

# PROGRAMACIÓN I - UNIDAD 4: SUBPROCESOS

Control de versiones:

Versión	Descripción cambios	Fecha
1.0	Inicial	

Contenido

Subprocesos .....3

    Ejemplo 1:.....3

    Ejemplo 2:.....3

Ventajas de los subprocesos .....4

Parámetros.....4

    Subprocesos con parámetros de entrada .....4

        Ejemplo 3:.....4

    Subprocesos con parámetros de salida .....5

        Ejemplo 4:.....5

Parámetros por valor .....6

    Ejemplo 5.....6

    Ejemplo 6.....6

Parámetros por referencia .....7

    Ejemplo 7:.....7

# Subprocesos

En muchos casos, nos encontraremos con tareas que tenemos que repetir varias veces en distintos puntos de nuestro programa. Si tecleamos varias veces el mismo fragmento de programa no sólo tardaremos más en escribir: además el programa final resultará menos legible, será más también será más fácil que cometamos algún error alguna de las veces que volvemos a teclear el fragmento repetitivo, o que decidamos hacer una modificación y olvidemos hacerla en alguno de los fragmentos. Por eso, conviene evitar que nuestro programa contenga código repetitivo. Una de las formas de evitarlo es usar "subprocesos" también llamadas "subrutinas", una posibilidad que la mayoría de lenguajes de programación permiten, y que en ocasiones recibe el nombre de "procedimientos" o de "funciones".

## Ejemplo 1:

Vamos a realizar un programa que muestre un Hola Mundo en varios idiomas, y que al final de cada idioma muestre una línea divisoria

```

Algoritmo ProgramaPrincipal
  Definir i como Entero

  Escribir "Hola mundo"

  //Escribo linea
  Para i<-1 Hasta 20 Con Paso 1 Hacer
  ..... Escribir Sin Saltar "-"
  Fin Para
  Escribir ""

  Escribir "Hello world"

  //Escribo linea
  Para i<-1 Hasta 20 Con Paso 1 Hacer
  ..... Escribir Sin Saltar "-"
  Fin Para
  Escribir ""

  Escribir "Ciao mondo"

  //Escribo linea
  Para i<-1 Hasta 20 Con Paso 1 Hacer
  ..... Escribir Sin Saltar "-"
  Fin Para
  Escribir ""
FinAlgoritmo

```

```

*** Ejecución Iniciada. ***
Hola mundo
-----
Hello world
-----
Ciao mondo
-----
*** Ejecución Finalizada. ***

```

## Ejemplo 2:

En el programa escrito del ejemplo 1 vimos que cada vez que tenía que mostrar una línea tenía que repetir la misma porción de código

```

Para i<-1 Hasta 20 Con Paso 1 Hacer
  ..... Escribir Sin Saltar "-"
Fin Para
Escribir ""

```

Entonces ahora vamos a hacer uso de un subproceso que escriba una línea.

Ejemplo en pseudocódigo	Ejemplo en C
<pre>Algoritmo ProgramaPrincipal   Escribir "Hola mundo"   EscribirLinea   Escribir "Hello world"   EscribirLinea   Escribir "Ciao mondo"   EscribirLinea FinAlgoritmo  SubProceso EscribirLinea   Definir i como Entero   Para i←1 Hasta 20 Con Paso 1 Hacer     Escribir Sin Saltar "-"   Fin Para   Escribir "" FinSubProceso</pre>	<div><pre>#include&lt;stdio.h&gt;  void escribirlinea();  int main() {   printf("Hola mundo\n");   EscribirLinea();   printf("Hello world\n");   EscribirLinea();   printf("Ciao mondo\n");   EscribirLinea();   return 0; }  void EscribirLinea() {   int i;   for (i=1;i&lt;=20;i+=1) {     printf("-");   }   printf("\n"); }</pre></div> <p>Nota: En C la palabra void antes de la subrutina indica que la misma no retorna ningún valor</p>

Vemos que en éste caso, creamos un subproceso llamado “EscribirLinea” y que desde el programa principal va a ser llamado cada vez que queramos mostrar una línea.

A simple vista vemos que no existe código redundante y que el programa es más legible.

## Ventajas de los subprocesos

- Las instrucciones dentro de ellos se pueden ejecutar varias veces.
- Facilita la división de tareas.
- Los subprocesos se pueden probar individualmente.
- Con subrutinas apropiadamente diseñadas, es posible ignorar cómo se realiza una tarea, sabiendo qué es lo que hacen.

