

Programación con PSEINT

1.- Introducción

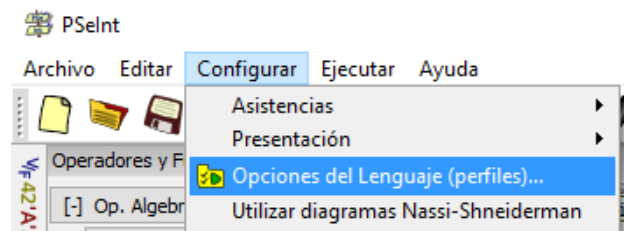
PSINT (Pseudocode interpreter) es un programa que utiliza pseudocódigo, un lenguaje de programación ficticio cuya principal misión es que el programador pueda centrarse en los aspectos lógicos de la programación, dejando el apartado técnico para cuando se vea la sintaxis de un lenguaje de programación verdadero. Además, es un entorno de desarrollo de programación o **IDE (Integrated Development Environment)**. Los programas pueden escribirse en un simple editor de textos. Sin embargo, un IDE ayuda a la programación: coloreando las palabras clave, mostrando las instrucciones clave cuando las tenemos a medio escribir, resaltando los errores, etc...

Podemos descargar PSEINT en la siguiente página: <http://pseint.sourceforge.net/>

En PSeInt podemos elegir entre distintos perfiles de Lenguaje:

El **perfil Estricto**, hace que PSEINT deba seguir las normas de la mayoría de los lenguajes de bajo nivel.

El **perfil Flexible** es menos riguroso y no da error si nos saltamos algunas de esas reglas.



En la asignatura funcionaremos en modo **estricto** porque, si bien es más complicado, facilitará el aprendizaje de cualquier otro lenguaje real. Aunque PSeInt subraya con rojo los errores de sintaxis, también permite verificarlos mediante **Ejecutar/Verificar Sintaxis**

Algunas de las normas comunes a otros lenguajes a seguir en el perfil estricto son:

- Al igual que muchos lenguajes como C o Javascript, las sentencias siempre finalizan en punto y coma.
- Los identificadores, o nombres de variables, deben constar sólo de letras, números y/o guión_bajo (_), comenzando siempre con una letra. No puede contener símbolos (@, #, ▼, [,], =, etc), ni eñes (ñ ó Ñ) y no puede tener espacios en blancos. .
- Los identificadores, o nombres de variables, no pueden coincidir con las palabras reservadas del lenguaje (instrucciones).
- Se pueden crear variables del mismo tipo en una sola línea separadas por comas
Definir n1,n2,suma Como Entero;
- Se pueden introducir comentarios después del ";" de una línea, o en una línea separadas, mediante el uso de la doble barra (//). Todo lo que precede a //, hasta el fin de la línea, no será tomado en cuenta al interpretar el algoritmo.
- No puede haber instrucciones fuera del proceso (antes de Algoritmo, o después de FindeAlgoritmo), aunque sí comentarios.
- Las estructuras no secuenciales pueden anidarse. Es decir, pueden contener otras adentro, pero la estructura contenida debe comenzar y finalizar dentro de la contenedora.
- En las constantes numéricas, el punto (.) es el separador decimal.

Algunas de las normas de PSEINT no comunes a otros lenguajes de programación son:

PSeInt no es **case sensitive**, por lo tanto colocar Escribir con mayúsculas y minúsculas es lo mismo y no genera errores de ningún tipo, pero por respeto a la sintaxis mostrada por los botones se debe escribir con mayúscula inicial, evitando así errores de formato.

Para ejecutar un programa puedes usar el ícono de ejecutar  o pulsar F9:

Cualquier programa en PSEINT tiene que tener al menos un Algoritmo (versiones antiguas utilizan la palabra **Proceso**).

```

1  Algoritmo sin_titulo
2
3  FinAlgoritmo
4

```

Conviene asignar un nombre al algoritmo, sustituyendo **sin_titulo** por un nombre que tenga relación con el programa. El nombre del pseudocódigo en ninguna sintaxis puede tener espacios y en sintaxis estricta tampoco caracteres acentuados.

No confundir el nombre del algoritmo con el del archivo que guardemos. Pueden ser distintos, aunque recomendamos que coincidan hacia tenerlo mejor localizado.

Diagramas de flujo

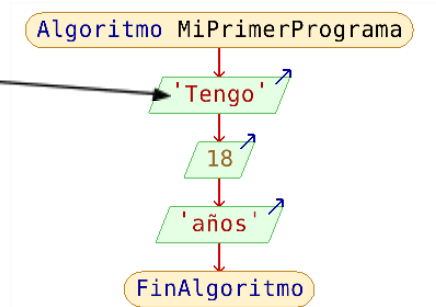
Permiten representar el funcionamiento de un algoritmo mediante símbolos asociados a las diferentes instrucciones. Los emplearemos al inicio para interpretar los programas.

2.- Empezando a programar con PSEINT

A continuación, deberás ir escribiendo y probando todos los siguientes programas en PSEINT. **De nada vale escribirlo sin entender lo que estás haciendo, ya que no sabrás realizar las prácticas que se manden y fallarás el examen.** Pregunta las dudas al profesor.

