



Ejercicios propuestos y resueltos en pseudocódigo

28/01/2013 / by Disco Duro de Roer



Hola a todos, hoy os quiero dejar unos cuantos ejercicios propuestos y resueltos (en un spoiler) de pseudocódigo.

Os recomiendo hacer estos ejercicios primero en papel, declarando las variables y hacer un seguimiento cuando lo terminéis de hacer para comprobar que hace lo que pide.



Los ejercicios resueltos están en un spoiler resueltos en papel y PSeInt (por si los quieres guardar en un fichero).

Aquí os dejo algunos posts anteriores para recordar algunos puntos por si lo necesitáis:

- [Variables y constantes.](#)
- [Operadores.](#)
- [Asignación.](#)
- [Operaciones de entrada y de salida.](#)
- [Funciones internas.](#)
- [Instrucciones de control.](#)
- [Instrucciones repetitivas o bucles.](#)
- [Seguimiento.](#)
- [Declaración.](#)
- [Análisis de un problema.](#)
- [Descarga de PSeInt.](#)

Si tienes alguna duda o quieres proponer alguna solución alternativa, deja un comentario en este post o mándanos un mensaje a administrador@discoduroderoer.es También dejare algunos comentarios para explicar que hace cada paso. Puedes ponerlos con //

1) Dadas dos **variables** numéricas **A** y **B**, que el usuario debe teclear, se pide realizar un algoritmo que intercambie los valores de ambas **variables** y muestre cuanto valen al final las dos **variables** (recuerda la asignación).

[Esconder «](#)

Papel:

```
1  Var A, B, C: numerica
2  Inicio
3      Escribir "Introduce el valor de A"
4      Leer A
5      Escribir "Introduce el valor de B"
6      Leer B
7      C<-A
```

**PSeInt:**

```
1 | Proceso ejercicio_1
2 |   Escribir "Introduce el valor de A"
3 |   Leer A
4 |   Escribir "Introduce el valor de B"
5 |   Leer B
6 |   C<-A
7 |   A<-B
8 |   B<-C
9 |   Escribir "A vale " A " y B vale " B
10 | FinProceso
```

2) Algoritmo que lea dos números, calculando y escribiendo el valor de su suma, resta, producto y división.

[Esconder «](#)

Papel:

```
1 | Var numero1, numero2, resultado: numerica
2 | Inicio
3 |   Escribir "Introduce el primer numero"
4 |   Leer numero1
5 |   Escribir "Introduce el segundo numero"
6 |   Leer numero2
7 |   resultado<-numero1+numero2
8 |   Escribir resultado
9 |   resultado<-numero1-numero2
10 |  Escribir resultado
11 |  resultado<-numero1*numero2
12 |  Escribir resultado
13 |  resultado<-numero1/numero2
14 |  Escribir resultado
15 | Fin
```

PSeInt:

```
1 | Proceso ejercicio_2
2 |   Escribir "Introduce el primer numero"
3 |   Leer numero1
4 |   Escribir "Introduce el segundo numero"
5 |   Leer numero2
6 |   //inicializamos la variable resultado a 0 (recomendable)
7 |   resultado<-0
8 |   //sumamos los numeros y escribimos su resultado
9 |   resultado<-numero1+numero2
10 |  Escribir resultado
11 |  //restamos los numeros y escribimos su resultado
12 |  resultado<-numero1-numero2
13 |  Escribir resultado
```



```

18      resultado<-numero1/numero2
19      Escribir resultado
20      FinProceso

```

3) Algoritmo que lea dos números y nos diga cual de ellos es mayor o bien si son iguales (recuerda usar la estructura condicional **SI**)

[Esconder «](#)

Papel:

```

1      Var numero1, numero2: numerica
2      Inicio
3          Escribir "Introduce el primer numero"
4          Leer numero1
5          Escribir "Introduce el segundo numero"
6          Leer numero2
7          Si (numero1>=numero2) Entonces
8              Si (numero1=numero2) Entonces
9                  escribir "los numeros " numero1 " " numero2 " son ig
10             Sino
11                 Escribir numero1 " es el mayor de los dos"
12             FinSi
13         Sino
14             Escribir numero2 " es el mayor de los dos"
15         FinSi
16     Fin

```

PSelnt:

```

1      Proceso ejercicio_3
2          Escribir "Introduce el primer numero"
3          Leer numero1
4          Escribir "Introduce el segundo numero"
5          Leer numero2
6          //comparamos los dos numeros,
7          //si el primero es mayor o igual que el segundo entra
8          Si (numero1>=numero2) Entonces
9              //Si el numero1 y numero2 son iguales entra y escribe que s
10             //Sino lo son escribe que el numero1 es el mayor
11                 Si (numero1=numero2) Entonces
12                     escribir "los numeros " numero1 " " numero2 " son i
13                 Sino
14                     Escribir numero1 " es el mayor de los dos"
15                 FinSi
16             //Si el primer Si es falso, escribe que el numero2 es mayo
17             Sino
18                 Escribir numero2 " es el mayor de los dos"
19             FinSi
20     FinProceso

```

[Esconder «](#)

NOTA: De esta forma que esta hecha, aunque dos de los números sean iguales, siempre dirá el mayor.