## Parámetros

En el ejemplo anterior utilizamos un subproceso que no utilizaba parámetros. Pero, ¿Qué son los parámetros y para qué se usan?

Los parámetros son datos que o bien se envían a un subproceso (parámetros de entrada) o es un dato que el subproceso devuelve (parámetro de salida).

### Subprocesos con parámetros de entrada

Ejemplo 3:

Modifiquemos el Ejemplo 2, ahora vamos a construir un subproceso llamado “EscribirSubrayado” que reciba cómo parámetro de entrada la frase que el usuario desea escribir subrayada.

Ejemplo en pseudocódigo	Ejemplo en C
<pre>Algoritmo ProgramaPrincipal     EscribirSubrayado("Hola mundo")     EscribirSubrayado("Hello world")     EscribirSubrayado("Ciao mondo") FinAlgoritmo  SubProceso EscribirSubrayado(frase)     Definir i como Entero     Escribir frase     Para i&lt;-1 Hasta 20 Con Paso 1 Hacer         Escribir Sin Saltar "-"     Fin Para     Escribir "" FinSubProceso</pre>	<pre>#include&lt;stdio.h&gt;  void EscribirSubrayado(char*);  int main() {      EscribirSubrayado("Hola mundo");     EscribirSubrayado("Hello world");     EscribirSubrayado("Ciao mondo");     return 0; }  void EscribirSubrayado(char* frase) {     int i;     printf("%s\n",frase);     for (i=1;i&lt;=20;i+=1) {         printf("-");     }     printf("\n"); }</pre>

Subprocesos con parámetros de salida

Existen algunos subprocesos devuelven un valor en la llamada, en muchos lenguajes a éstos se los llama “funciones” y a los subprocesos que no devuelven valores se los suele llamar “procedimientos”.

Ejemplo 4:

Vamos ahora a realizar una función que reciba como parámetro de entrada una nota y determine si el valor es válido o no devolviendo Verdadero o Falso.

Para esto se crea una función de nombre “ValidarNota” que recibe un valor real y devuelve un valor lógico.

Ejemplo en pseudocódigo	Ejemplo en C
<pre>Algoritmo ProgramaPrincipal     Definir valido Como Logico     Definir nota Como Real     Escribir "Ingrese una nota:"     Repetir         Leer nota         valido= ValidarNota(nota)     Mientras Que !valido FinAlgoritmo  Funcion valido &lt;- ValidarNota(nota)     Definir valido Como Logico     Si nota &gt;0 y nota&lt;11         valido=Verdadero     SiNo         valido=Falso         Escribir "Nota inválida. Ingrese nuevamente"     FinSi FinFuncion</pre> <div>*** Ejecución Iniciada. *** Ingrese una nota: &gt; 11 Nota inválida. Ingrese nuevamente &gt; 1 *** Ejecución Finalizada. ***</div>	<pre>#include&lt;stdio.h&gt; #include&lt;stdbool.h&gt;  bool ValidarNota(float nota);  int main() {     int nota;     bool valido;     printf("Ingrese una nota:\n");     do {         scanf("%i",&amp;nota);         valido = ValidarNota(nota);     } while (!valido);     return 0; }  bool ValidarNota(float nota) {     bool valido;     if (nota&gt;0 &amp;&amp; nota&lt;11) {         valido = true;     } else {         valido = false;         printf("Nota inválida. Ingrese nuevamente\n")     }     return valido; }</pre>



# Parámetros por valor

Cuando se utilizan parámetros por valor, el subproceso o función opera con copias de las variables que se envían en la llamada. Como son copias, si se modifica alguna variable en el subproceso, no se refleja en la variable de la llamada.

En PSeInt debe usarse la expresión “por valor” para indicar éste tipo de parámetros. Si no se indica, por defecto se supone que las variables se pasan por valor (excepto para los arreglos que se consideran por defecto que se pasan por referencia).

Entonces si usamos un parámetro por valor, la variable con la que trabajará el subproceso

## Ejemplo 5

En el siguiente ejemplo vamos a mostrar un caso de un subproceso que pasa parámetros por valor.

