

# **Pseudocódigo + Diagramas de Flujo + Introducción a los lenguajes: C y Python.**

**Apuntes realizados por los profesores:**

**Jesús Núñez.**

**Antonio Jesús Conejo.**

## Sumario

Introducción.....	3
Proceso de creación de un proceso ejecutable.....	3
Elementos básicos:.....	3
Mi Primer Programa.....	5
Programas Secuenciales.....	5
Programas con instrucciones condicionales.....	6
Programas con instrucciones condicionales compuestas.....	7
Programas con instrucciones condicionales anidadas.....	9
Instrucción Switch.....	9
Instrucciones Repetitivas:.....	11
INSTRUCCIÓN MIENTRAS ... HACER.....	12
Sentencia Para o for.....	13

# Introducción

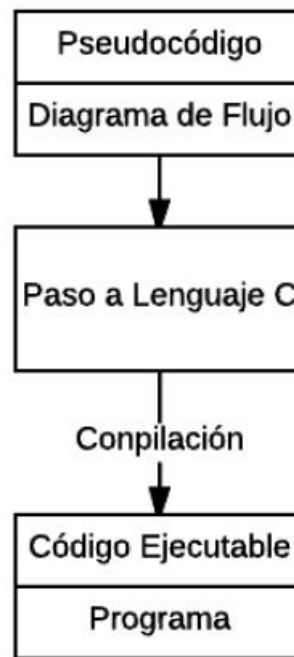
Con objeto de hacer un aprendizaje de la materia eminentemente práctico, vamos a aprender fundamentos de la programación mediante ejemplos. Estos ejemplos se realizaran en Pseudocódigo, Diagramas de flujo , Lenguaje C y Lenguaje Python..

## Proceso de creación de un proceso ejecutable

De un problema real analizamos que entrada de datos necesitamos, el proceso que hay que realizar y obtenemos un resultado. Para ello plasmamos una solución en forma de Pseudocódigo o diagrama de flujo, lo pasamos a un lenguaje de programación, lo compilamos o lo interpretamos para corregir errores obtener una aplicación o un fichero ejecutable o un script que luego podemos transformar en ejecutable.



shutterstock · 302671232



## Elementos básicos:

**Concepto de variable:** Objeto en el que podemos almacenar un dato. Los datos pueden ser de diferente tipo como entero (int), decimal (float), carácter (char),...

### Versión C.

*Ejemplo:*

En Lenguaje C	En Memoria
<code>int a;</code>	a--> <input type="text"/>
<code>A=14;</code>	a--> <input type="text" value="14"/>

### Versión Python.

Python es un lenguaje muy poco tipificado, no hace falta definir el tipo de datos antes de usarlo.

*Ejemplo:*

En Lenguaje Python	En Memoria
<code>A=14</code>	a--> <input type="text" value="14"/>

**Operaciones de escritura en pantalla:** Para escribir en pantalla.

En Lenguaje C	En Pantalla
<code>printf("Hola Mundo");</code>	Hola Mundo
<code>A=14; printf("El valor de a es %i", a);</code>	El valor de a es 14

**Versión Python.**

Casi igual que en C pero sin el ; final.

En Lenguaje Python	En Pantalla
<code>print("Hola Mundo")</code>	Hola Mundo
<code>a=14 print("El valor de a es", a)</code>	El valor de a es 14

**Operaciones de lectura desde teclado:** Para recoger datos introducidos desde el teclado.

En Lenguaje C	Significado
<code>scanf("%i",&amp;a);</code>	Almacena en <b>a</b> el valor que el usuario indique desde teclado.

**Versión Python.**

En Lenguaje Python	Significado
<code>a=input()</code>	Almacena en <b>a</b> el valor que el usuario indique desde teclado.
<code>a=input("Dime cuanto vale a: ")</code>	El programa pregunta "Dime cuanto vale a: "

**Operadores comunes a los dos lenguajes:**

Suma	+	Resta	-	Producto	*	División	/	Comparación	==
------	---	-------	---	----------	---	----------	---	-------------	----

**Estructura de un programa en C**

Código	Explicación
<code>#include &lt;nombre de la libreria&gt;</code>	Las librerías son elementos que contienen instrucciones para ser usadas. Si no incluyes las librerías las instrucciones serán desconocidas.
<code>int main() { Declaración de variables; Aquí van las instrucciones; }</code>	Inicio del programa. Se declaran todos los elementos que vamos a necesitar.  Fin

**Versión Python.**

No hay ninguna restricción en la estructura de un programa, puedes hacer lo que quieras. Lo único que el archivo en el que lo escribas lo grabas con la extensión .py

# Mi Primer Programa

Como es tradicional comenzamos nuestra clase de programación con el clásico “Hola Mundo”.

