

Pensamiento Computacional

Algoritmos y programación:
sentencias selectivas simples
y dobles sentencias de
control

Clase 14



Ingeniería en ciberseguridad

La excelencia no se improvisa



CLASE 14

14.1 Algoritmos y programación: sentencias selectivas simples y dobles sentencias de control

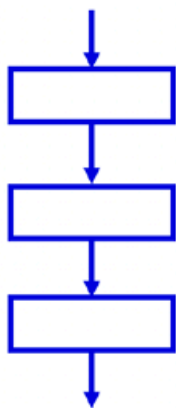
Existen tres tipos:

1. **Secuencial:** En un programa estructurado, las sentencias se escriben en una sucesión que se ejecuta de manera secuencial, es decir, una tras otra en el orden en que aparecen.
2. **Condicional:** Se evalúa una condición, y dependiendo del resultado (verdadero o falso), se ejecuta o no un bloque de instrucciones.
3. **Iterativa:** También evalúa una condición, y mientras esta sea verdadera, se repite un bloque de instrucciones varias veces.

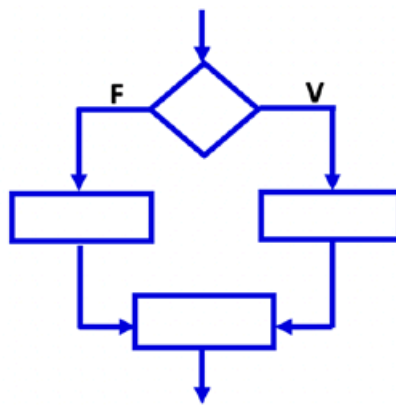
Figura 1.

Tipos de algoritmos para programación

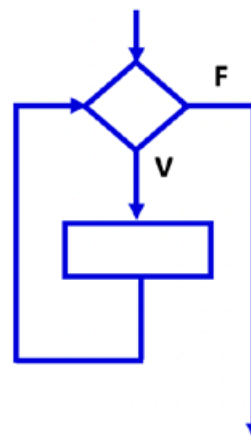
Secuencial



Condicional



Iterativa



Instrucción condicional compuesta

Sintaxis:

Si (condición) Entonces

acciones 1

SiNo

acciones 2

FinSi

Un condicional compuesto sirve para evaluar si una condición es verdadera o falsa.

Si es verdadera, realiza las acciones 1, y, si es falsa, las acciones 2.

Pseudocódigo

Algoritmo Condicionales

Definir num1, num2 **Como** Entero

num1 = 3

num2 = 5

Si num1 > num2 **Entonces**

..... **Escribir** "Num1 es mayor a Num2"

SiNo

..... **Escribir** "Num2 es mayor a Num1"

FinSi

FinAlgoritmo

Salida

 PSeInt - Ejecutando proceso CONDICIONALES

*** Ejecución Iniciada. ***

Num2 es mayor a Num1

*** Ejecución Finalizada. ***

Instrucción condicional simple

Sintaxis:

Si (condición) **Entonces**

acciones

FinSi

Este tipo de condicional carece de la rama “SiNo”, de modo que, si la expresión es falsa, no se realiza ninguna acción y la ejecución continúa por la siguiente instrucción.

Usando el ejemplo anterior se puede observar que no se imprime nada ya que no existe una sentencia en caso de que no se cumpla la condición.

Pseudocódigo

Algoritmo Condicionales

Definir num1, num2 **Como Entero**

num1 = 3

num2 = 5

Si num1 > num2 **Entonces**

Escribir "Num1 es mayor a Num2"

FinSi

FinAlgoritmo

Salida

 PSeInt - Ejecutando proceso CONDICIONALES

*** Ejecución Iniciada. ***

*** Ejecución Finalizada. ***

14.2 Algoritmos y programación: Sentencias selectivas múltiples

Instrucción condicional múltiple

Sintaxis:

Segun (expresión) Hacer

valor1: acciones 1

valor2: acciones 2

...

valorN: acciones N

De Otro Modo:

acciones que no han encajado en algún caso

Fin Segun

Se usa cuando hay más de dos opciones posibles y conocemos los posibles casos.

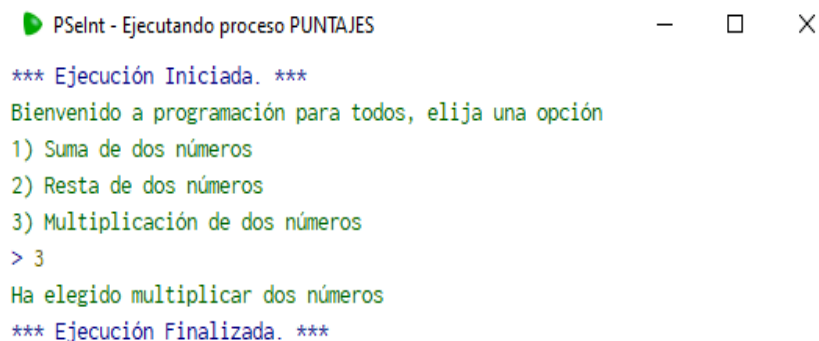
Los valores de 1 a N son los casos, por ejemplo, si tenemos un menú de 3 opciones, tenderemos 3 casos.

