



UNIVERSIDAD NACIONAL DE TRUJILLO  
UNT

# ESTRUCTURA DE CONTROL REPETITIVA O DE ITERACIÓN CONDICIONAL



For  
While  
Do While



# DEFINICIÓN

Se utilizan cuando en el desarrollo de la solución de un problema se hace necesario ejecutar una serie de instrucciones un número repetido de veces. El conjunto de instrucciones que se ejecuta repetidamente se llama ciclo o bucle. Cada vez que se ejecuta el bucle se dice que se ha producido una iteración.

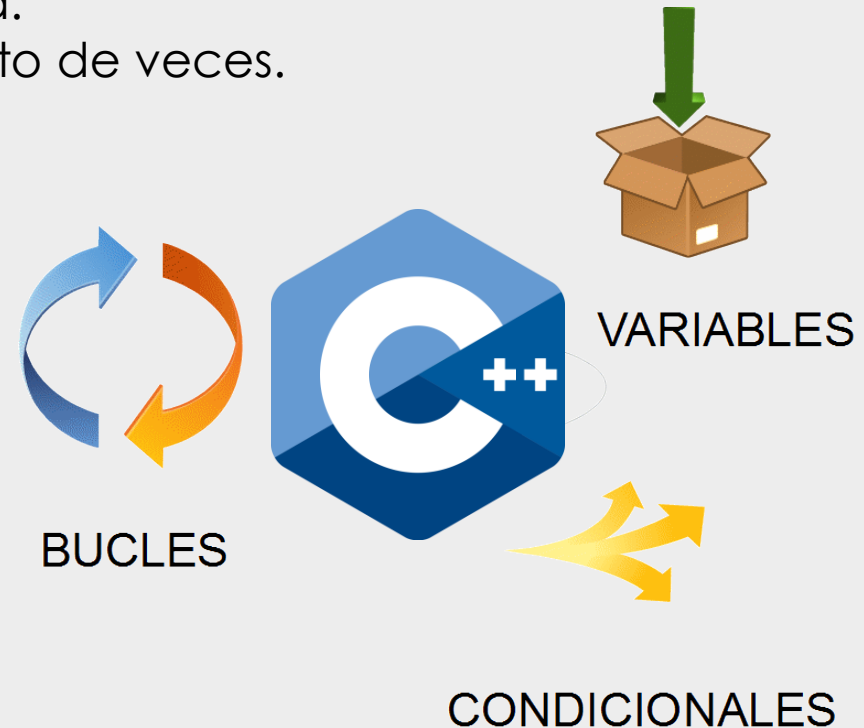




# Características de los ciclos

Deben incluir una condición de parada.  
Deben finalizar luego de un número finito de veces.  
C++ ofrece 3 procesos Repetitivos:

- 🎵 while
- 🎵 do ... while
- 🎵 for





# CONTADOR

Un contador es una forma de controlar a un bucle. Es una variable cuyo valor se incrementa o decrementa en una cantidad constante cada vez que se produce un determinado suceso o acción en cada repetición; dicha variable controla o determina la cantidad de veces que se repite un proceso o dato.

PSEUDOCODIGO	CODIFICACION
<b>entero contador</b> ← 1 <b>contador</b> ← <b>contador</b> +1	int contador = 1; //variable con valor inicial de 1 contador++; o ++contador; o contador+=1;



# ACUMULADOR O TOTALIZADOR

Un acumulador realiza la misma función que un contador con la diferencia de que el incremento o decremento es variable en lugar de constante.

Es una variable que acumula sobre sí misma un conjunto de valores, para de esta manera tener la acumulación de todos ellos en una sola variable.

PSEUDOCODIGO	CODIFICACION
<b>entero acumulador</b> $\leftarrow$ 0 <b>acumulador</b> $\leftarrow$ <b>acumulador</b> + <b>valor</b>	int acumulador = 0; acumulador = acumulador + valor;   o acumulador += valor;



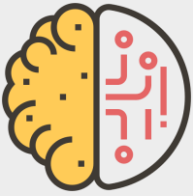
# CENTINELA

El centinela es una variable que inicia con un valor, luego dentro de un bucle este valor cambia, haciendo falsa la condición del ciclo y por lo tanto indicará el fin del ciclo (el usuario puede determinar cuándo hacerlo).

La repetición controlada por centinela se considera como una repetición indefinida (se desconoce el número de repeticiones).

## EJEMPLO

Un usuario puede introducir S o N para indicar si desea continuar o no. El bucle terminará cuando la respuesta del usuario sea "n" o "N". También se podrían manejar números.