Instrucción “Escribir”

```
1  Algoritmo MiPrimerPrograma
2      Escribir "Tengo";
3      Escribir 18;
4      Escribir "años";
5  FinAlgoritmo
```



La palabra reservada **Escribir** escribe en la pantalla lo que lleva escrito a continuación.

Si quisiéramos que todo apareciese en la misma línea deberíamos utilizar la instrucción reservada **Sin Saltar**. Modifica el programa para que quede así:

```
1  Algoritmo MiPrimerPrograma
2      Escribir "Tengo " Sin Saltar;
3      Escribir 18 Sin Saltar;
4      Escribir " años";
5  FinAlgoritmo
```

También es válido y más visual concatenar elementos mediante el separador coma.

Los textos van entre comillas, los números no. Podemos escribir un número entre comillas y será mostrado sin error, sin embargo, el lenguaje lo reconocerá como un texto y no como un número.

Para comprobar este hecho, prueba a escribir el siguiente código. En la línea dos, “2*6” es considerado texto y no realiza ninguna operación. matemática, mientras que en la línea cuatro sí. Recordar que en programación, el signo * sirve para multiplicar.

```
1  Algoritmo MiPrimerPrograma
2      Escribir "Tengo ",18," años";
3  FinAlgoritmo
```

Escribe un programa que nos muestre por pantalla lo siguiente, incluyendo las líneas en blanco:

```
*****  MI PRIMER PROGRAMA  *****
```

```
-----  HOLA MUNDO  ----->
```

```
*****
*****  GAME OVER  *****
*****
```

```
1  Algoritmo MiSegundoPrograma
2      Escribir "2*6" Sin Saltar;
3      Escribir "=" Sin Saltar;
4      Escribir 2*6;
5  FinAlgoritmo
```

3.- Variables

Variable es un valor que puede variar en la ejecución de un programa, distinguiremos el nombre de la variable y el valor que toma. PSEINT utiliza un número muy limitado de tipos de datos para crear variable:

Tipo de dato	Explicación
Entero	Nº sin decimales
Real	Nº con decimales
Caracter o cadena	Cadenas de caracteres es decir cualquier letra, palabra frase u oración.
Logico	El resultado de una expresión lógica, sólo tendrá dos únicos valores VERADERO o FALSO

En el modo estricto debemos declarar las variables al principio del algoritmo o proceso. Esto se hace así:

Definir Nombre_variable Como Tipo_de_dato;

Algunos ejemplos de declaración de variables:

```
Definir AlturaMontaña Como Entero;
Definir NotaExamen Como Real;
Definir Pais Como Caracter;
Definir Casado Como Logico;
```

Una vez definida la variable, para **asignarle un valor utilizamos los signos <-**

En el siguiente ejemplo asignamos el valor 25 a la variable edad. Por ejemplo:

```
AlturaMontaña<-3718
NotaExamen<-6.75
Pais<-"España"
Casado<-Verdadero
```

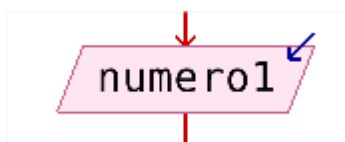
Algunas de las normas sobre utilización de variables a seguir en el perfil estricto, comunes a otros lenguajes, son:

- Los identificadores, o nombres de variables, deben constar sólo de letras, números y/o guión bajo (_), comenzando siempre con una letra. No puede contener símbolos (@, #, ▼, [,], =, etc.), ni eñes (ñ ó Ñ) y no puede tener espacios en blancos. .
- Los identificadores, o nombres de variables, no pueden coincidir con las palabras reservadas del lenguaje .PSEINT colorea de azul Las palabras reservadas.
- Se pueden crear variables del mismo tipo en una sola línea separadas por comas
Definir n1,n2,suma Como Entero;

Ejemplo de uso de variables

Este programa es poco útil, ya que siempre suma los mismos valores, el dos y el tres. Lo más lógico es que el propio programa nos pregunte qué números queremos sumar.

Esto se consigue con la **instrucción Leer**. Cuando ejecutamos un programa, esta instrucción espera a que escribamos con el teclado y, cuando pulsemos la tecla **intro**, asignará lo escrito a la variable que acompaña a la instrucción.



```
1  Algoritmo SumaConVariables
2      Definir n1 Como Entero;
3      Definir n2 Como Entero;
4      n1<-2;
5      n2<-3;
6      Escribir "La suma es ",n1+n2;
7  FinAlgoritmo

1  Algoritmo SumaConVariables
2      Definir n1 Como Entero;
3      Definir n2 Como Entero;
4      Escribir "Dime un número";
5      Leer n1;
6      Escribir "Dime un número";
7      Leer n2;
8      Escribir "El resultado de la suma es ",n1+n2;
9  FinAlgoritmo
```