Papel:

```
1  Var numero1, numero2, numero3: numerica
2  Inicio
3      Escribir "Introduce el primer numero"
4      Leer numero1
5      Escribir "Introduce el segundo numero"
6      Leer numero2
7      Escribir "Introduce el tercer numero"
8      Leer numero3
9      Si (numero1>numero2 AND numero1>numero3) Entonces
10         Escribir "el numero " numero1 " es el mayor"
11     Sino
12         Si (numero2>numero3) Entonces
13             Escribir "El numero " numero2 " es el mayor"
14         Sino
15             Escribir "El numero " numero3 " es el mayor"
16         FinSi
17     FinSi
18 Fin
```

PSelnt:

```
1  Proceso ejercicio_4
2      Escribir "Introduce el primer numero"
3      Leer numero1
4      Escribir "Introduce el segundo numero"
5      Leer numero2
6      Escribir "Introduce el tercer numero"
7      Leer numero3
8      //comparamos el numero1 con el numero2 y numero3
9      //Si las dos condiciones son verdaderas el numero1 es el ma
10     Si (numero1>numero2 Y numero1>numero3) Entonces
11         Escribir "el numero " numero1 " es el mayor"
12         //si el numero1 no es el mayor,
13         //comparamos el numero2 con el numero3
14     Sino
15         Si (numero2>numero3) Entonces
16             Escribir "El numero " numero2 " es el mayor"
17         Sino
18             Escribir "El numero " numero3 " es el mayor"
19         FinSi
20     FinSi
21 FinProceso
```

5) 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.



```

1  Var numero1, numero2, numero3, resultado: numerica
2  Inicio
3      Escribir "Introduce el primer numero"
4      Leer numero1
5      Escribir "Introduce el segundo numero"
6      Leer numero2
7      Escribir "Introduce el tercer numero"
8      Leer numero3
9      Si (numero1<0) Entonces
10         resultado<-numero1*numero2*numero3
11     Sino
12         resultado<-numero1+numero2+numero3
13     FinSi
14     Escribir resultado
15 Fin

```

PSelnt:

```

1  Proceso ejercicio_5
2      Escribir "Introduce el primer numero"
3      Leer numero1
4      Escribir "Introduce el segundo numero"
5      Leer numero2
6      Escribir "Introduce el tercer numero"
7      Leer numero3
8      //si el numero1 es menor que 0,
9      //multiplicara los numero y sino los sumara
10     Si (numero1<0) Entonces
11         resultado<-numero1*numero2*numero3
12     Sino
13         resultado<-numero1+numero2+numero3
14     FinSi
15     Escribir resultado
16 FinProceso

```

6) Realizar un algoritmo que lea un número por teclado. En caso de que ese número sea 0 o menor que 0, se saldrá del programa imprimiendo antes un mensaje de error. Si es mayor que 0, se deberá calcular su cuadrado y la raíz cuadrada del mismo, visualizando el número que ha tecleado el usuario y su resultado ("Del número X, su potencia es X y su raíz X"). Para calcular la raíz cuadrada se puede usar la función interna **RAIZ(X)** o con una potencia de 0,5.

[Esconder «](#)

Papel:

```

1  Var numero, potencia, raiz: numerica
2  Inicio
3      Escribir "Introduce un numero"
4      Leer numero

```



```

9      Escribir "Su raiz es " raiz_cuadrada
10     Sino
11     Escribir "Error, introduce un numero mayor que 0"
12     FinSi
13     Fin

```

PSeInt:

```

1  Proceso ejercicio_6
2      Escribir "Introduce un numero"
3      Leer numero
4      //si el numero es mayor que 0, calcula la potencia y la raiz
5      //sino muestra un mensaje de error y sale del programa
6      Si (numero>0) Entonces
7          potencia<-numero^2
8          raiz_cuadrada<-RAIZ(numero)
9          Escribir "Su potencia es " potencia
10         Escribir "Su raiz es " raiz_cuadrada
11     Sino
12         Escribir "Error, introduce un numero mayor que 0"
13     FinSi
14 FinProceso

```

7) Un colegio desea saber qué porcentaje de niños y qué porcentaje de niñas hay en el curso actual. Diseñar un algoritmo para este propósito (recuerda que para calcular el porcentaje puedes hacer una regla de 3).