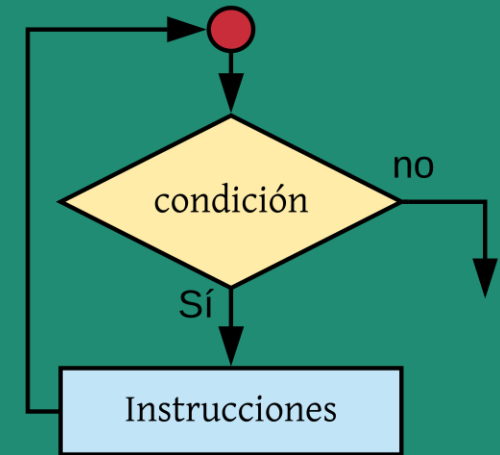
# BANDERA

Una bandera (flag, interruptor o marca), es una variable que puede tomar sólo dos valores opuestos, generalmente: 1 (verdadero) o 0 (falso), a lo largo de la ejecución del algoritmo o programa. Es muy utilizado en búsquedas.

PSEUDOCODIGO	CODIFICACION
<b>int band ← 0</b> <b>bandera ← 1</b>	<b>int band = 0;</b> <b>bandera = 1;</b>

# ESTRUCTURA DE CONTROL REPETITIVA MIENTRAS (WHILE)

## DIAGRAMA



## PSEUDOCODIGO

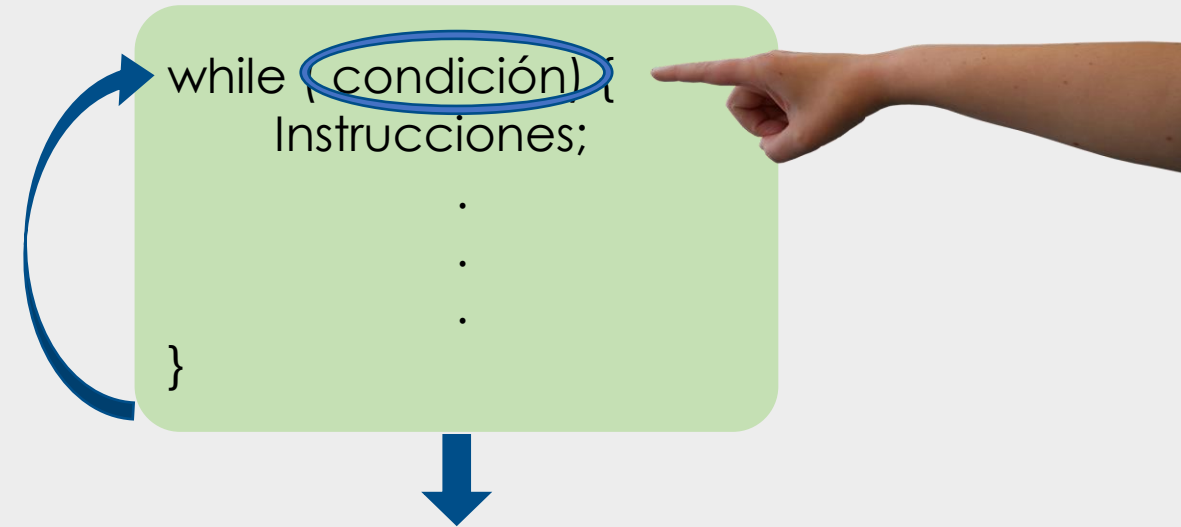
```
 mientras (condición)  
    Instrucciones  
fin mientras
```





## CODIFICACION

Al ejecutarse la instrucción mientras (while), se evaluará la expresión booleana suministrada en los paréntesis **(condición)**, y si su valor es verdadero (distinto de cero) se realizará el ciclo o bucle (una o varias instrucciones).



Después, la condición es reevaluada y se procede de la misma manera.

Cuando la condición se vuelve falsa (es decir, cero), en la siguiente evaluación se dará por terminado el ciclo mientras (while). La condición es cualquier expresión numérica, relacional o lógica.



**01.** Desarrollar una aplicación que muestre los 10 primeros números enteros positivos.

### **Algoritmo números**

```
entero i  
i ← 1  
escribir "Listar los 10 primeros números"  
mientras i ≤ 10 hacer  
    escribir "Número: ", i  
    i ← i + 1  
fin_mientras
```

**fin\_algoritmo**



CODIFICACION	SALIDA DE PANTALLA
<pre>#include &lt;iostream&gt; using namespace std;  int main(int argc, char *argv[]) {     int i;     i=1;     cout&lt;&lt;"Listar los 10 primeros números"&lt;&lt;endl;     while(i&lt;=10){         cout&lt;&lt;"Número: "&lt;&lt;i&lt;&lt;endl;         i++;     }     return 0; }</pre>	Listar los 10 primeros números Número: 1 Número: 2 Número: 3 Número: 4 Número: 5 Número: 6 Número: 7 Número: 8 Número: 9 Número: 10



**02.** Desarrolle una aplicación que muestre los números del 20 al 80 de 5 en 5

### **Algoritmo\_Sucesion**

entero i

i  $\leftarrow$  20

escribir “Sucesión: ”

mientras i  $\leq$  80 hacer

    escribir "Número: ", i

    i  $\leftarrow$  i+5

fin\_mientras

**fin\_algoritmo**



CODIFICACION	SALIDA DE PANTALLA
<pre>1  #include &lt;iostream&gt; 2  using namespace std; 3 4  int main(int argc, char *argv[]) { 5      int i; 6      i=20; 7      cout&lt;&lt;"Sucesión: "&lt;&lt;endl; 8      while(i&lt;=80){ 9          cout&lt;&lt;"Número: "&lt;&lt;i&lt;&lt;endl; 10         i=i+5; 11     } 12     return 0; 13 } 14</pre>	Sucesión: Número: 20 Número: 25 Número: 30 Número: 35 Número: 40 Número: 45 Número: 50 Número: 55 Número: 60 Número: 65 Número: 70 Número: 75 Número: 80