Si ningún valor coincide con alguno caso, se realizan las acciones de la parte “De otro modo”.

Pseudocódigo

```
1  Algoritmo Puntajes
2      Definir puntaje Como Entero;
3      Escribir 'Bienvenido a programación para todos, elija una opción';
4      Escribir '1) Suma de dos números';
5      Escribir '2) Resta de dos números';
6      Escribir '3) Multiplicación de dos números';
7      Leer puntaje;
8
9      Segun puntaje Hacer
10         1:
11             Escribir 'Ha elegido suma de dos números';
12         2:
13             Escribir 'Ha elegido resta de dos números';
14         3:
15             Escribir 'Ha elegido multiplicar dos números';
16         De Otro Modo:
17             Escribir 'La opción ingresada no existe';
18     Fin Segun
19 FinAlgoritmo
```

Salida



```
PSeInt - Ejecutando proceso PUNTAJES
*** Ejecución Iniciada. ***
Bienvenido a programación para todos, elija una opción
1) Suma de dos números
2) Resta de dos números
3) Multiplicación de dos números
> 3
Ha elegido multiplicar dos números
*** Ejecución Finalizada. ***
```

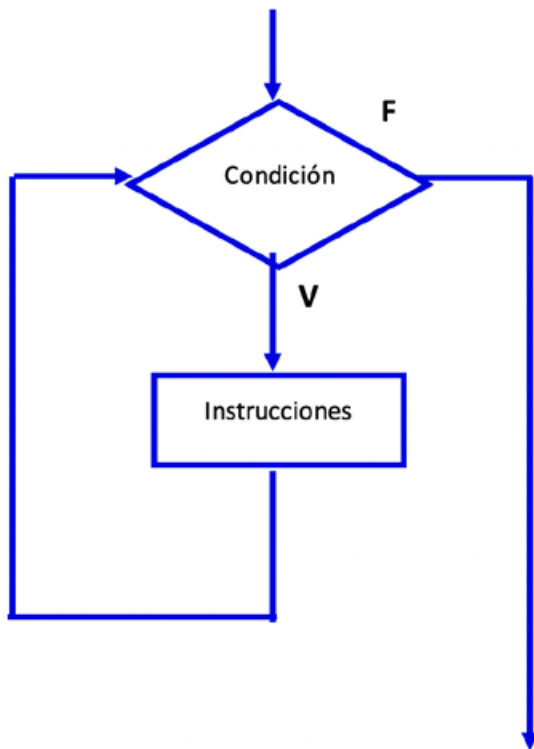
14.3 Algoritmos y programación: sentencias iterativas

Bucles

- Otro nombre para las sentencias **iterativas**.
- Un bucle es una instrucción que permite **repetir un segmento de código** un número determinado de veces o hasta que se cumpla una condición.
- **Cada vuelta** que da un bucle se conoce como iteración.
- Los bucles necesitan un límite o una condición que detenga su ejecución; de lo contrario, si no existe este límite, se dice que el bucle es infinito y puede saturar la memoria del computador.