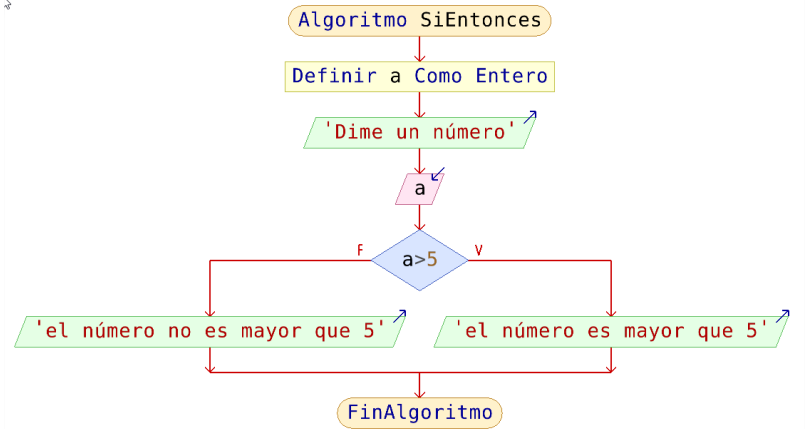
Si quisiéramos guardar el resultado para usarlo más adelante deberíamos utilizar otra nueva variable, ampliar el algoritmo para que el resultado se almacene en otra variable y muestre su valor con la instrucción Escribir.

4.- Selección simple. “Si Entonces”

Esta función es básica en programación. Si se cumple la condición dada por una expresión lógica realizará las instrucciones que siguen. Si no se cumple la condición ejecutará las instrucciones que siguen a Sino hasta Fin Si.

```

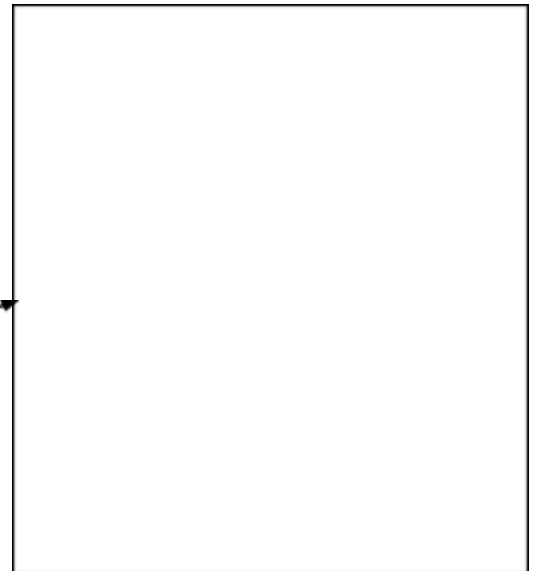
Si expresion logica Entonces
    acciones por verdadero
Sino
    acciones por falso
Fin Si
  
```



```

Algoritmo ComparacionEdades
    Definir EdadAlumno1 Como Entero;
    Definir EdadAlumno2 Como Entero;
    Escribir "Dime la edad del Alumno1";
    Leer EdadAlumno1;
    Escribir "Dime la edad del Alumno2";
    Leer EdadAlumno2;
    Si EdadAlumno1=EdadAlumno2 Entonces
        Escribir "Ambos alumnos tienen la misma edad";
    FinSi
    Si EdadAlumno1>EdadAlumno2 Entonces
        Escribir "El alumno1 es mayor que el alumno2";
    FinSi
    Si EdadAlumno1<EdadAlumno2 Entonces
        Escribir "El alumno2 es mayor que el alumno1";
    FinSi
FinAlgoritmo
  
```

Realiza el diagrama de flujo del algoritmo.

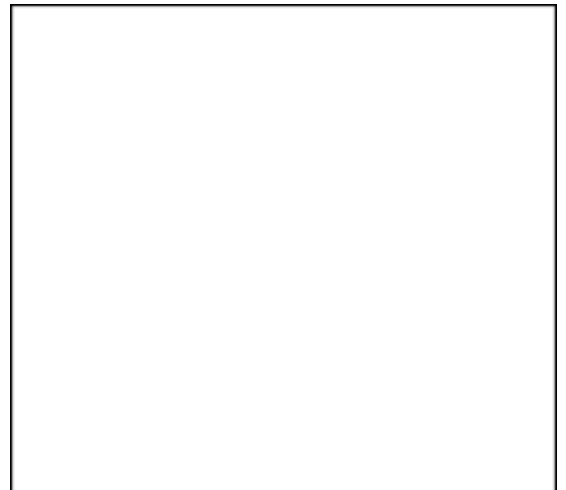


Cuando queremos comprobaciones más complejas, podemos utilizar los operadores lógicos Y, O, NO. Veamos el siguiente ejemplo:

```

Algoritmo Beca
    Definir EdadAlumno Como Entero;
    Escribir "Dime la edad del Alumno";
    Leer EdadAlumno;
    Si EdadAlumno>=18 Y EdadAlumno<=25 Entonces
        Escribir "El alumno SÍ recibe beca";
    Sino
        Escribir "El alumno NO recibe beca";
    FinSi
FinAlgoritmo
  
```

Realizar el diagrama de flujo



Operaciones relacionales

Permiten que el ordenador valore si se cumple o no una condición

Escribe el siguiente programa y escribe antes de ejecutar el resultado esperado VERDADERO/FALSO

```

Algoritmo OperadoresRelacionales
  Definir a como Entero;
  Definir b como Entero;
  a=6;
  b=5;
  Escribir a=b;
  Escribir a<>b
  Escribir a!=6
  Escribir a<b
  Escribir a>b
  Escribir a<=b
  Escribir a>=b
FinAlgoritmo