**03.** Desarrolle una aplicación que muestre los números del 100 al 50 de -2 en -2

### Algoritmo números

entero i

$i \leftarrow 100$

escribir "Listar los 10 primeros números"

mientras  $i \geq 50$  hacer

    escribir "Número: ", i

$i \leftarrow i - 2$

fin\_mientras

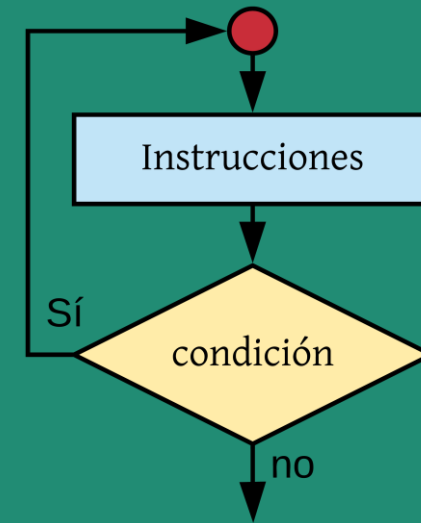
**fin\_algoritmo**



CODIFICACION	SALIDA DE PANTALLA																												
<pre>1  #include &lt;iostream&gt; 2  using namespace std; 3 4  int main(int argc, char *argv[]) { 5      int i; 6      i=100; 7      cout&lt;&lt;"Sucesión: "&lt;&lt;endl; 8      while(i&gt;=50){ 9          cout&lt;&lt;"Número: "&lt;&lt;i&lt;&lt;endl; 10         i=i-2; 11     } 12     return 0; 13 } 14</pre>	<p>Sucesión:</p> <table><tr><td>Número: 100</td><td>Número: 72</td></tr><tr><td>Número: 98</td><td>Número: 70</td></tr><tr><td>Número: 96</td><td>Número: 68</td></tr><tr><td>Número: 94</td><td>Número: 66</td></tr><tr><td>Número: 92</td><td>Número: 64</td></tr><tr><td>Número: 90</td><td>Número: 62</td></tr><tr><td>Número: 88</td><td>Número: 60</td></tr><tr><td>Número: 86</td><td>Número: 58</td></tr><tr><td>Número: 84</td><td>Número: 56</td></tr><tr><td>Número: 82</td><td>Número: 54</td></tr><tr><td>Número: 80</td><td>Número: 52</td></tr><tr><td>Número: 78</td><td>Número: 50</td></tr><tr><td>Número: 76</td><td></td></tr><tr><td>Número: 74</td><td></td></tr></table>	Número: 100	Número: 72	Número: 98	Número: 70	Número: 96	Número: 68	Número: 94	Número: 66	Número: 92	Número: 64	Número: 90	Número: 62	Número: 88	Número: 60	Número: 86	Número: 58	Número: 84	Número: 56	Número: 82	Número: 54	Número: 80	Número: 52	Número: 78	Número: 50	Número: 76		Número: 74	
Número: 100	Número: 72																												
Número: 98	Número: 70																												
Número: 96	Número: 68																												
Número: 94	Número: 66																												
Número: 92	Número: 64																												
Número: 90	Número: 62																												
Número: 88	Número: 60																												
Número: 86	Número: 58																												
Número: 84	Número: 56																												
Número: 82	Número: 54																												
Número: 80	Número: 52																												
Número: 78	Número: 50																												
Número: 76																													
Número: 74																													

# ESTRUCTURA DE CONTROL REPETITIVA HACER\_MIENTRAS (DO WHILE)

## DIAGRAMA



## PSEUDOCODIGO

hacer  
    Instrucciones  
mientras (condición)





En este proceso primero se realizan las instrucciones y luego se verifica la condición, si esta es verdadera, se realizan de nuevo las Instrucciones. Este proceso se ejecuta hasta que la condición llegue a ser falsa.

“do...while” es adecuado cuando no sabemos el número de veces que se debe repetir un ciclo, pero conocemos que se debe ejecutar por lo menos una vez.

## CODIFICACION

```
do{  
    Instrucciones;  
    .  
    .  
    .  
}while(condición);
```



A diferencia del while, al final del do\_while sí lleva punto y coma después de la instrucción while (condición). Donde condición es cualquier expresión numérica, relacional o lógica.