Figura 2

Estructura de condicional simple



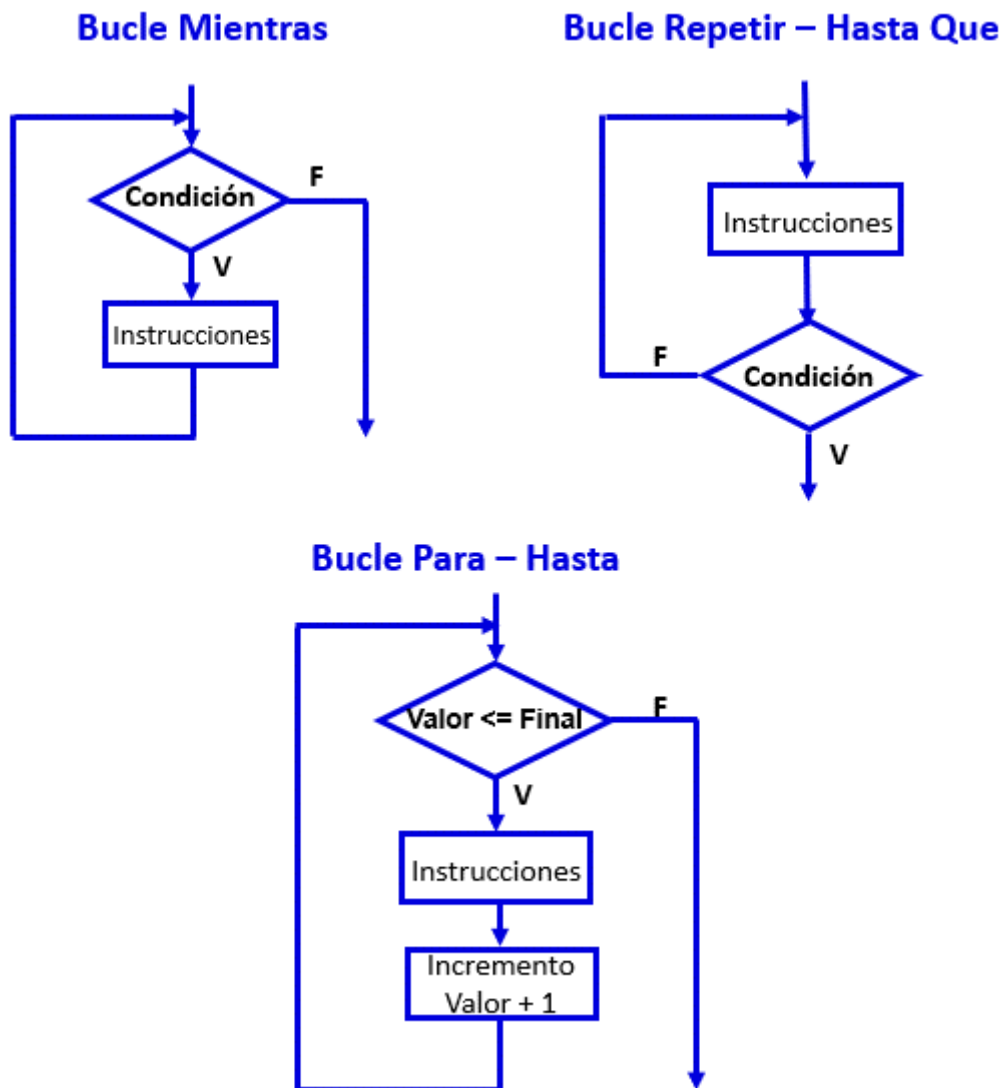
Tipos de Bucles

Existen tres tipos de bucles:

- **Mientras:** Este tipo de bucle se utiliza generalmente cuando no se sabe el número exacto de iteraciones, y el límite está dado por una condición. Mientras la condición sea verdadera, el bucle continuará ejecutándose.
- **Repetir – Hasta Que:** Este tipo de bucle se ejecuta al menos una vez, ya que la comprobación de la condición se realiza al final. Puede ser útil en programas donde, dependiendo del resultado de la primera iteración, se decida si se sigue iterando o no.
- **Para – Hasta:** Este bucle, a diferencia de los anteriores, se usa cuando se sabe exactamente el número de iteraciones que se desean realizar, sin depender de una condición. Para ejecutar este bucle, necesitamos un valor inicial y un valor final que indica dónde se detiene el bucle.

Figura 3

Estructura de Bucle de repetición.



Bucle mientras

Sintaxis:

Mientras (condición) Hacer

acciones

Fin Mientras

En el ejemplo:

- Hemos definido un **iterador** y un **límite**, los cuales son ingresados por teclado.

- Luego usamos el **bucle Mientras**, que trabaja con la condición: “iterador menor que límite”. Esta condición permite que el bucle se repita siempre que sea verdadera.
- En las acciones, podemos ver cómo se imprime el valor del iterador y luego se incrementa en 1 en cada iteración.
- Al finalizar la ejecución del bucle, se han realizado 5 iteraciones.

Pseudocódigo

```

1  Algoritmo IncrementarValor
2      Definir iterador, limite Como Entero;
3      iterador = 1;
4      Escribir 'Ingrese el límite: ';
5      Leer limite;
6      Mientras iterador ≤ limite Hacer
7          Escribir 'El valor del iterador es: ', iterador;
8          iterador = iterador + 1;
9      Fin Mientras
10 FinAlgoritmo

```

Salida

PSInt - Ejecutando proceso INCREMENTARVALOR

— □ X

*** Ejecución Iniciada. ***

Ingrese el límite:

> 5

El valor del iterador es: 1

El valor del iterador es: 2

El valor del iterador es: 3

El valor del iterador es: 4

El valor del iterador es: 5

*** Ejecución Finalizada. ***

Bucle repetir

Sintaxis:

Repetir

acciones

Hasta Que (condición)

En el ejemplo:

- Observa cómo el bucle ejecuta el código una primera vez y luego comprueba la condición.
- El bucle continuará iterando (repitiendo) hasta que el valor del iterador sea mayor que el límite. Cuando se cumpla esta condición, el bucle se detendrá.
- El bucle imprime 5 valores.