```

Operaciones lógicas Y, O, NO

Permiten combinar varias operaciones relacionales

Y: devuelve verdadero si las dos operaciones son verdaderas

4=4 Y 6>4 VERDADERO 4<2 Y 3=3 FALSO

O: devuelve verdadero si alguna de las operaciones es verdadera

4=2 O 1<4 VERDADERO 4>2 O 5<8 VERDADERO 4<=3 O 4>7 FALSO

NO: devuelve lo contrario de la operación

NO 4=2 VERDADERO NO 4>2 FALSO

Realiza diferentes programas para cada una de las condiciones, en todos se nos pide un número y nos escriba la palabra correcto sólo si se cumple la condición:

Ejemplo, condición el número es mayor que 5 y menor que 10.

Algoritmo OperadoresLogicas

```

Definir a como Entero;
Escribir "Dime un número";
Leer a;
Si a>5 Y a<10 Entonces
  Escribir "Correcto";
FinSi

```

FinAlgoritmo

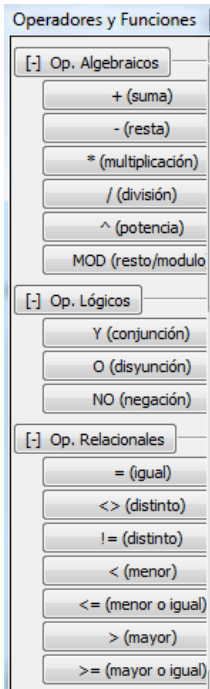
Condición a evaluar

Modifica el programa para las siguientes condiciones:

El número es menor que 10 o mayor que 20	
El número es 5, 7 o mayor que 10	
El número es mayor o igual que 10 y menor que 20	
El número es mayor que -5 y menor que 5	

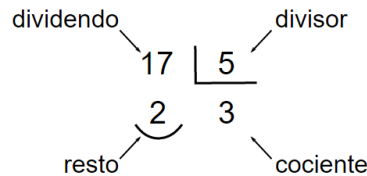
5.- Operaciones con PSEINT

PSEINT ofrece una ayuda de las operaciones matemáticas que puede utilizar en el lateral izquierdo.



Aparte de las operaciones más comunes, suma, resta, multiplicación, etc. Tenemos alguna otra:

La función **MOD** devuelve el resto de una división;



Si escribimos en PSEINT...

Ejemplo:

Escribir 17 MOD 5; //Mostrará en pantalla

Para calcular el cociente, utilizamos la función **TRUNC**. Lo que c la parte entera de un número.

Ejemplo:

Escribir TRUNC(7.25); //Mostrará en pantalla

Luego la parte entera de una operación división mostrará el cociente.

Ejemplo:

Escribir TRUNC(17/2); Mostrará en pantalla 8

La función **REDON**, redondea un número real al entero más cercano por abajo o por arriba:

Escribir REDON (3.456); mostrará 3

Escribir REDON (3.51); mostrará 4

La función **Azar(x)**, devuelve un número entero aleatorio desde 0 y x-1.

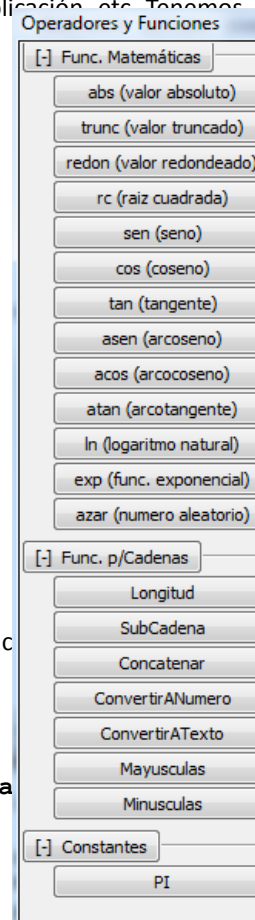
La función **Aleatorio (x,y)** devuelve un número entero aleatorio entre x e y.

1	Algoritmo Sorteo	1	Algoritmo Sorteo
2	Escribir azar(1000)	2	Escribir Aleatorio(1,10);
3	FinAlgoritmo	3	FinAlgoritmo
4		4	

Realizar un algoritmo que simule el lanzamiento de una moneda, puede salir caro o cruz. El ordenador nos pregunta qué elegimos, simula el lanzamiento de la moneda y nos dice si hemos acertado.

Realizar un algoritmo que dado el dividendo y divisor nos devuelva el cociente y el resto de la división.

FUNCIONES CON CADENAS DE CARACTERES



Permiten manipular las cadenas de caracteres.

Escribe el siguiente algoritmo para comprobar qué hacen las distintas funciones.

```

1  Algoritmo cadenas caracteres
2      Escribir Longitud("Esto es una cadena de caracteres");
3      Escribir Subcadena("murciélago",3,6);
4      Escribir Concatenar("Esto es","un murciélago");
5      Escribir Mayusculas("pepe");
6      Escribir Minusculas("BURRITO");
7      Escribir ConvertirANumero("123");
8      Escribir ConvertirATexto(123);
9  FinAlgoritmo
10

```

Longitud
SubCadena
Concatenar
ConvertirANumero
ConvertirATexto
Mayusculas
Minusculas

Pasa a mayúsculas o minúsculas una cadena

Devuelve la longitud de una cadena de caracteres

Devuelve la parte de la cadena entre las posiciones indicadas

Devuelve una cadena unión de dos.