**01.** Desarrollar una aplicación que muestre los 10 primeros números enteros positivos.

### Algoritmo enteros\_positivos

entero i

$i \leftarrow 1$

escribir "Listar los 10 primeros números: "

hacer

    escribir "Número: ", i

$i \leftarrow i + 1$

mientras  $i \leq 10$

**fin\_algoritmo**



CODIFICACION	SALIDA DE PANTALLA
<pre>#include &lt;iostream&gt; using namespace std;  int main(int argc, char *argv[]) {     int i;     i=1;     cout&lt;&lt;"Listar los 10 primeros números"&lt;&lt;endl;     do{         cout&lt;&lt;"Número: "&lt;&lt;i&lt;&lt;endl;         i++;     } while(i&lt;11);     return 0; }</pre>	Listar los 10 primeros números Número: 1 Número: 2 Número: 3 Número: 4 Número: 5 Número: 6 Número: 7 Número: 8 Número: 9 Número: 10



**02.** Desarrolle una aplicación que muestre los números del 20 al 80 de 5 en 5

### Algoritmo\_Sucesion

entero i

$i \leftarrow 20$

escribir "Sucesión: "

hacer

    escribir "Número: ", i

$i \leftarrow i + 5$

mientras  $i < 81$

fin\_algoritmo



CODIFICACION	SALIDA DE PANTALLA
<pre>1  #include &lt;iostream&gt; 2  using namespace std; 3 4  int main(int argc, char *argv[]) { 5      int i; 6      i=20; 7      cout&lt;&lt;"Sucesión: "&lt;&lt;endl; 8      do{ 9          cout&lt;&lt;"Número: "&lt;&lt;i&lt;&lt;endl; 10         i=i+5; 11     }while(i&lt;81); 12     return 0; 13 } 14</pre>	<p>Sucesión:</p> <p>Número: 20</p> <p>Número: 25</p> <p>Número: 30</p> <p>Número: 35</p> <p>Número: 40</p> <p>Número: 45</p> <p>Número: 50</p> <p>Número: 55</p> <p>Número: 60</p> <p>Número: 65</p> <p>Número: 70</p> <p>Número: 75</p> <p>Número: 80</p>



**03.** Desarrolle una aplicación que muestre los números del 100 al 50 de -2 en -2

### **Algoritmo\_Sucesion**

entero i

$i \leftarrow 100$

escribir "Sucesión: "

hacer

    escribir "Número: ", i

$i \leftarrow i - 2$

mientras  $i \geq 50$

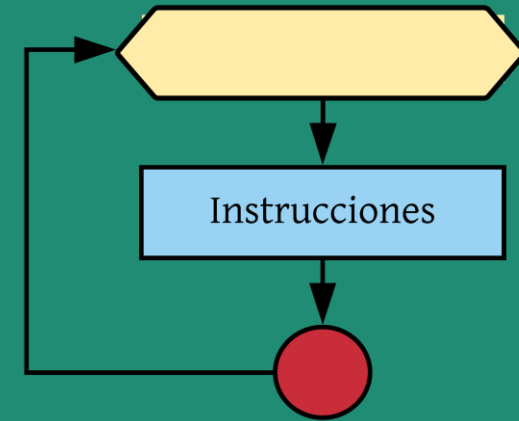
**fin\_algoritmo**



CODIFICACION	SALIDA DE PANTALLA																												
<pre>1  #include &lt;iostream&gt; 2  using namespace std; 3 4  int main(int argc, char *argv[]) { 5      int i; 6      i=100; 7      cout&lt;&lt;"Sucesión: "&lt;&lt;endl; 8      do{ 9          cout&lt;&lt;"Número: "&lt;&lt;i&lt;&lt;endl; 10         i=i-2; 11     }while(i&gt;=50); 12     return 0; 13 }</pre>	<p>Sucesión:</p> <table><tr><td>Número: 100</td><td>Número: 72</td></tr><tr><td>Número: 98</td><td>Número: 70</td></tr><tr><td>Número: 96</td><td>Número: 68</td></tr><tr><td>Número: 94</td><td>Número: 66</td></tr><tr><td>Número: 92</td><td>Número: 64</td></tr><tr><td>Número: 90</td><td>Número: 62</td></tr><tr><td>Número: 88</td><td>Número: 60</td></tr><tr><td>Número: 86</td><td>Número: 58</td></tr><tr><td>Número: 84</td><td>Número: 56</td></tr><tr><td>Número: 82</td><td>Número: 54</td></tr><tr><td>Número: 80</td><td>Número: 52</td></tr><tr><td>Número: 78</td><td>Número: 50</td></tr><tr><td>Número: 76</td><td></td></tr><tr><td>Número: 74</td><td></td></tr></table>	Número: 100	Número: 72	Número: 98	Número: 70	Número: 96	Número: 68	Número: 94	Número: 66	Número: 92	Número: 64	Número: 90	Número: 62	Número: 88	Número: 60	Número: 86	Número: 58	Número: 84	Número: 56	Número: 82	Número: 54	Número: 80	Número: 52	Número: 78	Número: 50	Número: 76		Número: 74	
Número: 100	Número: 72																												
Número: 98	Número: 70																												
Número: 96	Número: 68																												
Número: 94	Número: 66																												
Número: 92	Número: 64																												
Número: 90	Número: 62																												
Número: 88	Número: 60																												
Número: 86	Número: 58																												
Número: 84	Número: 56																												
Número: 82	Número: 54																												
Número: 80	Número: 52																												
Número: 78	Número: 50																												
Número: 76																													
Número: 74																													