Pseudocódigo	Diagrama de Flujo	Lenguaje C
Algoritmo Saludo; Escribe Hola Mundo; Fin	 <pre>                     graph TD                         Inicio([Inicio]) --&gt; Print[/Hola Mundo/]                         Print --&gt; Fin([Fin])                     </pre>	<pre>                     #include &lt;stdio.h&gt;                     int main(){                     printf("Hola Mundo");                     }                 </pre>

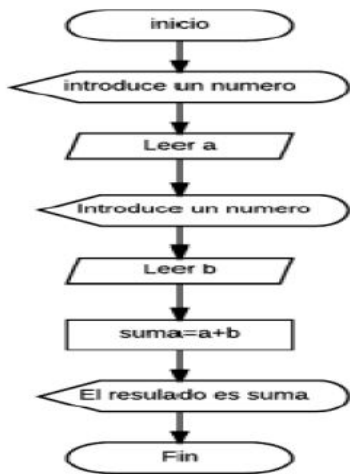
## Versión Python.

```
print("Hola Mundo")
```

# Programas Secuenciales

Un programa secuencial es aquel que ejecuta las instrucciones una detrás de otra en el mismo orden en el que están escritas.

Como ejemplo realizamos un algoritmo que realice la suma de dos números.

Pseudocódigo	Diagrama de Flujo	Lenguaje C
Algoritmo Suma; Escribe “Introduce número1”; Lee un número; Escribe “Introduce número2”; Lee otro número; Suma= número1 + número2; Escribe “El resultado es “ suma; Fin	 <pre>                     graph TD                         inicio([inicio]) --&gt; Input1[/introduce un numero/]                         Input1 --&gt; Process1[/Leer a/]                         Process1 --&gt; Input2[/Introduce un numero/]                         Input2 --&gt; Process2[/Leer b/]                         Process2 --&gt; Process3[suma=a+b]                         Process3 --&gt; Output[/El resultado es suma/]                         Output --&gt; Fin([Fin])                     </pre>	<pre>                     #include &lt;stdio.h&gt;                     int main(){                     int a,b,suma;                     printf("Introduce un numero");                     scanf("%i",&amp;a);                     printf("Introduce un numero");                     scanf("%i",&amp;b);                     suma=a+b;                     printf("La suma es %i",suma);                     }                 </pre>

## Versión Python.

```
a=input("Introduce un numero: ")
b=input("Introduce un numero: ")
suma=a+b
print "La suma es ",suma
```

```
Dime el valor de A: 12
Dime el valor de B: 23
La suma es 35
```

Si quieres lo grabas como **prueba1.py** o con el nombre que prefieras, luego compruebas si está bien. Por pantalla después de probarlo en el Sublime Text sale algo parecido, yo he metido los valores 12 y 23 para a y b respectivamente.

Ejercicio propuesto1: Algoritmo que lea dos números, calculando y escribiendo el valor de su suma, resta, producto y división.

## Programas con instrucciones condicionales

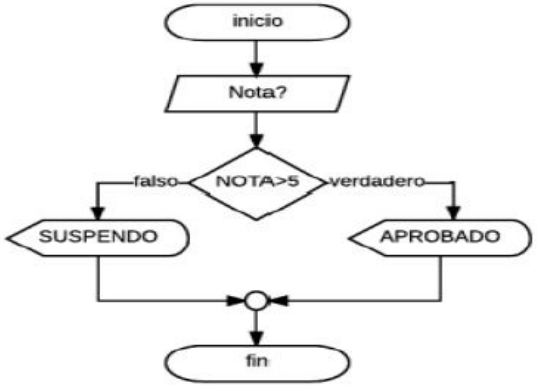
Dependiendo de si se cumple una condición se realizan unas instrucciones u otras.