Convierte a número una cadena (debe estar formada por números.

Convierte a texto una cadena de números.

Realizar un algoritmo que nos pida una palabra y nos la escriba en la pantalla con la primera y la última letras en mayúsculas.

Dime una palabra:

manzana

Tu palabra es Manzana

Realizar un algoritmo que genere frases de forma aleatoria, nos pregunta si queremos que invente una frase y la genera a partir de 3 sujetos, 3 verbos y 3 predicados.

Sujetos: el perro, el niño, el árbol

Verbos: crece, juega, canta

Predicados: con el sol, una canción, con un hueso

Me invento una frase?

si

el niño crece con el sol

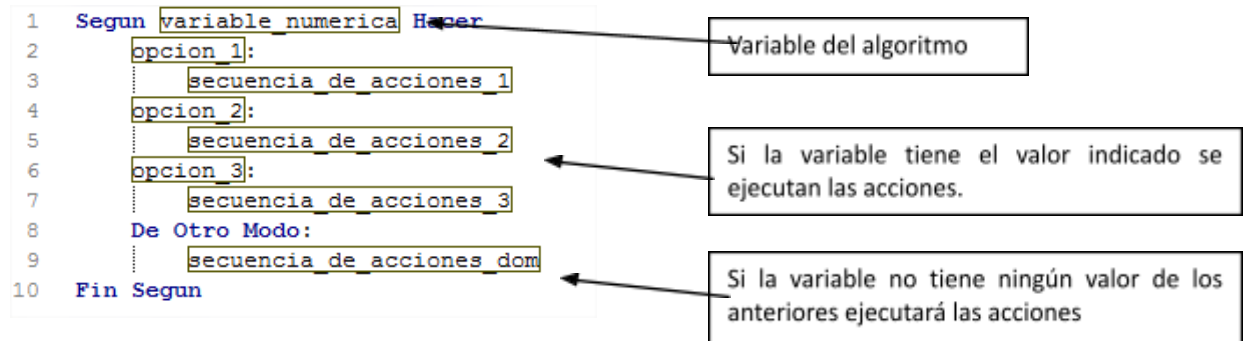
Realizar un juego de 3 preguntas en las que el ordenador nos dice si hemos acertado, la respuesta, la respuesta se puede escribir con mayúsculas o minúsculas.

Capital de España
madrid
Correcto

Capital de España
Madrid
Correcto

6.- Selección múltiple, “Segun Hacer”

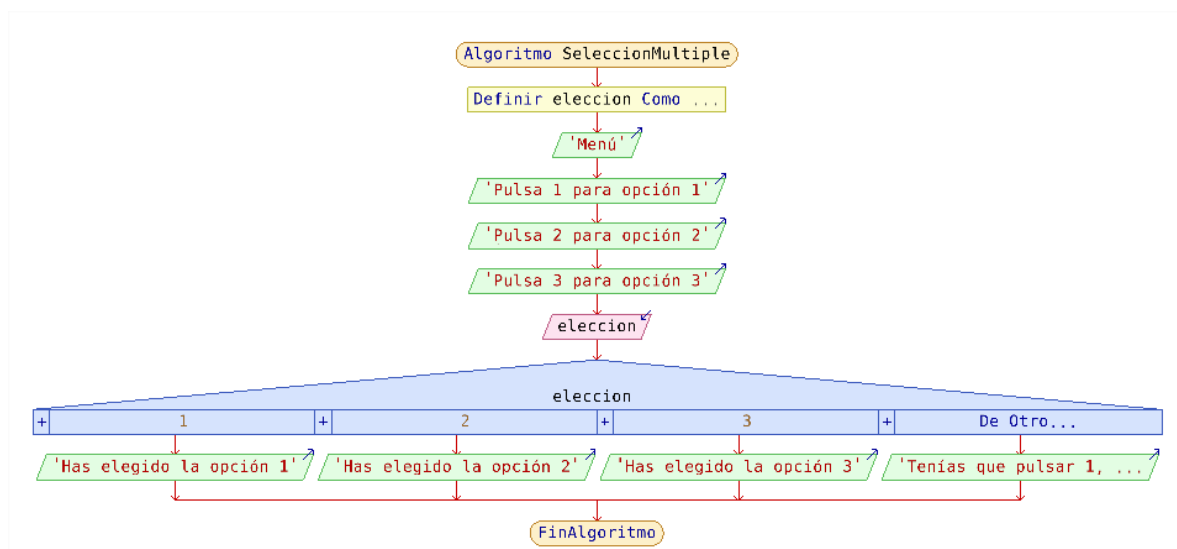
La siguiente instrucción permite que el programa realice unas instrucciones u otras en función del valor de una variable.