# ESTRUCTURA DE CONTROL REPETITIVA PARA (FOR)

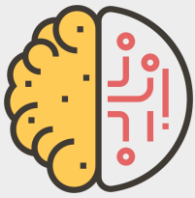
## DIAGRAMA



## PSEUDOCODIGO

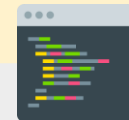
```
para (exp1; exp2; exp3)  
    Instrucciones  
fin para
```





El para (for) es la estructura repetitiva más utilizada y simple de manejar, ya que repite un conjunto de instrucciones un número determinado de veces. Una de sus aplicaciones principales son los arreglos.

“for”, es equivalente a un “while”, pero la sintaxis habitual de la orden “for” hace que sea especialmente útil cuando sabemos exactamente cuántas veces queremos que se repita (por ejemplo: 10 veces sería “for (i=1; i<=10; i++)”).





Dónde:

- ✓ **exp1:** Instrucciones de Inicialización. Se ejecuta 1 vez y se va a exp2.
- ✓ **exp2:** Condición que controla el proceso Repetitivo. Si esta expresión es verdadera se ejecutan las Instrucciones y se va a exp3. Si esta expresión es falsa el for termina.
- ✓ **exp3:** Instrucciones de incremento de variables. Se ejecutan y luego se va a exp2.

## CODIFICACION

```
for ( exp1; exp2; exp3) {  
    Instrucciones;  
    .  
    .  
    .  
}
```



**01.** Desarrollar una aplicación que muestre los 10 primeros números enteros positivos.

### Algoritmo números

```
entero i  
escribir "Listar los 10 primeros números: "  
Para i  $\leftarrow$  1 hasta 10 inc 1 hacer  
    escribir " Número: ", i  
fin_para
```

**fin\_algoritmo**



CODIFICACION	SALIDA DE PANTALLA
<pre>#include &lt;iostream&gt; using namespace std;  int main(int argc, char *argv[]) {     int i;     cout&lt;&lt;"Listar los 10 primeros números"&lt;&lt;endl;     for(i=1;i&lt;=10;i++){         cout&lt;&lt;"Número: "&lt;&lt;i&lt;&lt;endl;     }     return 0; }</pre>	Listar los 10 primeros números Número: 1 Número: 2 Número: 3 Número: 4 Número: 5 Número: 6 Número: 7 Número: 8 Número: 9 Número: 10



**02.** Desarrolle una aplicación que muestre los números del 20 al 80 de 5 en 5

### **Algoritmo\_Sucesion**

entero i

escribir "Sucesión: "

Para i  $\leftarrow$  20 hasta 80 inc 5 hacer

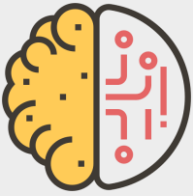
    escribir " Número: ", i

fin\_para

**fin\_algoritmo**



CODIFICACION	SALIDA DE PANTALLA
<pre>1  #include &lt;iostream&gt; 2  using namespace std; 3 4  int main(int argc, char *argv[]) { 5      int i; 6      cout&lt;&lt;"Sucesión: "&lt;&lt;endl; 7      for(i=20;i&lt;=80;i+=5){ 8          cout&lt;&lt;"Número: "&lt;&lt;i&lt;&lt;endl; 9      } 10     return 0;</pre>	Sucesión: Número: 20 Número: 25 Número: 30 Número: 35 Número: 40 Número: 45 Número: 50 Número: 55 Número: 60 Número: 65 Número: 70 Número: 75 Número: 80



**03.** Desarrolle una aplicación que muestre los números del 100 al 50 de -2 en -2

DIAGRAMA DE FLUJO	PSEUDOCODIGO
<pre>graph TD; Inicio([Inicio]) --&gt; Proceso{{i ← 100, 50, -2}}; Proceso --&gt; Salida[i]; Salida --&gt; Decisión((i)); Decisión --&gt; Inicio; Decisión --&gt; Fin([Fin]);</pre>	<p><b>Algoritmo números</b></p> <p>entero i escribir "Sucesión: " Para i ← 100 hasta 50 inc -2 hacer     escribir " Número: ", i fin_para</p> <p><b>fin_algoritmo</b></p>