Pseudocódigo	Diagrama de Flujo	Lenguaje C
Algoritmo Saludo; si condición hacer instrucciones; sino instrucciones; Fin		<pre>#include &lt;stdio.h&gt; int main(){     if (condición){         instrucciones;     }     else {         instrucciones;     } }</pre>
		<b>Lenguaje Python</b>
En Python lo único que hay que tener en cuenta es que las instrucciones debajo del if, deben de sesgarse o agregar una sangria a la derecha debajo de la condición a evaluar. Lo mismo para el else. No olvidar los dos puntos.		<pre>If condicion:     instrucciones else:     instrucciones</pre>

Para la construcción de condiciones tenemos los siguientes operadores

Menor que <	Mayor que >	Igual ==	Menor o igual que <=	Mayor o igual que >=	Distinto !=
a<5	a>5	a==5	a<=5	a>=5	a!=5

Ejemplo: Leer una nota por teclado e indicar en pantalla si está aprobado o suspenso.

Pseudocódigo	Diagrama de Flujo	Lenguaje C
<p>Algoritmo Aprobado;                      Leer nota;                      si nota&gt;5 hacer                          Escribir “Aprobado”;                      sino                          Escribir “Suspenso”;                      Fin</p>	 <pre>                     graph TD                         inicio([inicio]) --&gt; Nota[/Nota?/]                         Nota --&gt; Cond{NOTA&gt;5}                         Cond -- verdadero --&gt; Aprobado{{APROBADO}}                         Cond -- falso --&gt; Suspenso{{SUSPENDO}}                         Aprobado --&gt; Merge(( ))                         Suspenso --&gt; Merge                         Merge --&gt; fin([fin])                     </pre>	<pre>#include &lt;stdio.h&gt; int main(){ int nota; printf("Nota ?"); scanf("%i",&amp;nota); if (nota==5){     printf("Sacaste 5"); } else {     printf("No sacas 5"); } }</pre>
<div style="background-color: black; color: white; padding: 10px; display: inline-block;">                 Nota? 7                  No sacas 5             </div>		<p style="text-align: center;"><b>Lenguaje Python</b></p> <pre>nota=input("Nota? ") if (nota==5):     print "Sacaste 5" else:     print "No sacas 5"</pre>

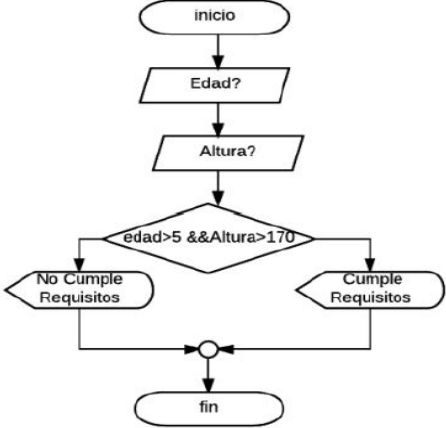
Ejercicio propuesto 2: Diseñar un algoritmo que pida por teclado tres números; si el primero es negativo, debe imprimir el producto de los tres y si no lo es, imprimirá la suma.

## Programas con instrucciones condicionales compuestas

Una condición puede estar formada por varias partes y para ello tenemos dos operadores:

Y	&& en C and en Python	Ejemplo (a>5 && a<7) Esta condición es Verdadera en el caso que todas las condiciones sean verdaderas. En cualquier otro caso es falsa.
O	en C or en Python	Ejemplo (a==5    a==7) Esta condición solo es Falsa cuando todas las condiciones que la forman son falsas. En cualquier otro caso es verdadera.

Ejemplo de algoritmo que indique si se cumplen las condiciones para presentarse a una oposición de Policía Local sabiendo que hay que ser mayor de 18 años y medir más de 170 cm.

Pseudocódigo	Diagrama de Flujo	Lenguaje C
<p>Algoritmo Requisitos;            Leer edad ;            Leer Altura;            si edad&gt;5 y altura&gt;170            hacer                Escribir “Cumple Requisitos”            sino                Escribir “No Cumple Requisitos”;            Fin</p>	 <pre> graph TD     Inicio([inicio]) --&gt; Edad[/Edad?/]     Edad --&gt; Altura[/Altura?/]     Altura --&gt; Condicion{edad&gt;5 &amp;&amp;Altura&gt;170}     Condicion -- No Cumple Requisitos --&gt; Merge(( ))     Condicion -- Cumple Requisitos --&gt; Merge     Merge --&gt; Fin([fin])           </pre>	<pre> #include &lt;stdio.h&gt; int main(){ int edad; int altura; printf("Edad ?"); scanf("%i",&amp;edad); printf("Altura ?"); scanf("%i%",&amp;altura); if (edad&gt;18 &amp;&amp;altura&gt;170) { printf("Cumple"); } else { printf("No Cumple"); } }           </pre>
<b>Código de Python</b>		<b>Lenguaje Python</b>
<pre> edad=input("Dime la edad: ") altura=input("Dime la altura: ") if (edad&gt;18 and altura&gt;170): print "Cumple" else: print "No Cumple"           </pre>		<p>Mucho cuidado con sangrar las instrucciones que tiene que realizar cada condición.</p>