Algoritmo SeleccionMultiple

```

Definir eleccion Como Entero;
Escribir "Menú"
Escribir "Pulsa 1 para opción 1";
Escribir "Pulsa 2 para opción 2";
Escribir "Pulsa 3 para opción 3";
Leer eleccion
Segun eleccion Hacer
  1:
    Escribir "Has elegido la opción 1";
  2:
    Escribir "Has elegido la opción 2";
  3:
    Escribir "Has elegido la opción 3";
  De Otro Modo:
    Escribir "Tenías que pulsar 1, 2 ó 3";
Fin Segun
FinAlgoritmo
  
```



Realizar un algoritmo en el que el ordenador nos pide dos valores, luego nos pregunta qué queremos hacer

1 Sumar/2 restar/3 multiplicar/4 dividir, le introducimos la opción y el ordenador nos devuelve la operación solicitada.

7.- Iteración tipo “Para Hasta”

Los bucles “Para Hasta” se utilizan para que un trozo de algoritmo se repita el número de veces que le indiquemos, mientras una variable va cambiando de valor.

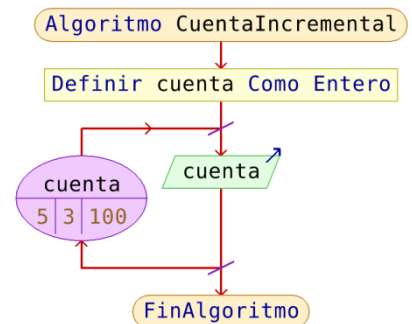
Su utilización es:

```
Para <variable> <- <inicial> Hasta <final> Con Paso <paso> Hacer
    <instrucciones>
FinPara
```

Veamos algunos ejemplos

//En el primer ejemplo se realiza una cuenta desde el 5 hasta el 100 avanzando de 3 en 3. En la primera repetición cuenta vale 5, en la segunda repetición vale 8, en la tercera 11,

```
Algoritmo CuentaIncremental
    Definir cuenta como Entero;
    Para cuenta<-5 Hasta 100 Con Paso 3 Hacer
        Escribir cuenta;
    FinPara
FinAlgoritmo
```



En el segundo ejemplo se realiza una cuenta desde el 50 hasta el 20 retrocediendo de 2 en 2

```
Algoritmo CuentaIncremental
    Definir cuenta como Entero;
    Para cuenta<-50 Hasta 20 Con Paso -2 Hacer
        Escribir cuenta;
    FinPara
FinAlgoritmo
```

Si quiero hacer un programa completo que permita todas las posibilidades, debería escribir algo como lo siguiente:

```
Algoritmo CuentaIncremental
    Definir num, intervalo, cuenta como Entero;
    Escribir "¿Hasta qué número quieres llegar?";
    Leer num;
    Escribir "¿Cuál es el intervalo entre número y número?";
    Leer intervalo;
    Para cuenta<-1 Hasta num Con Paso intervalo Hacer
        Escribir cuenta;
    FinPara
FinAlgoritmo
```

Cuando simplemente quiero que algo se repita un determinado número de veces, por ejemplo repetir Hola 10 veces:

<pre>Algoritmo Repeticion Definir i Entero; Para i<-1 Hasta 10 Con Paso 1 Hacer Escribir "Hola"; FinPara FinAlgoritmo</pre>	<pre>Algoritmo Repeticion Definir i Entero; Para i<-1 Hasta 10 Hacer Escribir "Hola"; FinPara FinAlgoritmo</pre>	Si el paso es 1 se puede omitir
--	---	--

Realizar un algoritmo que nos escriba 20 veces numeradas una frase

Realizar un algoritmo que nos devuelva la tabla de multiplicar del número 3, utilizando un bucle Para Hasta.

- 1.- Esta es la frase a copiar
- 2.- Esta es la frase a copiar
- 3.- (Así hasta 20 veces)

Realizar un algoritmo que nos pida un número del 1 al 10 y nos devuelva la tabla de multiplicar de dicho número.

Realizar un algoritmo que haga una cuenta atrás desde 10 a 0 con una espera de un segundo.

Utiliza la nueva instrucción **Esperar** Esperar 3 Segundos; El algoritmo se detendrá durante 3 segundos.

La instrucción "**Esperar**" puede utilizarse para pausar el algoritmo durante un intervalo de tiempo predefinido, indicando a continuación de la palabra clave la longitud y unidad de dicho intervalo. Las unidades válidas son **Segundos** y **Milisegundos**.

Añade al programa anterior la instrucción "Borrar Pantalla" para que cuando acabe la cuenta atrás se borre la pantalla y aparezca toda la pantalla pintada del símbolo *, utiliza la orden Para Hasta para no repetir la orden escribir

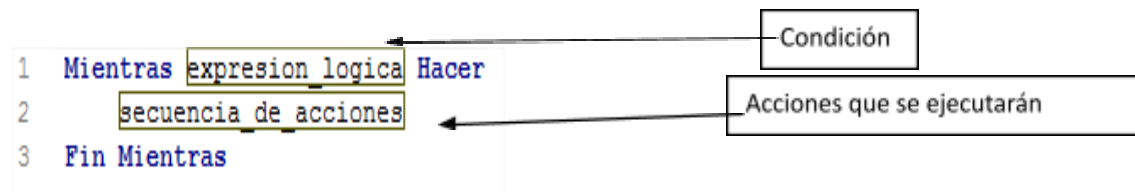
La instrucción "**Borrar Pantalla**" (o "Limpiar Pantalla") permite, como su nombre lo indica, borrar la pantalla y colocar el cursor en la esquina superior izquierda.

