, multiplicaciones u operaciones sucesivas. La principal diferencia con el contador es que el incremento o decremento de cada suma es variable en lugar de constante como en el caso del contador.

Acumuladores

Ejemplos de Acumuladores

- **Análisis Financiero:** Los acumuladores son cruciales para calcular totales, como el saldo de una cuenta bancaria. Cada transacción (ingreso o gasto) modifica el saldo total acumulado de la cuenta.
- **Sistemas de Facturación:** En un sistema de facturación, un acumulador puede sumar el total de los costos de los productos o servicios adquiridos, aplicando los impuestos y descuentos correspondientes para obtener el monto final a pagar.
- **Procesamiento de Datos y Estadísticas:** Los acumuladores son utilizados en la ciencia de datos para sumar grandes volúmenes de información, como el total de kilómetros recorridos por una flota de vehículos, o para calcular promedios, como la temperatura media en un período específico.

Ciclos - For

En algunos programas o algoritmos desde la misma concepción podemos identificar que necesitamos repetir una estructura o una tarea un numero conocido de veces. Por ejemplo, en ejercicios anteriores se ha necesitado ingresar una cantidad de valores fija para calcular su valor promedio, ahora este tipo de estructuras (repetitivas) nos permiten optimizar nuestros algoritmos de manera directa sin necesidad de repetir instrucciones la cantidad de veces que un elemento se necesite múltiples ocasiones. Este tipo de estructuras donde se conoce exactamente la cantidad de elementos a repetir tiene una denominación y se le conoce como “For Loops” o “Ciclos Para”.

Ciclos - While

En otros casos existen programas en donde no conocemos la cantidad de elementos o datos a repetir en una estructura, el número de repeticiones dependerá de las decisiones o valores que pueda o puedan tomar algunas variables o de controles internos de este. En otros casos es solo esperar a que una condición o valor se cumpla o se encuentre. Este tipo de estructuras donde no se conoce la cantidad de elementos a repetir tiene una denominación y se le conoce como “While Loops” o “Ciclos Mientras”.

A continuación, veremos en detalle su estructura y beneficios.

FOR

Para, también llamado FOR, es la estructura algorítmica adecuada para utilizar en un ciclo que se ejecutará un número definido de veces.

El número de veces del ciclo no depende de las proposiciones dentro de el mismo. El número de veces se obtiene del planteamiento del problema o de una lectura que indica que el número de iteraciones se debe realizar para N ocurrencias.