Ejemplo en pseudocódigo	Ejemplo en C
<pre>1  Algoritmo ProgramaPrincipal 2      Definir nro Como Entero 3      nro=1 4      Escribir "Nro al principio, en el programa ppal tiene el valor: ",nro 5      sub(nro) 6      Escribir "Nro en el programa ppal luego de invocar subproceso sub: ",nro 7 8  FinAlgoritmo 9 10 SubProceso sub( nro Por Valor ) 11     nro=2 12     Escribir "Nro en el subproceso por_valor cambia a: ", nro 13 FinSubProceso</pre> 	<pre>#include&lt;stdio.h&gt;  void sub(float nro);  int main() {     int nro;     nro = 1;     printf("Nro al principio, en el programa ppal tiene el valor: %i\n",nro);     sub(nro);     printf("Nro en el programa ppal luego de invocar subproceso sub: %i\n",nro);     return 0; }  void sub(float nro) {     nro = 2;     printf("Nro en el subproceso por_valor cambia a: %f\n",nro); }</pre>

Nótese que hubiese sido lo mismo si omitíamos el “Por Valor”, ya que por defecto en PSeInt si no se especifica el tipo, se asume que es un parámetro por valor (excepto arreglos).

```
SubProceso sub( nro )
    nro=2
    Escribir "Nro en el subproceso por_valor cambia a: ", nro
FinSubProceso
```

Tener en cuenta también que en éstos casos no es necesario que el nombre de la variable que se envía en la llamada coincida con la copia definida en el subproceso.

## Ejemplo 6

En el siguiente ejemplo se envía un parámetro por valor y el nombre de las variables no coinciden. Sin embargo, el programa funciona correctamente, ya que no son las mismas, la variable “n” solo toma el valor que tiene la variable “nota” en el programa principal, ya que solo es una copia.

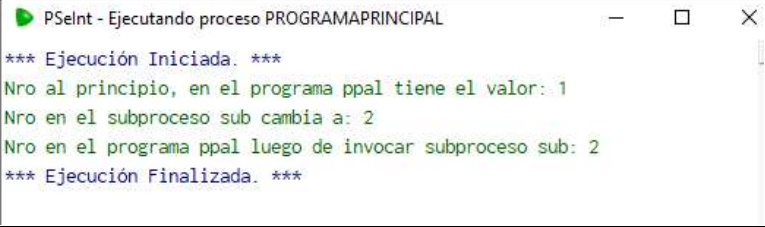
```
Algoritmo ProgramaPrincipal
    Definir valido Como Logico
    Definir nota Como Real
    Escribir "Ingrese una nota:"
    Repetir
        Leer nota
        valido= ValidarNota(nota)
    Mientras Que !valido
FinAlgoritmo

Funcion rtdo <- ValidarNota(n)
    Definir rtdo Como Logico
    Si n >0 y n<11
        rtdo=Verdadero
    SiNo
        rtdo=Falso
        Escribir "Nota inválida. Ingrese nuevamente"
    FinSi
FinFuncion
```

# Parámetros por referencia

En éste caso, el subproceso no manipula una copia de la variable enviada, sino que utiliza la misma, y si la variable se modifica en el subproceso, entonces se modifica también desde dónde se invocó la llamada. Cuando usemos parámetros por referencia no podemos cambiar el nombre de la variable recibida como parámetro, debe ser la misma que en la llamada.

Ejemplo 7:

Ejemplo en pseudocódigo	Ejemplo en C
<div><pre>Algoritmo ProgramaPrincipal   Definir nro Como Entero   nro=1   Escribir "Nro al principio, en el programa ppal tiene el valor: ",nro   sub(nro)   Escribir "Nro en el programa ppal luego de invocar subproceso sub: ",nro FinAlgoritmo  SubProceso sub( nro Por Referencia )   nro=2   Escribir "Nro en el subproceso sub cambia a: ", nro FinSubProceso</pre></div> <div></div>	<pre>#include&lt;stdio.h&gt;  void sub(float *nro);  int main() {   int nro;   nro = 1;   printf("Nro al principio, en el programa ppal tiene el valor: %i\n",nro);   sub(nro);   printf("Nro en el programa ppal luego de invocar subproceso sub: %i\n",nro);   return 0; }  void sub(float *nro) {   (*nro) = 2;   printf("Nro en el subproceso sub cambia a: %f\n",(*nro)); }</pre>