CODIFICACION	SALIDA DE PANTALLA																												
<pre>1  #include &lt;iostream&gt; 2  using namespace std; 3 4  int main(int argc, char *argv[]) { 5      int i; 6      cout&lt;&lt;"Sucesión: "&lt;&lt;endl; 7      for(i=100;i&gt;=50;i-=2){ 8          cout&lt;&lt;"Número: "&lt;&lt;i&lt;&lt;endl; 9      } 10     return 0;</pre>	<p>Sucesión:</p> <table><tr><td>Número: 100</td><td>Número: 72</td></tr><tr><td>Número: 98</td><td>Número: 70</td></tr><tr><td>Número: 96</td><td>Número: 68</td></tr><tr><td>Número: 94</td><td>Número: 66</td></tr><tr><td>Número: 92</td><td>Número: 64</td></tr><tr><td>Número: 90</td><td>Número: 62</td></tr><tr><td>Número: 88</td><td>Número: 60</td></tr><tr><td>Número: 86</td><td>Número: 58</td></tr><tr><td>Número: 84</td><td>Número: 56</td></tr><tr><td>Número: 82</td><td>Número: 54</td></tr><tr><td>Número: 80</td><td>Número: 52</td></tr><tr><td>Número: 78</td><td>Número: 50</td></tr><tr><td>Número: 76</td><td></td></tr><tr><td>Número: 74</td><td></td></tr></table>	Número: 100	Número: 72	Número: 98	Número: 70	Número: 96	Número: 68	Número: 94	Número: 66	Número: 92	Número: 64	Número: 90	Número: 62	Número: 88	Número: 60	Número: 86	Número: 58	Número: 84	Número: 56	Número: 82	Número: 54	Número: 80	Número: 52	Número: 78	Número: 50	Número: 76		Número: 74	
Número: 100	Número: 72																												
Número: 98	Número: 70																												
Número: 96	Número: 68																												
Número: 94	Número: 66																												
Número: 92	Número: 64																												
Número: 90	Número: 62																												
Número: 88	Número: 60																												
Número: 86	Número: 58																												
Número: 84	Número: 56																												
Número: 82	Número: 54																												
Número: 80	Número: 52																												
Número: 78	Número: 50																												
Número: 76																													
Número: 74																													



# EJERCICIOS RESUELTOS



**01.** Desarrollar una aplicación que permita calcular la sumatoria de todos los números desde el 1 hasta el número ingresado por el usuario a través del teclado e imprima el resultado.



### PSEUDOCODIGO

#### Algoritmo suma\_números

```
entero i, suma, n
suma ← 0
hacer
    escribir "Valor de n: "
    leer n
mientras n <= 0
Para i ← 1 hasta n inc 1 hacer
    suma ← suma + i
fin_para
escribir "La suma de 1 hasta ", n, " es :",
suma
```

**fin\_algoritmo**



## PSEUDOCODIGO

### Algoritmo suma\_números

entero i, suma, n

suma  $\leftarrow$  0

hacer

    escribir "Valor de n: "

    leer n

mientras n  $\leq$  0

Para i  $\leftarrow$  1 hasta n inc 1 hacer

    suma  $\leftarrow$  suma + i

fin\_para

escribir "La suma de 1 hasta ", n, " es :",

suma

**fin\_algoritmo**



CODIFICACION	SALIDA DE PANTALLA
<pre>1  #include &lt;iostream&gt; 2  using namespace std; 3 4  int main(int argc, char *argv[]) { 5      int i, n, suma=0; 6      do{ 7          cout&lt;&lt;"Valor de n: "; 8          cin&gt;&gt;n; 9      } while(n&lt;=0); 10     for(i=1;i&lt;=n;i++) 11     { 12         suma=suma+i; 13     } 14     cout&lt;&lt;"La suma de 1 hasta "&lt;&lt;n&lt;&lt;" es: "&lt;&lt;suma&lt;&lt;endl; 15     return 0; 16 }</pre>	<p>Valor de n: -2 Valor de n: 0 Valor de n: 4 La suma de 1 hasta 4 es: 10</p>



**02.** Desarrollar una aplicación que permita imprimir en pantalla la tabla de multiplicar de cualquier número entero ingresado por el usuario a través del teclado.

### PSEUDOCODIGO

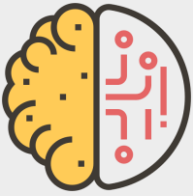
#### Algoritmo tabla\_de\_multiplicar

```
entero i, n
hacer
    escribir " Ingresa un número entero: "
    leer n
mientras n <= 0
    escribir " Tabla de Multiplicar de ", n
    Para i ← 1 hasta 12 inc 1 hacer
        escribir i, " x ", n, " = ", i*n
    fin_para
```

**fin\_algoritmo**



CODIFICACION	SALIDA DE PANTALLA
<pre>1  #include &lt;iostream&gt; 2  using namespace std; 3 4  int main(int argc, char *argv[]) { 5      int i, n; 6      do{ 7          cout&lt;&lt;"Ingresa un número entero: "; 8          cin&gt;&gt;n; 9      } while(n&lt;=0); 10     cout&lt;&lt;"Tabla de Multiplicar del "&lt;&lt;n&lt;&lt;endl; 11     cout&lt;&lt;"-----"&lt;&lt;endl; 12     for(i=1;i&lt;=12;i++) 13     { 14 15         cout&lt;&lt;i&lt;&lt;" x "&lt;&lt;n&lt;&lt;" = "&lt;&lt;(i*n)&lt;&lt;endl; 16     } 17     return 0; 18 }</pre>	<p>Ingresa un número entero: -5 Ingresa un número entero: 0 Ingresa un número entero: 5 Tabla de Multiplicar del 5 ----- 1 x 5 = 5 2 x 5 = 10 3 x 5 = 15 4 x 5 = 20 5 x 5 = 25 6 x 5 = 30 7 x 5 = 35 8 x 5 = 40 9 x 5 = 45 10 x 5 = 50 11 x 5 = 55 12 x 5 = 60</p>