**Ejercicio Propuesto 3:**

- a) Diseñar un algoritmo que pida por teclado tres números; si el primero es negativo y el segundo es positivo, debe imprimir el producto de los tres y si no lo es, imprimirá la suma.
- b) Diseñar un algoritmo que lea 3 números y muestre el mayor por pantalla. (Haciendo uso de condiciones compuestas



# Programas con instrucciones condicionales anidadas

El anidamiento de instrucciones condicionales consiste en introducir una instrucción si dentro de otro si . Para ver un ejemplo vamos a resolver un algoritmo que lea dos números y nos diga cual de ellos es mayor o bien si son iguales.

Pseudocódigo	Diagrama de Flujo	Lenguaje C
<p>Algoritmo MayorMenor;                      Leer a;                      Leer b;                      si a=b hacer                          Escribir "Son iguales";                      sino                          si a&gt;b hacer                              Escribir "a mayor que b";                          sino                              Escribir "b mayor que a";                      Fin</p>		<pre>#include &lt;stdio.h&gt; int main(){ int a; int b; printf("Introduce numero "); scanf("%i",&amp;a); printf("Introduce numero "); scanf("%i",&amp;b); if (a==b) { printf("Son iguales"); } else { if(a&gt;b){ printf("a mayor que b"); } else{ printf("b mayor que a"); } } }</pre>

Código de Python	Lenguaje Python
<pre>a=input("Introduce numero a: ") b=input("Introduce numero b: ") if (a==b): print("Son iguales") elif (a&gt;b): print("a mayor que b") else: print("b mayor que a")</pre>	<p>Mucho cuidado con sangrar las instrucciones que tiene que realizar cada condición.</p>

**Ejercicio propuesto 4:**

a) Programa que lea el color de un semáforo e indique si está en rojo -->"para" si está en amarillo -->"Precaución", y si está en verde -->"Pasar" (Usando condicionales anidados).

b) Diseñar un algoritmo que lea una nota numérica e indique en pantalla si tiene Insuficiente, Suficiente, Bien, Notable, Sobresaliente.

## Instrucción Switch.

Esta sentencia permite seleccionar las acciones a realizar de acuerdo al valor que tome una variable.

Instrucción	Ejemplo; Escriba un programa en Lenguaje C que solicite el ingreso de un número del 1 al 3 y que muestre por pantalla el número seleccionado en letras.
<pre>switch(variable) {</pre>	<pre>#include &lt;stdio.h&gt; void main()</pre>

```

case valor_1:
    //acciones
    break;
case valor_2:
    //acciones
    break;
.
.
.
case valor_n:
    //acciones
    break;
defalut:
    //acciones
    break;
}