[Esconder «](#)

Papel:

```

1  Var numero_niños, numero_niñas, resultado: numerica
2  Inicio
3      Escribir "Introduce el numero de niños"
4      Leer numero_niños
5      Escribir "Introduce el numero de niñas"
6      Leer numero_niñas
7      porcentaje_niños<-numero_niños*100/(numero_niños+numero_niña
8      porcentaje_niñas<-100-porcentaje_niños
9      Escribir "Hay un " porcentaje_niños " % de niños
10     Escribir "Hay un " porcentaje_niñas " % de niñas
11     Fin

```

PSeInt:

```

1  Proceso Ejercicio_7
2      Escribir "Introduce el numero de niños"
3      Leer numero_niños
4      Escribir "Introduce el numero de niñas"
5      Leer numero_niñas

```



```

10  Escribir "Hay un " porcentaje_niñas " % de niñas"
11  FinProceso

```

8) Una tienda ofrece un descuento del 15% sobre el total de la compra durante el mes de octubre. Dado un mes y un importe, calcular cuál es la cantidad que se debe cobrar al cliente.

[Esconder «](#)

Papel:

```

1  Var mes: cadena
2      importe, total:numerico
3  Inicio
4      Escribir "escribe el importe de la compra"
5      Leer importe
6      Escribir "Introduce el mes"
7      Leer mes
8      Si (mes="octubre") Entonces
9          total<-importe*0.85
10     Sino
11         total<-importe
12     FinSi
13     Escribir total
14 Fin

```

PSelnt:

```

1  Proceso ejercicio_8
2      Escribir "escribe el importe de la compra"
3      Leer importe
4      Escribir "Introduce el mes"
5      Leer mes
6      //Si el mes es octubre, se aplicara el descuento
7      Si (mes="octubre") Entonces
8          total<-importe*0.85
9      Sino
10         total<-importe
11     FinSi
12     Escribir total
13 FinProceso

```

9) Realizar un algoritmo que dado un número entero, visualice en pantalla si es par o impar. En el caso de ser 0, debe visualizar “el número no es par ni impar” (para que un numero sea par, se debe dividir entre dos y que su resto sea 0)



```

1  Var numero: numerica
2  Inicio
3      Escribir "Introduce un numero"
4      Leer numero
5      Si (numero=0) Entonces
6          Escribir "El " numero " no es par ni impar"
7      Sino
8          Si (numero MOD 2=0) Entonces
9              Escribir "El " numero " es par"
10         Sino
11             Escribir "El " numero " no es par"
12         FinSi
13     FinSi
14 Fin

```

PSeInt:

```

1  Proceso ejercicio_9
2      Escribir "Introduce un numero"
3      Leer numero
4      Si (numero=0) Entonces
5          Escribir "El " numero " no es par ni impar"
6      Sino
7          //comprobamos si el numero es par
8          Si (numero MOD 2=0) Entonces
9              Escribir "El " numero " es par"
10         Sino
11             Escribir "El " numero " no es par"
12         FinSi
13     FinSi
14 FinProceso

```

10) Modificar el algoritmo anterior, de forma que si se teclea un cero, se vuelva a pedir el número por teclado (así hasta que se teclee un número mayor que cero) (recuerda la estructura mientras).

[Esconder «](#)

Papel:

```

1  Var numero: numerica
2  Inicio
3      Escribir "Introduce un numero"
4      Leer numero
5      Mientras (numero<=0) hacer
6          Escribir "escribe un numero mayor que 0"
7          Leer numero
8      FinMientras
9      Si (numero MOD 2=0) Entonces
10         Escribir "El " numero " es par"
11     Sino
12         Escribir "El " numero " no es par"

```



PSeInt.

```

1  Proceso ejercicio_10
2      Escribir "Introduce un numero"
3      Leer numero
4      //Hasta que no se introduzca un numero mayor que 0 no saldra
5      Mientras (numero<=0) hacer
6          Escribir "escribe un numero mayor que 0"
7          Leer numero
8      FinMientras
9      Si (numero MOD 2=0) Entonces
10         Escribir "El " numero " es par"
11     Sino
12         Escribir "El " numero " no es par"
13     FinSi
14 FinProceso

```

11) Algoritmo que nos diga si una persona puede acceder a cursar un ciclo formativo de grado superior o no. Para acceder a un grado superior, si se tiene un titulo de bachiller, en caso de no tenerlo, se puede acceder si hemos superado una prueba de acceso.

[Esconder «](#)

Papel:

```

1  Var bachiller, prueba_acceso: cadena
2  Inicio
3      Escribir "¿Tienes el titulo de bachiller?"
4      Leer bachiller
5      si (bachiller="si") Entonces
6          Escribir "Puedes acceder al grado superior"
7      Sino
8          Escribir "¿Tienes la prueba de acceso superada?"
9          Leer prueba_acceso
10         si (prueba_acceso="si") Entonces
11             Escribir "Puedes acceder al grado superior"
12         Sino
13             Escribir "No puedes acceder a un grado superior"
14         FinSi
15     FinSi
16 Fin

```

PSeInt:

```

1  Proceso ejercicio_11
2      Escribir "¿Tienes el titulo de bachiller?"