Borrar Pantalla;

```
*****
*****
*****
*****
*****
*****
*****
*****
```

8.- Iteración "Mientras Hacer"

Sirve para que se ejecuten unas instrucciones cuando se cumpla una condición.



Copia y comprueba el siguiente algoritmo

```

Algoritmo BucleSinSalida
Definir respuesta como Caracter;
Respuesta<-" "
    Mientras respuesta<>"la clave" Hacer
        Escribir "Escribe la clave";
        Leer respuesta;
    Fin Mientras
Escribir "Bravo! has sabido escribir ->la clave";
FinAlgoritmo
  
```

Realizar un algoritmo que no acabe nunca, el ordenador entra en un "bucle sin salida" programa que no tiene fin.

Con PSEINT se detiene un programa al pulsar combinación Ctrl+c

Realizar un algoritmo en el que el ordenador genere un número aleatorio entre 1 y 10. El ordenador nos va

```

Estoy pensando un número.
¿Qué número es?
5
Mi número es mayor, vuelve a intentarlo.
7
Mi número es menor, vuelve a intentarlo.
6
  
```

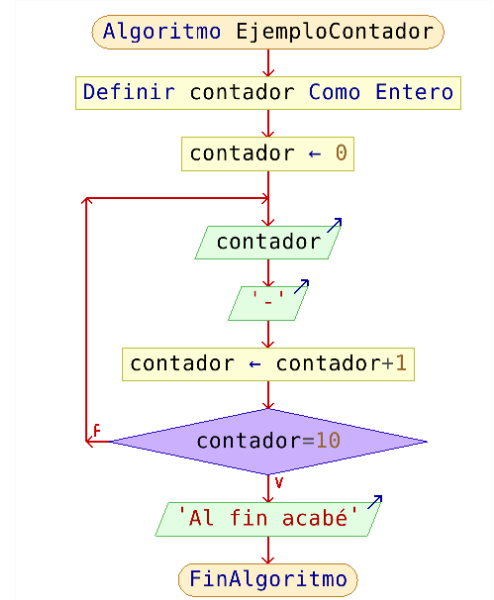
preguntando ¿Qué número es? Y responde “mi número es mayor” o “mi número es menor” hasta que acertamos, entonces responde “Has acertado” y se acaba el programa.

9.- Iteración tipo “Repetir Hasta que”

Con este tipo de estructura **se ejecutará una vez todas las acciones**, si se cumple la condición se saldrá del bloque si no se volverán a repetir hasta que se cumpla la condición.

```
1 + Repetir
2   secuencia de acciones
3   Hasta Que expresion logica
```

```
Algoritmo EjemploContador
  Definir contador como entero;
  contador<-0
  Repetir
    Escribir contador Sin Saltar;
    Escribir "-" Sin Saltar;
    contador<-contador+1
  Hasta Que contador=10;
  Escribir "Al fin acabé";
FinAlgoritmo
```



Modifica el algoritmo para que el programa escriba el valor de la variable contador al final.

Modifica el algoritmo para que nos pregunte hasta qué número quieres contar y el ordenador cuente hasta ese número.

10.- Estructuras importantes en programación

Contador

El programa anterior es un ejemplo de un “contador”, consiste en utilizar una variable dentro de un bucle de manera que cada vez que se repite la instrucción la variable se incrementa en un valor. Se puede utilizar en distintos tipos de bucles, “Para Hasta”, “Mientras hacer”, “Repetir Hasta Que”

Fíjate que en los tres ejemplos el resultado es el mismo:

<pre>Algoritmo EjemploContador Definir contador como entero; contador<-0 Repetir Escribir contador; contador<-contador+1 Hasta Que contador=10; FinAlgoritmo</pre>	<pre>Algoritmo EjemploContador Definir contador como entero; contador<-0; Mientras contador<10 Hacer Escribir contador; contador<-contador+1 Fin Mientras FinAlgoritmo</pre>
<pre>Algoritmo EjemploContador Definir contador,i como entero; contador<-0;</pre>	Básicamente lo que hacemos es <pre>x<-0 x<-x+0</pre>

<pre> Para i<-0 Hasta 9 Con Paso 1 Hacer Escribir contador; contador=contador+1 Fin Para FinAlgoritmo </pre>	La variable convierte su valor en el valor previo más 1.
---	--

Acumulador

Si en lugar de sumar siempre uno sumamos otros valores la estructura se convierte en un acumulador. El siguiente programa nos pregunta por un número y lo va sumando a la variable total hasta que su valor es mayor que 50.

Algoritmo EjemploAcumulador

```

Definir total, entrada como entero;
entrada<-0;
Mientras total<100 Hacer
    Escribir "Dime un número para sumar";
    Leer entrada;
    total=total+entrada;
    Escribir "Acumulado ", total;
Fin Mientras
Escribir "Al fin llegamos a ", total;
FinAlgoritmo

```

Con este tipo de estructura podemos ir cambiando el valor con diferentes operaciones, suma, resta, multiplicación y división.