```

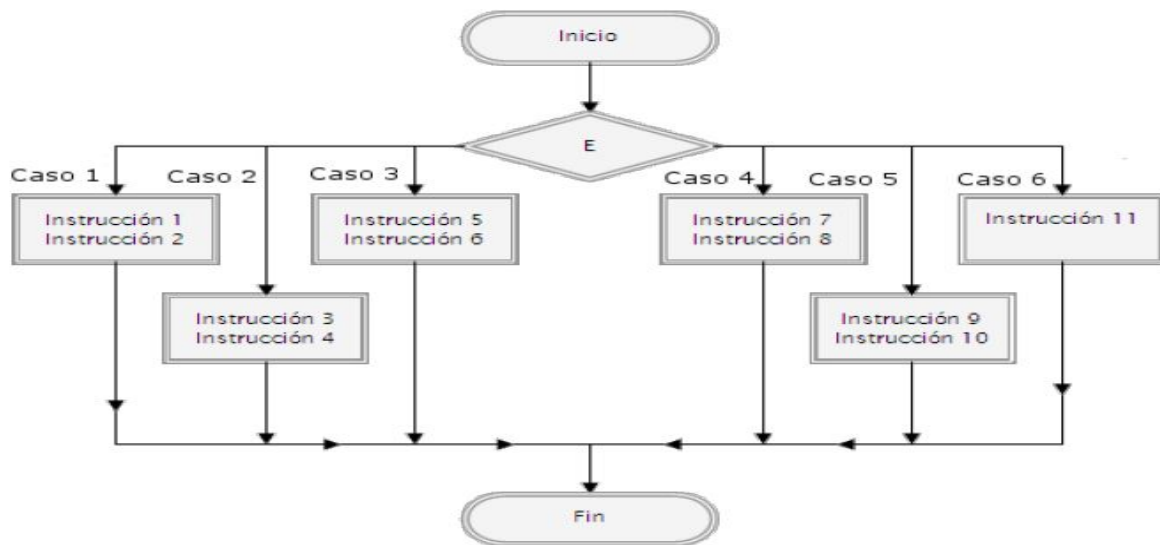
Si el valor que toma la variable no coincide con los valores especificados en los casos, se realizarán las acciones definidas en la etiqueta default.

```

{
int opcion;
printf("Ingrese un numero entero: ");
scanf("%d",&opcion);
switch(opcion)
{
case 1:
    printf("Ha ingresado el numero uno\n");
    break;
case 2:
    printf("Ha ingresado el numero dos\n");
    break;
case 3:
    printf("Ha ingresado el numero tres\n");
    break;
defalut:
    printf("Ha ingresado un numero no valido\n");
    break;
}
}

```

**Diagrama de flujo:**



**Versión Python**

No existe esta instrucción en lenguaje Python se implementa como un if anidado.

**Ejercicio propuesto 5:**

- a) Haciendo uso de la instrucción switch lee una nota numérica y muestra en pantalla la nota en formato texto.
- b) Haz un pequeño diccionario en el que se le inserte una palabra en castellano y nos la traduzca al inglés.

En Python intenta hacerlo con if anidados.

## Instrucciones Repetitivas:

Estas instrucciones realizan un conjunto de instrucciones mientras se cumpla una condición. En estas instrucciones debemos preguntarnos:

¿Cual es la condición que controla el bucle?

¿Que acción dentro del bucle modifica la condición y evitamos caer en un bucle infinito?

Son de mucha utilidad las variables auxiliares como:

**Contadores.** Variables que almacenan el número de veces que se realiza una acción. Ejemplo:

Declaración	inicialización	Contabilización o incremento
int contador;	contador=0;	contador=contador+1;

### Versión Python.

```
contador=0
contador=contador+1
```

**Acumuladores.** Variables que acumulan el valor de operaciones sucesivas. Ejemplo:

Declaración	inicialización	Acumulación
int suma; int nuevodato;	Suma=0; leer nuevodato;	Mientras que suma<20 hacer Leer nuevodato; suma=suma+nuevodato;

### Versión Python.

```
suma=0
nuevodato=input()
while suma<20:
    suma=suma+nuevodato
    nuevodato=input()
print(" El valor de la suma es", suma)
```

**Entradas controladas por el usuario. El usuario indica si quiere continuar o no.**

Declaración	inicialización	Acumulación
char seguir;	Seguir='s';	Mientras que seguir != 'n' hacer escribir("Hola si quieres sigo"); leer seguir;

### Versión Python.

```
seguir="s"
while seguir!="n":
    print("Hola si quieres sigo")
    seguir=raw_input("Sigo ? ")
```

NOTA: raw\_input() lee cadenas de caracteres o caracteres solo desde teclado. Mientras input() lee números.

# INSTRUCCIÓN MIENTRAS ... HACER

Pseudocódigo	Diagrama de flujo	Lenguaje C
<b>Mientras [Condición] Hacer</b> Instrucción 1 Instrucción 2 . . . Instrucción n <b>Repetir</b>		While (condición) { instrucción 1; .... instrucción n; }

La entrada al bucle queda controlada mediante la evaluación de una condición (verdadera o falsa). En general, el bucle se repetirá varias veces hasta que se incumpla la condición de entrada, para continuar el flujo del programa por la siguiente instrucción después de Repetir. Sin embargo, podría darse el caso particular de que en la primera evaluación no se cumpla la condición, con lo cual el bloque de instrucciones del bucle no llegaría a ejecutarse siquiera una vez. Si tras la primera pasada la condición cambia y se convierte en falsa, el bloque se ejecuta una única vez.