**03.** Ingresar N números enteros y reportar la cantidad de pares y la cantidad de impares.

```
Para i ← 1 hasta n inc 1 hacer
    escribir "Ingrese número: "
    Leer x
    Si  $x \bmod 2 = 0$  entonces
         $cp \leftarrow cp + 1$ 
    sino
         $ci \leftarrow ci + 1$ 
    fin_si
fin_para
escribir "La cantidad de pares es: ", cp
escribir "La cantidad de impares es: ", ci

fin_algoritmo
```



CODIFICACION	SALIDA DE PANTALLA
<pre>#include &lt;iostream&gt; using namespace std;  int main(int argc, char *argv[]) {     int i,n,x,cp=0,ci=0;     do{         cout&lt;&lt;"Cantidad de números : ";         cin&gt;&gt;n;     }while(n&lt;=0);     for(i=1;i&lt;=n;i++)     {         cout&lt;&lt;"Ingrese número : ";         cin&gt;&gt;x;         if(x%2==0) cp = cp + 1;         else ci = ci + 1;     }     cout&lt;&lt;"La cantidad de pares es : "&lt;&lt;cp&lt;&lt;endl;     cout&lt;&lt;"La cantidad de impares es : "&lt;&lt;ci&lt;&lt;endl;     return 0; }</pre>	<p>Cantidad de números : 5</p> <p>Ingrese número : 1</p> <p>Ingrese número : 2</p> <p>Ingrese número : 3</p> <p>Ingrese número : 4</p> <p>Ingrese número : 5</p> <p>La cantidad de pares es : 2</p> <p>La cantidad de impares es : 3</p>





**04.** Se desea calcular independientemente la suma de los pares e impares de los numero desde el 1 hasta el 50.

### **Algoritmo suma\_pares\_impares**

```
entero i, sp, si
sp  $\leftarrow$  0, si  $\leftarrow$  0
Para i  $\leftarrow$  1 hasta 50 inc 1 hacer
    Si i mod 2=0 entonces
        sp  $\leftarrow$  sp + i
    sino
        si  $\leftarrow$  si + i
    fin_si
fin_para
escribir "La suma de pares es: ", sp
escribir "La suma de impares es: ", si
```

**fin\_algoritmo**



CODIFICACION	SALIDA DE PANTALLA
<pre>#include &lt;iostream&gt; using namespace std;  int main(int argc, char *argv[]) {     int i,sp=0,si=0;     for(i=1;i&lt;=50;i++)         if(i%2==0) sp=sp+i;         else si=si+i;     cout&lt;&lt;"La suma de pares es : "&lt;&lt;sp&lt;&lt;endl;     cout&lt;&lt;"La suma de impares es : "&lt;&lt;si&lt;&lt;endl;     return 0; }</pre>	<p>La suma de pares es : 650 La suma de impares es : 625</p>



## 05. Calcular el factorial de un número entero.

### PSEUDOCODIGO

#### Algoritmo calculo\_factorial

entero i, num

Real f

$f \leftarrow 1$

Hacer

    escribir "Ingrese el número: "

    leer num

mientras num  $\leq 0$

    Para i  $\leftarrow 1$  hasta num inc 1 hacer

$f \leftarrow f * i$

    fin\_para

    escribir "El factorial de ", num, "! es: ", f

fin\_algoritmo



CODIFICACION	SALIDA DE PANTALLA
<pre>1  #include &lt;iostream&gt; 2  using namespace std; 3 4  int main(int argc, char *argv[]) { 5      int i, num; 6      double f=1; 7      do{ 8          cout&lt;&lt;"Ingresa un número: "; 9          cin&gt;&gt;num; 10     } while(num&lt;=0); 11     for(i=1;i&lt;=num;i++) 12         f=f*i; 13     cout&lt;&lt;"El factorial es: "&lt;&lt;f&lt;&lt;endl; 14 15     return 0; 16 }</pre>	<p>Ingresa un número: 6 El factorial es: 720</p>



**06.** Ingresar  $n$  números. Se pide calcular el promedio de ellos

### PSEUDOCODIGO

#### Algoritmo promedio

```
entero  $n, i$   
real  $x, s, p$   
 $s \leftarrow 0$   
Hacer  
    escribir "Cantidad de números a ingresar: "  
    leer  $n$   
    mientras  $n \leq 0$   
        Para  $i \leftarrow 1$  hasta  $n$  inc 1 hacer  
            escribir "Ingrese número: ",  $i$ , ": "  
            leer  $x$   
             $s \leftarrow s + x$   
        fin_para  
         $p \leftarrow s/n$   
        escribir "El Promedio de los ",  $n$ , " números  
        ingresados es: ",  $p$   
fin_algoritmo
```