```



```

7      Escribir "¿Tienes la prueba de acceso superada?"
8      Leer prueba_acceso
9      si (prueba_acceso="si") Entonces
10         Escribir "Puedes acceder al grado superior"
11      Sino
12         Escribir "No puedes acceder a un grado superior"
13      FinSi
14  FinSi
15  FinProceso

```

12) Desarrollar un algoritmo que nos calcule el cuadrado de los 9 primeros números naturales (recuerda la estructura desde-hasta)

[Esconder «](#)

Papel:

```

1  Var num, res: numerica
2  Inicio
3      Desde num<-1 Hasta 9 In 1
4          res<-num^2
5          Escribir num " " res
6      FinDesde
7  Fin

```

PSeInt:

```

1  Proceso ejercicio_12
2      Para num<-1 Hasta 9 Con Paso 1
3          res<-num^2
4          Escribir num " " res
5      FinPara
6  FinProceso

```

NOTA: este ejercicio también se puede hacer con la estructura **Mientras**, pero lo hemos hecho con la estructura **Desde-Hasta** porque sabemos el rango.

13) Se pide representar el algoritmo que nos calcule la suma de los **N** primeros números naturales. **N** se leerá por teclado (no tenemos porque llamar a la variable N, podemos llamarla como queramos).

[Esconder «](#)

Papel:

```

1  Var N, contador, suma: numerica
2  Inicio
3      Leer N
4      suma<-0
5      Desde contador<-1 Hasta 5 In 1

```



10 | Fin**PSeInt:**

```

1  Proceso ejercicio_13
2  Leer N
3  suma<-0
4  Para contador<-1 Hasta 5 Con Paso 1
5      suma<-N+suma
6      N<-N+1
7  FinPara
8  Escribir suma
9  FinProceso

```

14) Se pide representar el algoritmo que nos calcule la suma de los **N** primeros números pares. Es decir, si insertamos un 5, nos haga la suma de 6+8+10+12+14.

[Esconder «](#)

Papel:

```

1  Var N, contador, suma: numerica
2  Inicio
3  Leer N
4  contador<-0
5  Mientras (contador<5) Hacer
6      si (N MOD 2=0) Entonces
7          suma<-N+suma
8          contador<-contador+1
9      FinSi
10     N<-N+1
11 FinMientras
12 Escribir suma
13 Fin

```

PSeInt:

```

1  Proceso ejercicio_14
2  Leer N
3  contador<-0
4  limite<-N
5  Mientras (contador<limite) Hacer
6      si (N MOD 2=0) Entonces
7          suma<-N+suma
8          contador<-contador+1
9      FinSi
10     N<-N+1
11 FinMientras
12 Escribir suma
13 FinProceso

```

[Esconder «](#)**Papel:**

```

1  Var num, contador, suma: numerica
2  Inicio
3    Leer num
4    suma<-0
5    contador<-1
6    Mientras (num<>-1) Hacer
7      suma<-suma+num
8      contador<-contador+1
9      Leer num
10   FinMientras
11   Escribir suma/(contador-1)
12   Fin

```

PSeInt:

```

1  Proceso ejercicio_15
2    Leer num
3    suma<-0
4    contador<-1
5    Mientras (num<>-1) Hacer
6      suma<-suma+num
7      contador<-contador+1
8      Leer num
9    FinMientras
10   Escribir suma/(contador-1)
11   FinProceso

```

16) Teniendo en cuenta que la clave es “eureka”, escribir un algoritmo que nos pida una clave. Solo tenemos 3 intentos para acertar, si fallamos los 3 intentos nos mostrara un mensaje indicándonos que hemos agotado esos 3 intentos. (Recomiendo utilizar un interruptor). Si acertamos la clave, saldremos directamente del programa.

[Esconder «](#)**Papel:**

```

1  Var clave: cadena
2    acierto: booleano
3    contador: numerica
4  Inicio
5    contador<-0
6    acierto<-Falso
7    Mientras (contador<3 AND acierto=falso) Hacer
8      Leer clave
9      si (clave="eureka") Entonces
10         Escribir "la clave es correcta"

```



```

15     si (contador=3 AND acierto=falso) Entonces
16         Escribir "Ya no tienes mas intentos"
17     FinSi
18     Fin

```

PSeInt:

```

1     Proceso ejercicio_16
2         contador<-0
3         //interruptor
4         acierto<-Falso
5         //usamos un interruptor, cuando acertemos,
6         //cambiara y la condicion sera falsa
7         Mientras (contador<3 Y acierto=falso) Hacer
8             //ponemos aqui leer porque con las variables
9             //iniciales entra en el bucle
10            Escribir "introduce la clave"
11            Leer clave
12            si (clave="eureka") Entonces
13                Escribir "la clave es correcta"
14                //el interruptor cambia cuando acertamos
15                acierto<-Verdadero
16            FinSi
17            contador<-contador+1
18        FinMientras
19        //este mensaje solo aparecera si hemos agotado
20        //todos los intentos y no hemos acertado
21        si (contador=3 Y acierto=falso) Entonces
22            Escribir "Ya no tienes mas intentos"
23        FinSi
24    FinProceso

```

17) Algoritmo que lea números enteros hasta teclear 0, y nos muestre el máximo, el mínimo y la media de todos ellos. Piensa como debemos inicializar las variables.