Pseudocódigo

```
1  Algoritmo IncrementarValorRepetir
2      Definir iterador, limite Como Entero;
3      iterador = 1;
4      Escribir 'Ingrese el límite: ';
5      Leer limite;
6
7      Repetir
8          Escribir 'El valor del iterador es: ', iterador;
9          iterador = iterador + 1;
10     Hasta Que iterador > limite;
11
12 FinAlgoritmo
```

Salida

```
PSInt - Ejecutando proceso INCREMENTARVALORREPETIR  -  □  X
*** Ejecución Iniciada. ***
Ingrese el límite:
> 5
El valor del iterador es: 1
El valor del iterador es: 2
El valor del iterador es: 3
El valor del iterador es: 4
El valor del iterador es: 5
*** Ejecución Finalizada. ***
```

Bucle Para - Hasta

Sintaxis:

Para variable <- valor inicial hasta límite con pasos valor hacer

acciones

Fin Para

Los valores de valor inicial y límite siempre son enteros.

En el siguiente ejemplo:

- En este bucle se conoce el número exacto de iteraciones que se van a ejecutar. El iterador comienza en 1 y el límite es 5, por lo que se realizarán 5 iteraciones (vueltas).
- Observa que en el cuerpo del bucle no hay ninguna sentencia que aumente en 1 el valor del iterador; simplemente se imprime su valor.
- Con este tipo de bucle, el **incremento del iterador** se realiza en la parte denominada **Paso 1**. En este paso, podemos modificar el valor 1 por el incremento que deseemos aplicar en cada iteración.

Pseudocódigo

```
1  Algoritmo IncrementarValorPara
2      Definir iterador, limite Como Entero;
3      Escribir 'Ingrese el límite: ';
4      Leer limite;
5
6      Para iterador ← 1 Hasta limite Con Paso 1 Hacer
7          Escribir 'El valor del iterador es: ', iterador;
8      Fin Para
9
10 FinAlgoritmo
```

Salida

```
PSInt - Ejecutando proceso INCREMENTARVALORPARA
*** Ejecución Iniciada. ***
Ingrese el límite:
> 5
El valor del iterador es: 1
El valor del iterador es: 2
El valor del iterador es: 3
El valor del iterador es: 4
El valor del iterador es: 5
*** Ejecución Finalizada. ***
```

REFERENCIAS

- De Roer, D. D., & De Roer, D. D. (2022, 12 junio). Variables y constantes en pseudocódigo. *Disco Duro de Roer*. <https://www.discoduroderoer.es/variables-y-constantes-en-pseudocodigo/>
- González, J. D. M. (2020, 26 abril). Final y constantes. <https://www.programarya.com/Cursos/Java/Sistema-de-Tipos/Final-y-Constantes>
- Gómez, J., & Morales, A. (2018). *Estructuras de datos y algoritmos en pseudocódigo*. Editorial Rama.
- Martínez, J., & Gutiérrez, P. (2019). *Fundamentos de programación con pseudocódigo*. Ediciones Académicas.
- Rodríguez, M., & Pérez, D. (2020). *Programación avanzada: Técnicas y herramientas*. Editorial Alfaomega.
- Variables en pseudocódigo. (s. f.). *Abrirllave.com*. <https://www.abrirllave.com/pseudocodigo/variables.php>

GLOSARIO

Bucle: Un bucle es una estructura de control en la programación que permite ejecutar repetidamente un conjunto de instrucciones hasta que se cumpla una condición específica. Los bucles son fundamentales para automatizar tareas repetitivas y procesar secuencias de datos de manera eficiente. Existen diferentes tipos de bucles, como *for* (para), *while* (mientras) y *do-while* (repetir hasta que), cada uno adecuado para distintas situaciones, según cómo y cuándo se evalúa la condición de finalización (Rodríguez & Gutiérrez, 2019).

Sentencias: Las sentencias son las instrucciones individuales en un programa que ejecutan una acción específica. Pueden ser de varios tipos, como sentencias de asignación, que almacenan valores en variables; sentencias de control, que alteran el flujo de ejecución del programa (como condicionales y bucles); y sentencias de entrada/salida, que permiten la interacción con el usuario o con otros sistemas. Las sentencias son los bloques fundamentales que, cuando se combinan, forman el algoritmo o el programa completo (López & Pérez, 2020).



La excelencia no se improvisa

síguenos