CODIFICACION	SALIDA DE PANTALLA
<pre>#include &lt;iostream&gt; using namespace std;  int main(int argc, char *argv[]) {     int n,i;     double x,s=0,p;     do{         cout&lt;&lt;"Cantidad de números a ingresar: ";         cin&gt;&gt;n;     }while(n&lt;=0);     for(i=1;i&lt;=n;i++)     {         cout&lt;&lt;"Ingrese número : ";         cin&gt;&gt;x;         s=s+x;     }     p=s/n;     cout&lt;&lt;"El Promedio de los "&lt;&lt;n&lt;&lt;" números ingresados es: "&lt;&lt;p;     return 0; }</pre>	<p>Cantidad de números a ingresar: 3</p> <p>Ingrese número : 10</p> <p>Ingrese número : 20</p> <p>Ingrese número : 15</p> <p>El Promedio de los 3 números ingresados es : 15</p>



07. Ingresar n números y reportar el promedio de los positivos y el promedio de los negativos.



### PSEUDOCODIGO

#### Algoritmo promedio\_positivos\_negativos

entero n, i, x, sp, sn, cp, cn

real pp, pn

hacer

    escribir "Valor de n: "

    leer n

mientras n <= 0

Para i ← 1 hasta n inc 1 hacer

    escribir "Ingrese número: "

    leer x

    Si x > 0 entonces

        cp ← cp + 1

        sp ← sp + x

    sino

        Si x < 0 entonces

            cn ← cn + 1

            sn ← sn + x

        fin\_si

    fin\_si

fin\_para

Si cp > 0 entonces

    pp ← sp / cp

    escribir "Promedio de Positivos ", pp

sino

    escribir "No se Ingresaron Positivos"

fin\_si

Si cn > 0 entonces

    pn ← sn / cn

    escribir "Promedio de Negativos ", pn

sino

    escribir "No se Ingresaron negativos"

fin\_si

fin\_algoritmo



## CODIFICACION

```
#include <iostream>
using namespace std;

int main(int argc, char *argv[]) {
    int n,i,x,sp=0,sn=0,cp=0,cn=0;
    float pp,pn;
    do{
        cout<<"Valor de n : ";
        cin>>n;
    }while(n<=0);
    for(i=1;i<=n;i++)
    {
        cout<<"Ingrese número : ";
        cin>>x;
        if(x>0)
        {
            cp++;
            sp=sp+x;
        }
        else if(x<0)
        {
            cn++;
            sn=sn+x;
        }
    }
}
```

```
if(cp>0)
{
    pp=(float) sp/cp;
    cout<<"El Promedio de Positivos es : "<<pp<<endl;
}
else
    cout<<"No se Ingresaron Positivos"<<endl;
if(cn>0)
{
    pn=(float) sn/cn;
    cout<<"El Promedio de Negativos es : "<<pn<<endl;
}
else
    cout<<"No se Ingresaron Negativos"<<endl;
return 0;
}
```





## SALIDA DE PANTALLA

Valor de n : 10

Ingrese número : 4

Ingrese número : 7

Ingrese número : 3

Ingrese número : -1

Ingrese número : 5

Ingrese número : 6

Ingrese número : 11

Ingrese número : 12

Ingrese número : 14

Ingrese número : 23

El Promedio de Positivos es : 9.44444

El Promedio de Negativos es : -



**08.** Ingresar n números, Calcular el mayor y el menor de ellos.

### PSEUDOCODIGO

#### Algoritmo mayor\_menor

```
entero i, n
real x, mayor, menor
hacer
    escribir "Cantidad de números: "
    leer n
    mientras n <= 0
        mayor ← -1E30
        menor ← 1E30
```

```
Para i ← 1 hasta n inc 1 hacer
    escribir "Cantidad de números a ingresar: "
    leer x
    Si x > mayor entonces
        mayor ← x
    fin_si
    Si x < menor entonces
        menor ← x
    fin_si
fin_para
escribir "El mayor es: ", mayor
escribir "El menor es: ", menor
fin_algoritmo
```



CODIFICACION	SALIDA DE PANTALLA
<pre>#include &lt;iostream&gt; using namespace std;  int main(int argc, char *argv[]) {     int n,i;     double x,mayor,menor;     do{         cout&lt;&lt;"Cantidad de números a ingresar : ";         cin&gt;&gt;n;     }while(n&lt;=0);     mayor=-1e30;     menor=1e30;     for(i=1;i&lt;=n;i++)     {         cout&lt;&lt;"Ingrese número : ";         cin&gt;&gt;x;         if(x&gt;mayor) mayor=x;         if(x&lt;menor) menor=x;     }     cout&lt;&lt;"El mayor es : "&lt;&lt;mayor&lt;&lt;endl;     cout&lt;&lt;"El menor es : "&lt;&lt;menor&lt;&lt;endl;     return 0; }</pre>	<p>Cantidad de números a ingresar : 4 Ingrese número : 1 Ingrese número : 20 Ingrese número : 4 Ingrese número : 32 El mayor es : 32 El menor es : 1</p>