FOR

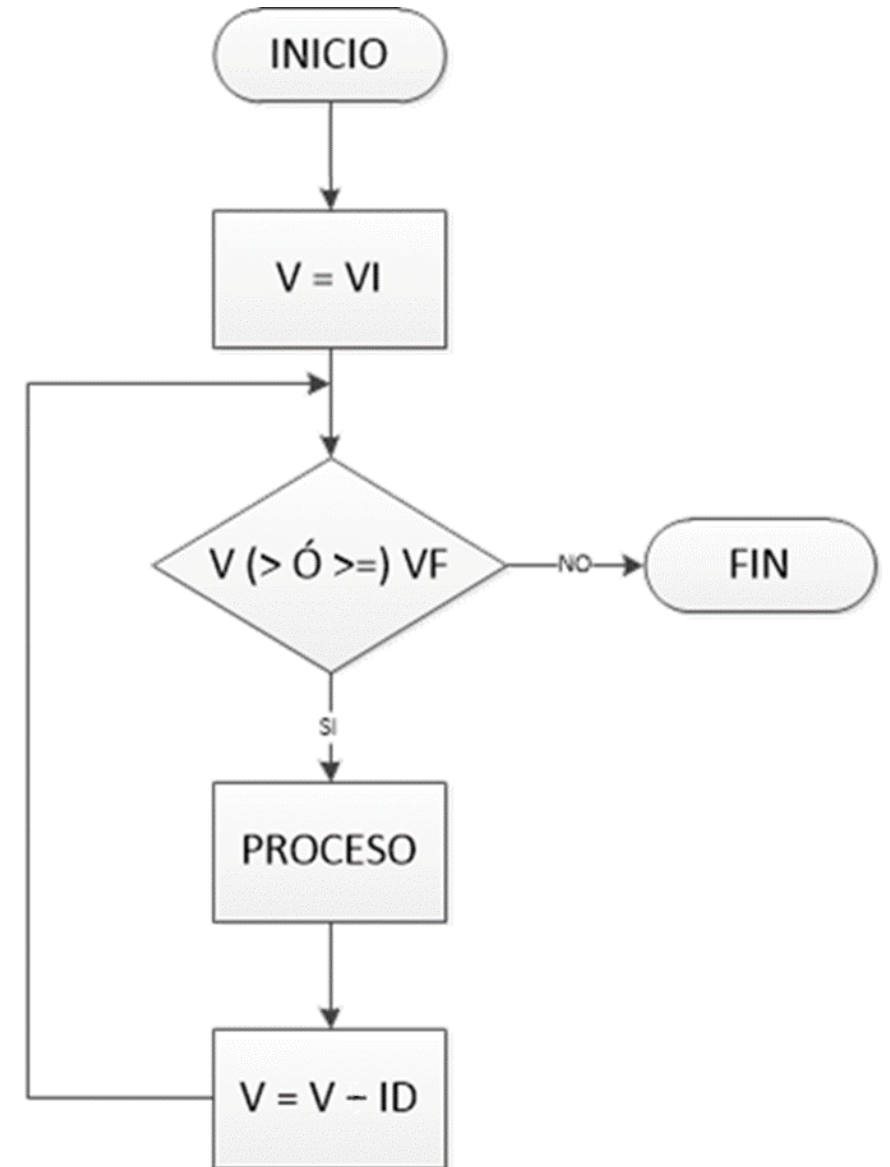
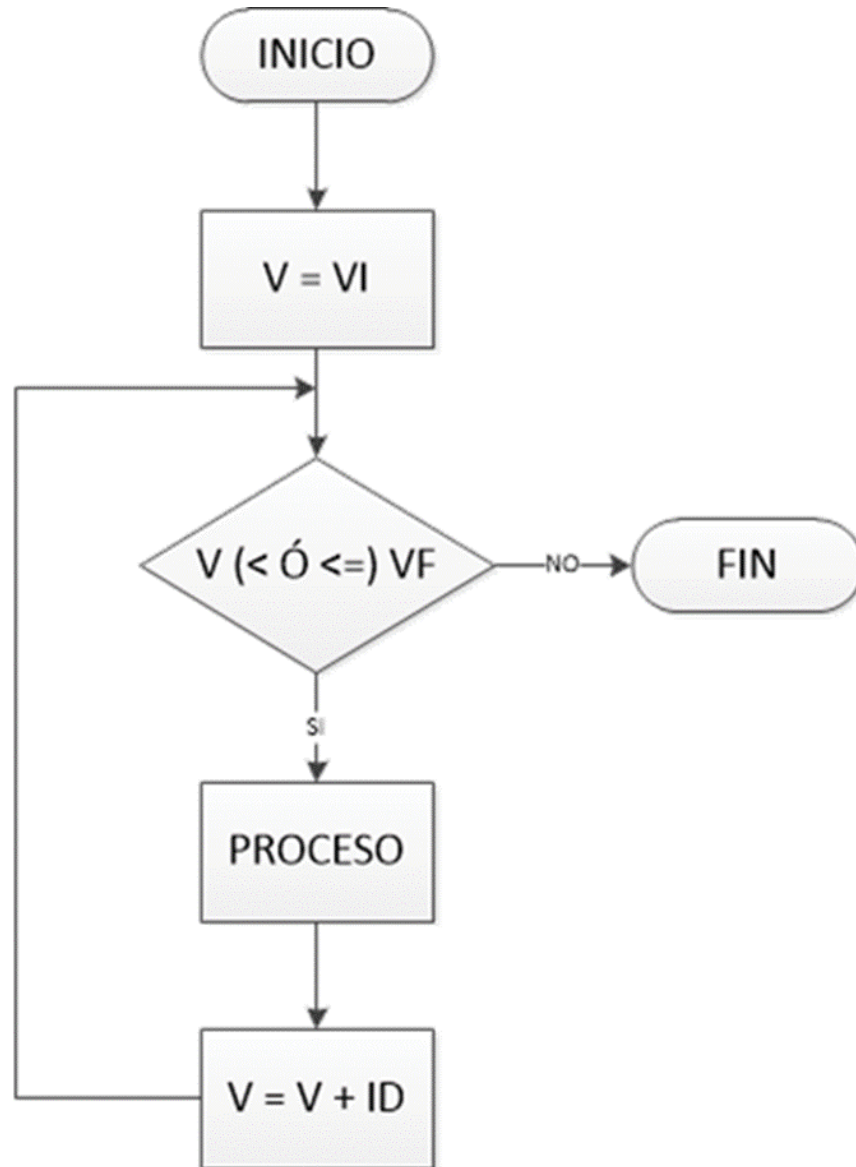
Para los ciclos Para (FOR) se definen las siguientes variables:

- V: es la variable de control del ciclo.
- VI: es el valor inicial.
- VF: es el valor final.
- ID: es el incremento o decremento, según sea la estructura ascendente o descendente.