Ejemplo: Realizar una lista de los 10 primeros números pares.

Código	Explicación																																	
<pre>#include &lt;stdio.h&gt; int main() { int i=0;//variable para el control del bucle int par=0; //variable con los números pares while (i&lt;10){ printf("%i - ",par); par=par+2; i=i+1; } getc(); return 0; }i</pre>	<table border="1"> <thead> <tr> <th>i</th> <th>Par</th> <th>Acción</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>escribir en pantalla 0</td></tr> <tr><td>1</td><td>2</td><td>escribir en pantalla 2</td></tr> <tr><td>2</td><td>4</td><td>escribir en pantalla 4</td></tr> <tr><td>3</td><td>6</td><td>escribir en pantalla 6</td></tr> <tr><td>4</td><td>8</td><td>escribir en pantalla 8</td></tr> <tr><td>5</td><td>10</td><td>escribir en pantalla 10</td></tr> <tr><td>6</td><td>12</td><td>escribir en pantalla 12</td></tr> <tr><td>7</td><td>14</td><td>escribir en pantalla 14</td></tr> <tr><td>8</td><td>16</td><td>escribir en pantalla 16</td></tr> <tr><td>9</td><td>18</td><td>escribir en pantalla 18</td></tr> </tbody> </table>	i	Par	Acción	0	0	escribir en pantalla 0	1	2	escribir en pantalla 2	2	4	escribir en pantalla 4	3	6	escribir en pantalla 6	4	8	escribir en pantalla 8	5	10	escribir en pantalla 10	6	12	escribir en pantalla 12	7	14	escribir en pantalla 14	8	16	escribir en pantalla 16	9	18	escribir en pantalla 18
i	Par	Acción																																
0	0	escribir en pantalla 0																																
1	2	escribir en pantalla 2																																
2	4	escribir en pantalla 4																																
3	6	escribir en pantalla 6																																
4	8	escribir en pantalla 8																																
5	10	escribir en pantalla 10																																
6	12	escribir en pantalla 12																																
7	14	escribir en pantalla 14																																
8	16	escribir en pantalla 16																																
9	18	escribir en pantalla 18																																

### Versión Python.

```
i=0
par=0
while (i<10):
    print par,"-“
    par=par+2
    i=i+1
```

```
0 -
2 -
4 -
6 -
8 -
10 -
12 -
14 -
16 -
18 -
```

**Ejercicio propuesto 6:**

- Realiza un algoritmo que muestre en pantalla una lista de los 100 primeros números naturales.
- Realiza un algoritmo que sume una diez números naturales y muestre su suma en pantalla.
- Realiza un algoritmo que esté leyendo números hasta que se introduzca un número negativo.

## Sentencia Para o for.

El bucle for es una variante del bucle while y, al igual que éste, puede iterar cero o más veces. Sin embargo, el bucle for sólo se suele usar cuando se conoce el número exacto de veces que tiene que iterar el bucle.

En lenguaje C, para escribir una instrucción for (repetitiva para) se utiliza la sintaxis:

Pseudocódigo	Diagrama de flujo	En C
<pre>Para (Iniciar;condicion;incremento){ instrucciones; }</pre>		<pre>for ( &lt;expresión_1&gt; ; &lt;expresión_2&gt; ; &lt;expresión_3&gt; ) { &lt;bloque_de_instrucciones &gt; }</pre>

Por otro lado, el bucle for es ideal usarlo cuando, de antemano, ya se sabe el número de veces (iteraciones) que tiene que ejecutarse un determinado bloque de instrucciones.

Ejemplo: Por tanto, dicho problema, se puede resolver con una instrucción repetitiva for de la siguiente forma:

Ejemplo: Escribir una lista del 1 al 10	Elementos
<pre>#include &lt;stdio.h&gt; int main() {     int contador;      for ( contador=1 ; contador&lt;=10 ; contador++ )         printf( "%d ", contador );     return 0; }</pre>	<p>la inicialización de la variable contador, la condición de salida del bucle y el incremento de la variable contador</p>

### Versión Python.

```
for i in range(10):
    print(i+1)
```

Nota: ponemos i+1 pues en el ciclo for empieza contando por 0.

### Ejercicio propuesto 7: Con la sentencia for

- Realiza un algoritmo que muestre en pantalla una lista de los 100 primeros números naturales.
- Realiza un algoritmo que sume una diez números naturales y muestre su suma en pantalla.