[Esconder «](#)

Papel:

```

1     Var maximo, minimo, num, suma, media, contador: numerica
2     Inicio
3         Leer num
4         minimo<-num
5         maximo<-num
6         suma<-0
7         Mientras (num<>0) Hacer
8             si (num>maximo) Entonces
9                 maximo<-num

```



```

14      suma<-suma+num
15      contador<-contador+1
16      leer num
17      FinMientras
18      media<-suma/(contador)
19      Escribir "El maximo es " maximo
20      Escribir "El minimo es " minimo
21      Escribir "La media es " media
22      Fin

```

PSeInt:

```

1      Proceso ejercicio_17
2      Leer num
3      //maximo y el minimo se inician con el numero que
4      //insertemos para que lo podamos modificar
5      //durante el programa
6      minimo<-num
7      maximo<-num
8      suma<-0
9      Mientras (num<>0) Hacer
10         si (num>maximo) Entonces
11             maximo<-num
12         FinSi
13         si (num<minimo) Entonces
14             minimo<-num
15         FinSi
16         suma<-suma+num
17         contador<-contador+1
18         leer num
19     FinMientras
20     media<-suma/(contador)
21     //escribimos los resultados
22     Escribir "El maximo es " maximo
23     Escribir "El minimo es " minimo
24     Escribir "La media es " media
25     FinProceso

```

18) Algoritmo que visualice la cuenta de los números que son múltiplos de 2 o de 3 que hay entre 1 y 100.

[Esconder «](#)

Papel:

```

1      Var num: numerica
2      Inicio
3          Desde num<-1 Hasta 100 In 1
4              //asi indicamos si un numero es multiplo de 3 o de 2
5              si (num MOD 2=0 OR num MOD 3=0) Entonces
6                  Escribir num
7              FinSi
8          FinDesde

```



```

1  Proceso ejercicio_18
2      Para num<-1 Hasta 100 Con Paso 1
3          //asi indicamos si un numero es multiplo de 3 o de 2
4          si (num MOD 2=0 o num MOD 3=0) Entonces
5              Escribir num
6          FinSi
7      FinPara
8  FinProceso

```

19) Leer tres números que denoten una fecha (día, mes, año). Comprobar que es una fecha válida. Si no es válida escribir un mensaje de error. Si es válida escribir la fecha cambiando el número del mes por su nombre. Ej. si se introduce 1 2 2006, se deberá imprimir "1 de febrero de 2006". El año debe ser mayor que 0. (Recuerda la estructura segun sea).

NOTA: en PSeInt, si queremos escribir sin que haya saltos de linea, al final de la operacion escribir escribimos **sin saltar**.

[Esconder «](#)

Papel:

```

1  Var dia, mes, año: numerica
2  Inicio
3      Escribir "Introduce el día"
4      Leer dia
5      Escribir "Introduce el mes"
6      Leer mes Escribir "Introduce el año"
7      Leer año
8      //comprobamos que la fecha es correcto
9      si ((dia<=31 y dia>=0) AND (mes<=12 y mes>=0) AND año>0) Ent
10         Escribir dia sin saltar
11         //usamos el segun sea para escribir el mes
12         Segun mes Hacer
13             1:
14                 escribir " de enero de" sin saltar
15             2:
16                 escribir " de febrero de " sin saltar
17             3:
18                 escribir " de marzo de " sin saltar
19             4:
20                 escribir " de abril de " sin saltar
21             5:
22                 escribir " de mayo de " sin saltar
23             6:
24                 escribir " de junio de " sin saltar
25             7:
26                 escribir " de julio de " sin saltar
27             8:
28                 escribir " de agosto de " sin saltar
29             9:

```




```

34         escribir " de noviembre de " sin saltar
35     12:
36         escribir " de diciembre de " sin saltar
37     FinSegun
38     Escribir año
39     Sino
40         Escribir "error"
41     FinSi
42     Fin

```

PSeInt:

```

1  Proceso ejercicio_19
2  Escribir "Introduce el dia"
3  Leer dia
4  Escribir "Introduce el mes"
5  Leer mes
6  Escribir "Introduce el año"
7  Leer año
8  //comprobamos que la fecha es correcto
9  si ((dia<=31 y dia>=0) y (mes<=12 y mes>=0) y año>0) Entonces
10     Escribir dia sin saltar
11     //usamos el segun sea para escribir el mes
12     Segun mes Hacer
13         1:
14             escribir " de enero de" sin saltar
15         2:
16             escribir " de febrero de " sin saltar
17         3:
18             escribir " de marzo de " sin saltar
19         4:
20             escribir " de abril de " sin saltar
21         5:
22             escribir " de mayo de " sin saltar
23         6:
24             escribir " de junio de " sin saltar
25         7:
26             escribir " de julio de " sin saltar
27         8:
28             escribir " de agosto de " sin saltar
29         9:
30             escribir " de septiembre de " sin saltar
31         10:
32             escribir " de octubre de " sin saltar
33         11:
34             escribir " de noviembre de " sin saltar
35         12:
36             escribir " de diciembre de " sin saltar
37     FinSegun
38     Escribir año
39     Sino
40         Escribir "error"
41     FinSi
42     FinProceso