V, contador del ciclo normalmente es representado por las letras “i, j, k”

- El ciclo se ejecuta mientras V es menor, menor igual, mayor o mayor igual al valor de VF, el valor de V se incrementa o disminuye en cada iteración. Cuando V supera el valor de VF el ciclo se detiene.

FOR



Usando i como variable en ciclos

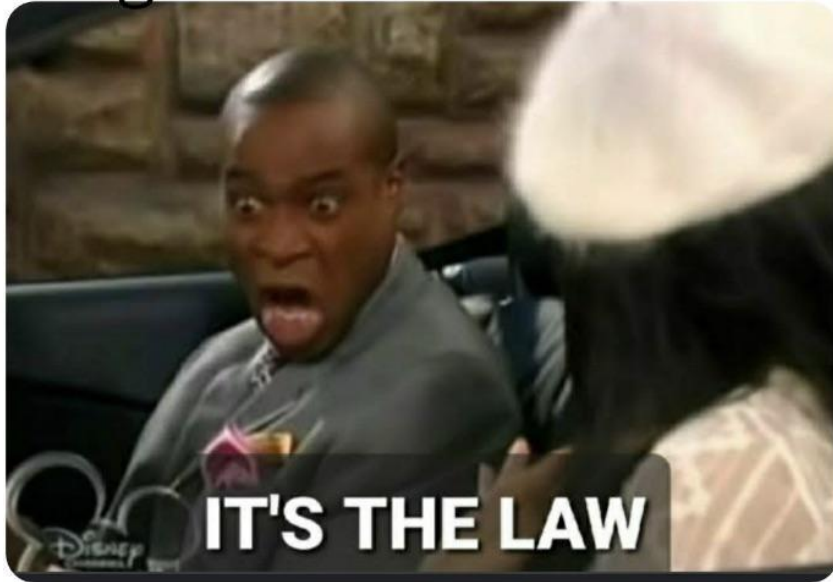
La tradición de usar i como variable preferida en los ciclos, especialmente en los ciclos FOR, proviene de las primeras décadas de la programación y tiene sus raíces en la matemática y en el lenguaje de programación Fortran, uno de los primeros lenguajes de programación de alto nivel, desarrollado en los años 50.

- De las matemáticas:
 - En matemáticas, la letra i se ha utilizado tradicionalmente para representar el primer índice en sumas y otras operaciones repetitivas. Esta práctica se remonta al siglo XIX y ha influido en la cultura de la programación.
- De Fortran:
 - Fortran, siendo un lenguaje ampliamente usado en la comunidad científica y de ingeniería, adoptó esta convención. En Fortran, las variables que comienzan con las letras I, J, K, L, M, o N son automáticamente de tipo entero (integer), lo que las hacía ideales para ser usadas como contadores o índices en los bucles. Esto se debió en parte a la limitación del lenguaje y a la facilidad de escritura y comprensión del código.
- Otros Lenguajes:
 - Con el paso del tiempo, esta convención se ha mantenido y ha sido adoptada por otros lenguajes de programación, como C, C++, Java, y JavaScript, entre otros. Usar i como variable de iteración se ha convertido en una práctica estándar debido a la familiaridad y la expectativa de que i represente un índice o contador en un bucle.
- Es una regla no escrita.
 - Aunque el uso de i es simplemente una convención y los programadores pueden elegir cualquier nombre de variable, la práctica de usar i , seguido por j y luego k para bucles anidados, sigue siendo muy común por su claridad y brevedad, facilitando la lectura y comprensión del código, especialmente en contextos donde se realizan operaciones iterativas o de índices.

Memes

Others : why do you always
use i,j variabes in loops?

Programmers :



Mis neuronas eligiendo el
nombre de una variable



Aah pero cuando eligen el nombre de
una variable para iterar



Carla Notarobot 🤖👤

@CarlaNotarobot

My team had a debate on what the
best looping variable name is

i won

7:38 AM · 19 Jan 22 · [Twitter for Android](#)

223 Retweets 20 Quote Tweets 2,508 Likes



WHILE

Mientras, también llamado WHILE, es la estructura adecuada para utilizar en un ciclo cuando no sabemos el número de veces que este se ha de repetir.

Dicho número depende de las proposiciones dentro del ciclo.