 $x \leftarrow x * 2$
 $x \leftarrow x - 1$
 $x \leftarrow x / 2$

Realiza un programa que nos pida un número y nos dé la multiplicación de todos los números desde 1 hasta el número dado:	Realiza un programa en el que el ordenador nos haga 3 preguntas cuando se dé una respuesta correcta suma 3 puntos y si la respuesta es incorrecta resta un punto.
Dime un número. 5  El factorial es. 120	Capital de España Madrid  Correcto, llevas 3 puntos Capital de Francia Madrid  Falso, llevas 2 puntos

Bucles anidados

Mediante dos bucles “Para Hasta” anidados, podemos generar pares de valores de dos variables.

```

Algoritmo EjemploAcumulador
  Definir i,j como entero;
  Para i<-0 Hasta 5 Con Paso 1 Hacer
    Para j<-0 Hasta 4 Con Paso 1 Hacer
      Escribir "(" ,i ,"," ,j ,") ";
    Fin Para
  Fin Para
FinAlgoritmo

```

El bucle interior debe de estar dentro del bucle principal

Realiza un algoritmo que nos dé las 25 frases posibles de combinar, 5 sujetos y 5 predicados.

Sujetos: La niña, el perro, el árbol, el coche, la casa

Predicados: es verde, juega con el sol, hace sombra, tiene ruedas, crece deprisa

EJERCICIOS PSEINT

1 Realiza un programa que pida la nota de un examen y muestre la calificación: INSUFICIENTE, SUFICIENTE, BIEN, NOTABLE, SOBRESALIENTE. Para que el programa sea perfecto debe mostrar un mensaje que diga “Esa nota no es válida” sin introducimos un valor menor que cero o diez.

2 Realiza un programa que pida la edad de un alumno y diga si recibe beca o no. Sólo reciben beca los alumnos con 18, 20 y 25 años.

3 Realiza un programa que pida un número entero y diga si es par o impar. Para saber si un número es par, al dividirlo entre 2 su resto debe ser cero.

4 Realiza un programa que pida la altura de un alumno y diga si puede entrar o no en una de las atracciones de un parque temático. Sólo los alumnos que miden entre 1 metro 2 metros pueden hacerlo ((los de 2 justo no pueden). Necesitarás utilizar la función **TRUNC**.

5 Diseña un programa que pida 3 números enteros y los muestre ordenados de mayor a menor.

6 Diseña un programa en el que el ordenador nos pida un número y nos devuelva el factorial de dicho número.

7 El programa pide el número de niños a los que se les va a medir la altura. Después va pidiendo la altura de esos niños. Al final debe mostrar en pantalla cuántos miden más de 1.60 metros.

AYUDA. Para hacerlo deberás utilizar un bucle y dentro comparar con la instrucción **Si**. Para llevar la cuenta, utiliza una variable CONTADOR que vaya incrementando su valor en 1 cada vez que se cumple la condición.

8 Diseña un programa que pida un número y diga si es primo. Para conseguirlo, el programa debe dividir ese número entre todos los números inferiores a él (exceptuando el 1). Si en ninguna de las divisiones, el resto no es cero, el número no será primo. Recuerda que la función que obtenía el resto de una división era MOD.

9 Realiza un programa en el que el ordenador nos da 3 números aleatorios entre 1 y 50 sin que se pueda repetir ninguno.

10 Realiza un programa en el que el ordenador nos pida una palabra y nos devuelva las letras al revés.

Ejemplo: “Dime una palabra” (introducimos una palabra) manzana

El ordenador nos devuelve **anaznam**

11 Realiza un programa con el siguiente funcionamiento:

El ordenador nos pide un número entero.

El ordenador nos pide un segundo número entero, puede ser mayor o menor

El ordenador nos devuelve por pantalla: la suma, la diferencia, el producto, el cociente y el resto de dividir el mayor por el menor.

```
Dime un número
14
Dime un segundo número
3
La suma de los números es 17
La diferencia de los números es 11
El cociente de los números es 4,66
El resto de dividir los números es 2
```

Se introducen cualquier número por teclado

12 Realiza un programa en el que el ordenador genere una suma del número 5 más un número aleatorio entre 1 y 10 y nos pregunte por el resultado. Le damos la respuesta y nos dice si es correcto o falso.

```
**Inicio **
¿Cuánto suman?
5 + 7
| 11
Fallaste
***Fin *****

***Inicio***
¿Cuánto suman?
5 + 9
| 14
Correcto
***Fin***
```

El ordenador pregunta la suma entre 5 y un número aleatorio entre 1 y 10

Si fallamos

Si acertamos

13 Realiza un programa con el siguiente funcionamiento.

El ordenador realiza tres preguntas, respondemos por teclado, si la respuesta es correcta saldrá por pantalla "Correcto" si es falsa "Fallaste", al final nos dará por pantalla el número de aciertos.

```
¿Capital de España?
|Madrid
Correcto
¿Capital de Francia?
|Londres
Fallaste
¿Capital de Alemania?
|Berlín
Correcto
Has tenido 2 aciertos
```

El ordenador nos dirá el número de preguntas acertadas