```



vale el 50% y la parte teórica el 40%. El algoritmo leerá el nombre del alumno, las tres notas, escribirá el resultado y volverá a pedir los datos del siguiente alumno hasta que el nombre sea una cadena vacía. Las notas deben estar entre 0 y 10, si no lo están, no imprimirá las notas, mostrara un mensaje de error y volverá a pedir otro alumno.

[Esconder «](#)

Papel:

```

1  Var alumno: cadena
2      nota_teoría, nota_problemas, nota_teoría, nota_final: numer
3  Inicio
4      Escribir "Introduce el nombre del alumno"
5      Leer alumno
6      Mientras alumno<>"" Hacer
7          Escribir "Introduce la nota practica"
8          leer nota_practica
9          Escribir "Introduce la nota de problemas"
10         leer nota_problemas
11         Escribir "Introduce la nota de teoría"
12         Leer nota_teoría
13         //NOTA: este si esta agrupado por falta de espacio
14         si (nota_practica<=10 AND nota_practica>=0)
15         AND (nota_problemas<=10 AND nota_problemas>=0)
16         AND (nota_teoría<=10 AND nota_teoría>=0) Entonces
17             Escribir "El alumno " alumno
18             Escribir "La nota practica es " nota_practica
19             Escribir "La nota de problemas es " nota_problemas
20             Escribir "La nota de teoría es " nota_teoría
21             nota_practica<-nota_practica*0.1
22             nota_problemas<-nota_problemas*0.5
23             nota_teoría<-nota_teoría*0.4
24             nota_final<-nota_practica+nota_problemas+nota_teoría
25             Escribir "La nota final es " nota_final
26         Sino
27             Escribir "Has escrito una nota incorrecta, vuelve a
28         FinSi
29
30         Escribir "Introduce el nombre de otro alumno"
31         Leer alumno
32     FinMientras
33 Fin

```

PSelnt:

```

1  Proceso ejercicio_20
2      Escribir "Introduce el nombre del alumno"
3      Leer alumno
4      Mientras alumno<>"" Hacer
5          //si introducimos un nombre de alumno
6          //correcto, entrara en el bucle
7          Escribir "Introduce la nota practica"

```



```

12      Leer nota_teoria
13      //comprobamos si las notas tienen un rango correcto
14      //NOTA: este si esta agrupado por falta de espacio
15      si (nota_practica<=10 y nota_practica>=0)
16      y (nota_problemas<=10 y nota_problemas>=0)
17      y (nota_teoria<=10 y nota_teoria>=0) Entonces
18          Escribir "El alumno " alumno
19          Escribir "La nota practica es " nota_practica
20          Escribir "La nota de problemas es " nota_problemas
21          Escribir "La nota de teoria es " nota_teoria
22          nota_practica<-nota_practica*0.1
23          nota_problemas<-nota_problemas*0.5
24          nota_teoria<-nota_teoria*0.4
25          nota_final<-nota_practica+nota_problemas+nota_teoria
26          Escribir "La nota final es " nota_final
27      Sino
28          Escribir "Has escrito una nota incorrecta, vuelve a in
29      FinSi
30          Escribir "Introduce el nombre de otro alumno"
31          Leer alumno
32      FinMientras
33      FinProceso

```

21) Algoritmo que lea un número entero (lado) y a partir de él cree un cuadrado de asteriscos con ese tamaño. Los asteriscos sólo se verán en el borde del cuadrado, no en el interior.

Ejemplo, para lado = 4 escribiría:

```

* * * *
*   *
*   *
*   *
* * * *

```

[Esconder «](#)

NOTA: este tipo de ejercicios cambian mucho en papel y en PSeInt, en papel es mas sencillo que en PSeInt.

Papel:

```

1      Var lado, asterisco, espacio: numerica
2      Inicio
3          Leer lado
4          Desde asterisco<-1 Hasta lado In 1
5              Escribir "*"
6          FinDesde

```



```

11      Escribir " "
12      FinDesde
13      Escribir "*"
14      escribir " "
15      FinDesde
16      Desde asterisco<-1 Hasta lado In 1
17      Escribir "*"
18      FinDesde
19      Fin

