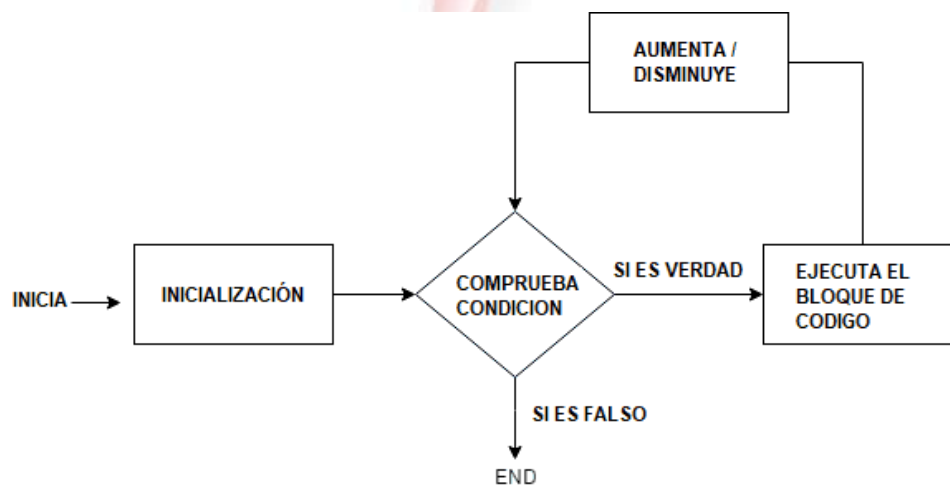


# CICLO FOR

Esquema for:



- Condición de inicialización: Aquí, inicializamos la variable en uso. Marca el inicio de un ciclo for. Se puede usar una variable ya declarada o se puede declarar una variable, solo local para el bucle.



- Condición de prueba: se usa para probar la condición de salida de un bucle. Debe devolver un valor booleano. También es un bucle de control de entrada cuando se verifica la condición antes de la ejecución de las instrucciones de bucle.
- Ejecución de instrucción: una vez que la condición se evalúa como verdadera, se ejecutan las instrucciones en el cuerpo del bucle.
- Incremento/Decremento: se usa para actualizar la variable para la siguiente iteración.
- Terminación de bucle: cuando la condición se vuelve falsa, el bucle termina marcando el final de su ciclo de vida.

### Sintaxis del Ciclo For en Java:

La sintaxis de un ciclo for es simple en Java, en realidad en la mayoría de los lenguajes de alto nivel es incluso muy similar, de hecho, con tan solo tener bien claros los 3 componentes del ciclo for (inicio, final y tamaño de paso) tenemos prácticamente todo hecho

```
for(int i = valor inicial; i <= valor final; i = i + paso)  
{  
    ....  
    ....  
    Bloque de Instrucciones....  
    ....  
    ....  
}
```

Vamos ahora a ver línea por línea el anterior código para comprender todo y quedar claros. Posteriormente veremos un ejemplo con valores reales.

#### Línea 1:

En esta línea está prácticamente todo lo esencial de un ciclo for. La sintaxis es simple, tenemos una variable de control llamada i que es tipo entero (int), cabe notar que la variable se puede llamar como nosotros lo deseemos y puede ser del tipo de queramos también, sin embargo, en la mayoría de los casos se usa la "i" como nombre y el entero como tipo, pero somos libres de modificar esto a



nuestro gusto. Esta variable "i" se le asigna un valor inicial que puede ser cualquier número correspondiente al tipo de dato asignado. Posteriormente lo que haremos será especificar hasta donde irá nuestro ciclo por medio del valor final, ten en cuenta que cada uno de estos componentes es separado por un punto y coma ";", también es importante saber que la condición final puede ser cualquier cosa, mayor, menor, mayor o igual, menor o igual, sin embargo no tiene sentido que la condición sea por ejemplo un igual, pues nuestra variable de control siempre va a cambiar entre valores, menores o mayores que el valor final deseado, si fuera un igual no tendríamos un error de sintaxis, pero nuestro for básicamente no haría nada de nada. Finalmente, el ultimo componente de esta primera línea es el tamaño del paso, este componente se especifica aumentando en la cantidad deseada la variable de control.

#### **Línea 2:**

En la línea 2 tenemos una llave abriendo "{", lo cual como seguramente ya sabrás indica que allí comienza el bloque de instrucciones que se ejecutarán cada vez que el ciclo de un "giro". Esta llave no es del todo obligatoria, sin embargo, si no la ponemos solo se ejecutará dentro de nuestro ciclo la primera línea inmediatamente posterior a la declaración del ciclo, de modo que, si deseamos que se ejecuten varias líneas dentro de nuestro ciclo, debemos usar las llaves

#### **Línea 3 a 7:**

En estas líneas es donde estarán todas las operaciones que queramos llevar a cabo de manera iterativa durante la ejecución del ciclo, este bloque podrá tener la cantidad de líneas necesarias incluso, como veremos más adelante dentro de estas podría haber uno o más ciclos, así que podrías tener todo un programa dentro de un ciclo.

#### **Línea 8:**

En esta última línea hacemos uso de la llave cerrando "}", una vez más como seguramente ya sabrás esta nos indica que allí termina el bloque del ciclo for y se dará por terminada la ejecución de este para continuar ejecutando el resto del algoritmo.