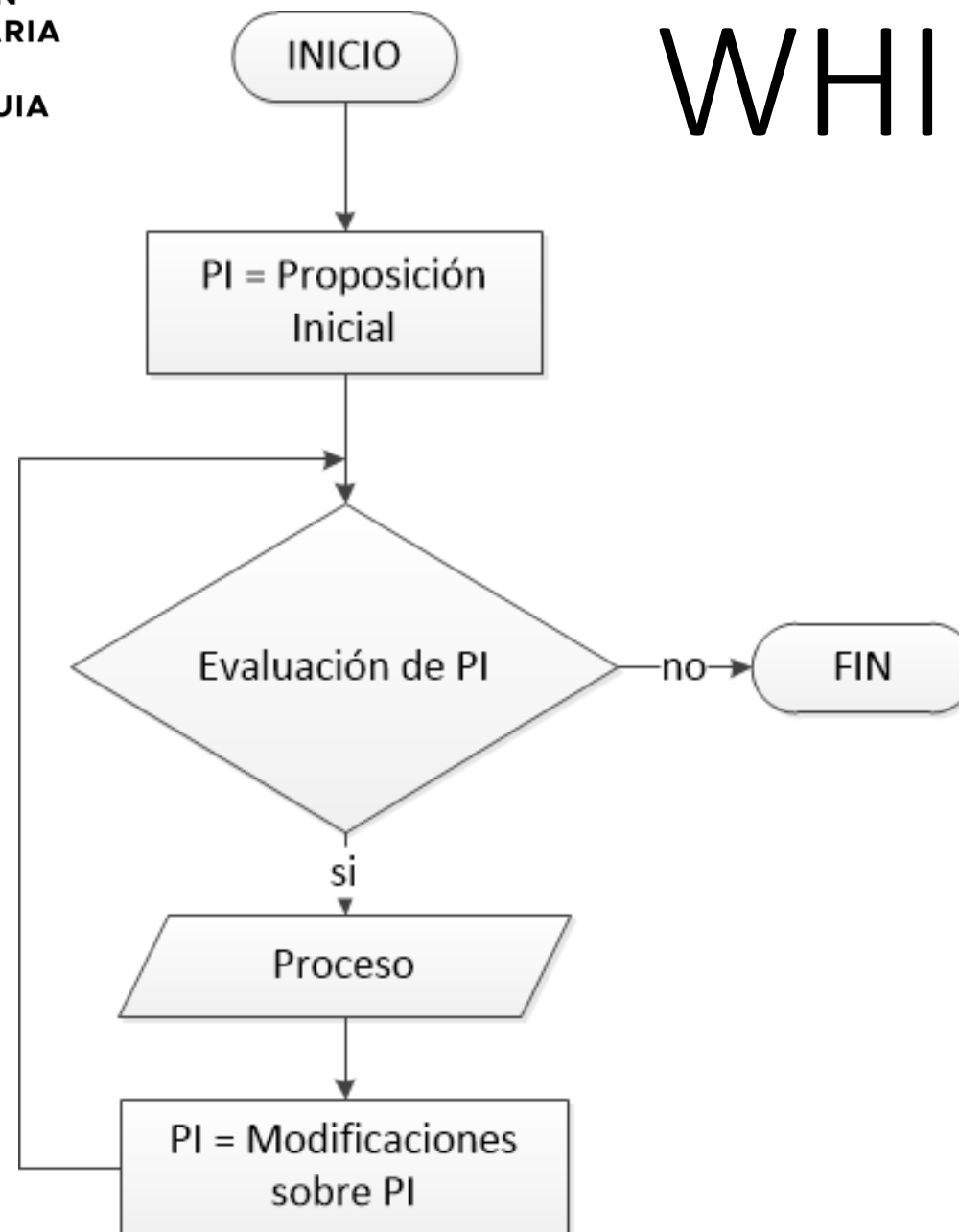
El ciclo mientras (WHILE) se distinguen dos partes:

- Ciclo: Conjunto de instrucciones que se ejecutarán repetidamente.
- Condición de terminación: La evaluación de esta condición permite decidir cuándo finalizará la ejecución del ciclo. La condición se evalúa al inicio del mismo.

Para los ciclos Mientras (WHILE) se definen las siguientes variables:

- PI: La proposición inicial, debe tener un valor verdadero inicialmente, si el valor de PI es falso, entonces el ciclo no se ejecutará.
- Debe existir también un enunciado dentro del ciclo que afecte la condición, para evitar que el ciclo se ejecute indefinidamente.

WHILE



Iterable

```
1 a = ['posicion0', 'posicion1', 'posicion2']  
2 for i in a:  
3     print(i)
```

Iterable:

```
a = ['posicion0', 'posicion1', 'posicion2']
```

Iter()

```
itr = iter(a)
```

next(itr)

'posicion0'

next(itr)

'posicion1'

next(itr)

'posicion2'

next(itr)

StopIteration

Break - Continue