```

PSeInt:

```

1      Proceso ejercicio_21
2      Leer lado
3      Para asterisco<-1 Hasta lado Con Paso 1
4      Escribir "*" Sin Saltar
5      FinPara
6      escribir " "
7      Para asterisco<-1 Hasta lado-2 Con Paso 1
8      Escribir "*" Sin Saltar
9      Para espacio<-1 Hasta lado-2 Con Paso 1
10     Escribir " " sin saltar
11     FinPara
12     Escribir "*" Sin Saltar
13     escribir " "
14     FinPara
15     Para asterisco<-1 Hasta lado Con Paso 1
16     Escribir "*" Sin Saltar
17     FinPara
18     FinProceso

```

22) Algoritmo que lea un número entero (altura) y a partir de él cree una escalera invertida de asteriscos con esa altura. Debera quedar así, si ponemos una altura de 5.

```

*****
****
***
**
*

```

[Esconder «](#)

Papel:

```

1      Var altura, asterisco, espacio, espacio_linea, asterisco_linea:
2      Inicio
3      Leer altura

```



```

8      Escribir " "
9      FinDesde
10     Desde asterisco_linea<-1 hasta asterisco In 1
11         Escribir "*"
12     FinDesde
13     asterisco<-asterisco-1
14     espacio<-espacio+1
15     Escribir " "
16 FinDesde
17 Fin

```

PSeInt:

```

1  Proceso ejercicio_22
2  Leer altura
3  //iniciamos las variables segun lo necesitamos
4  asterisco<-altura
5  espacio<-0
6  Para linea<-1 hasta altura con paso 1
7      //escribimos los espacios iniciales
8      Para espacio_linea<-0 hasta espacio con paso 1
9          Escribir " " sin saltar
10     FinPara
11     //escribimos los asteriscos de la escalera
12     Para asterisco_linea<-1 hasta asterisco con paso 1
13         Escribir "*" sin saltar
14     FinPara
15     //aumentamos los espacios y disminuimos los asteriscos
16     asterisco<-asterisco-1
17     espacio<-espacio+1
18     Escribir " "
19 FinPara
20 FinProceso

```

23) Algoritmo que dado un año, nos diga si es bisiesto o no. Un año es bisiesto bajo las siguientes condiciones:

- Un año divisible por 4 es bisiesto y no debe ser divisible entre 100.
- Si un año es divisible entre 100 y además es divisible entre 400, también resulta bisiesto.

NOTA: este ejercicio tiene muchas formas de hacerlo si quieres comprobar que es correcto, puedes probarlo [aquí](#), también encontraras información sobre las condiciones de cuando un año es bisiesto.

[Esconder «](#)

Papel:



```

5      si (año MOD 4=0 y año MOD 100<>0)Entonces
6          Escribir "El año " año " es bisiesto"
7      Sino
8          si (año MOD 400=0 y año MOD 100=0) Entonces
9              Escribir "El año " año " es bisiesto"
10         Sino
11             escribir "El año " año " no es bisiesto"
12         FinSi
13     FinSi
14 Fin

```

PSelnt:

```

1      Proceso ejercicio_23
2          Escribir "Introduce un año"
3          leer año
4          si (año MOD 4=0 y año MOD 100<>0)Entonces
5              Escribir "El año " año " es bisiesto"
6          Sino
7              si (año MOD 400=0 y año MOD 100=0) Entonces
8                  Escribir "El año " año " es bisiesto"
9              Sino
10                 escribir "El año " año " no es bisiesto"
11             FinSi
12         FinSi
13     FinProceso

```

24) El siguiente es el menú de un restaurante de bocadillos. Diseñar un algoritmo capaz de leer el número de unidades consumidas de cada alimento ordenado y calcular la cuenta total. Vamos a suponer que estos precios son fijos, es decir, que son constantes (recuerda que en PSelnt no se usa comas para separar la parte decimal de la parte entera).

PRODUCTO	PRECIO
Bocadillo de jamón	1,5 €
Refresco	1,05 €
Cerveza	0,75 €

[Esconder «](#)

Papel:

```

1      Inicio
2          JAMON=1.5
3          REFRESCO=1.05
4          CERVEZA=0.75
5          total=0
6          Escribir "Introduce la cantidad de bocadillos de jamon"
7          Leer cant_jamon
8          Escribir "Introduce la cantidad de refresco"
9          Leer cant_refresco

```



```

14      +(cant_cerveza*CERVEZA))
15      Escribir total
16      Fin

```

PSeInt:

```

1      Proceso ejercicio_24
2          JAMON=1.5
3          REFRESCO=1.05
4          CERVEZA=0.75
5          total=0
6          Escribir "Introduce la cantidad de bocadillos de jamon"
7          Leer cant_jamon
8          Escribir "Introduce la cantidad de refresco"
9          Leer cant_refresco
10         Escribir "Introduce la cantidad de cerveza"
11         Leer cant_cerveza
12         //NOTA: los dos siguientes van en una linea
13         total<-((cant_jamon*JAMON)+(cant_refresco*REFRESCO)
14         +(cant_cerveza*CERVEZA))
15         Escribir total
16     FinProceso

```

25) Algoritmo que nos calcule las raíces de una ecuación de segundo grado. Controlar el caso de que las soluciones sean imaginarias.

[Esconder «](#)

Papel:

```

1      Var a,b, c, raizcua, x1, x2: numerica
2      Inicio
3          Escribir "escibe el valor de a"
4          Leer a
5          Escribir "escibe el valor de b"
6          Leer b
7          Escribir "escibe el valor de c"
8          Leer c
9          neg<-b
10         raizcua<-(b^2-4*a*c)^0.5
11         Si raizcua<=0 Entonces
12             Escribir "La ecuacion no se puede resolver"
13         Sino
14             x1<-(neg+raizcua)/(2*a)
15             x2<-(neg-raizcua)/(2*a)
16             Escribir x1
17             Escribir x2
18         FinSi
19     Fin

```

PSeInt:



```
5 Leer b
6 Escribir "escibe el valor de c"
7 Leer c
8 neg<--b
9 raizcua<-(b^2-4*a*c)^0.5
10 Si raizcua<=0 Entonces
11     Escribir "La ecuacion no se puede resolver"
12 Sino
13     totalx1<-(NEG+Raizcua)/(2*a)
14     totalx2<-(NEG-Raizcua)/(2*a)
15     Escribir Totalx1
16     Escribir Totalx2
17 FinSi
18 FinProceso
```

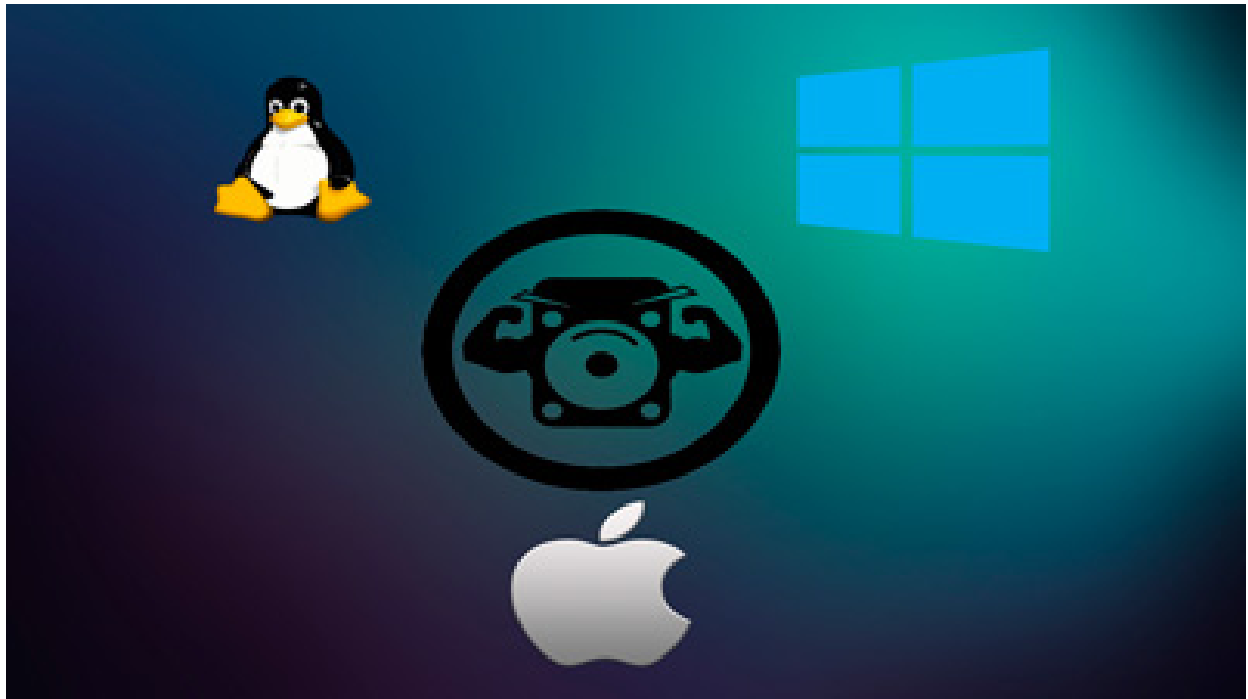
También os dejamos los videos que tenemos en Youtube haciendo ejercicios de pseudocódigo.

Ejercicios Pselnt - Básicos #1



Espero que os sea de ayuda.

[bucles](#)[constantes](#)[discoduroderoer](#)[ejercicios](#)[papel](#)[propuesto](#)[pseint](#)[pseudocodigo](#)[resuelto](#)[variables](#)



Caracteres comodines

30/10/2012



Entrada y salida de datos en Java

17/09/2013

USB-AV Antivirus 2013



DONACIONES

Si te gusta mi trabajo, invítame a un café.

Donar

SUSCRÍBETE A NUESTRO BOLETÍN

Correo Electrónico *

¡SUSCRÍBETE!

¿BUSCAS TRABAJO RELACIONADA CON LA INFORMÁTICA?

neuvoo

Su búsqueda de empleo comienza aquí.