

Ejercicios de análisis y evaluación

EJEMPLO 1

ENUNCIADO:

Compara los siguientes procedimientos para realizar la misma acción

```
Funcion DIVISION_DE_DOS_NUMEROS_opc3 ()  
  Escribir "Introduce el dividendo: "  
  Leer dividendo;  
  Escribir "Introduce el divisor: "  
  Leer divisor;  
  resultado<- dividendo/divisor;  
  Escribir "El resultado de dividir ", dividendo, " entre ",divisor, " es: ", resultado;  
Fin Funcion  
  
Funcion DIVISION_DE_DOS_NUMEROS_opc2 (dividendo,divisor)  
  resultado<- dividendo/divisor;  
  Escribir "El resultado de dividir ", dividendo, " entre ",divisor, " es: ", resultado;  
Fin Funcion  
  
Funcion resultado <- DIVISION_DE_DOS_NUMEROS_opc1 ( )  
  Escribir "Introduce el dividendo: "  
  Leer dividendo;  
  Escribir "Introduce el divisor: "  
  Leer divisor;  
  resultado<- dividendo/divisor;  
Fin Funcion
```

Escribe en cada caso como los utilizarías en un programa que a partir de dos números nos devolviera su cociente.

SOLUCIÓN

- Opción 1: Es una función que devuelve un valor, por lo que dentro del programa no se puede invocar por separado sino siempre dentro de una expresión.



```

1  Funcion resultado <- DIVISION_DE_DOS_NUMEROS_opc1 ( )
2      Escribir "Introduce el dividendo: "
3      Leer dividendo;
4      Escribir "Introduce el divisor: "
5      Leer divisor;
6      resultado<- dividendo/divisor;
7  Fin Funcion
8
9
10 Algoritmo DIVISION_DE_DOS_NUMEROS OPCION1
11     res<- DIVISION_DE_DOS_NUMEROS_opc1 ( );
12     Escribir "El resultado es: ",res;
13 FinAlgoritmo
14

```

Opción 2: Es una función que no devuelve un valor (procedimiento) por lo tanto puede ser invocada por separado, si bien necesita que se le pasen dos parámetros para realizarse, que el programa habrá tenido que proporcionar antes.

```

Funcion DIVISION_DE_DOS_NUMEROS_opc2 (dividendo,divisor)
    resultado<- dividendo/divisor;
    Escribir "El resultado de dividir ", dividendo, " entre ",divisor, " es: ", resultado;
Fin Funcion

Algoritmo DIVISION_DE_DOS_NUMEROS OPCION2
    Escribir "Introduce el dividendo: "
    Leer dividendo;
    Escribir "Introduce el divisor: "
    Leer divisor;
    DIVISION_DE_DOS_NUMEROS_opc2(dividendo,divisor);
FinAlgoritmo

```

- Opción 3: Es un procedimiento que no tiene parámetros, por lo que los datos habrán de ser recabados por el propio procedimiento.

```

Funcion DIVISION_DE_DOS_NUMEROS_opc3 ()
    Escribir "Introduce el dividendo: "
    Leer dividendo;
    Escribir "Introduce el divisor: "
    Leer divisor;
    resultado<- dividendo/divisor;
    Escribir "El resultado de dividir ", dividendo, " entre ",divisor, " es: ", resultado;
Fin Funcion

Algoritmo DIVISION_DE_DOS_NUMEROS OPCION3
    DIVISION_DE_DOS_NUMEROS_opc3();
FinAlgoritmo

```



La elección entre unas u otras vendrá determinada por lo que sea más conveniente en el programa. La opción 3 es la que resulta más conveniente cuando una misma expresión se va a utilizar repetidas veces a lo largo del código, mientras que la opción 1 puede resultar conveniente si ese resultado quisiéramos introducirlo en posteriores instrucciones para tomar decisiones o realizar otras operaciones.

EJEMPLO2

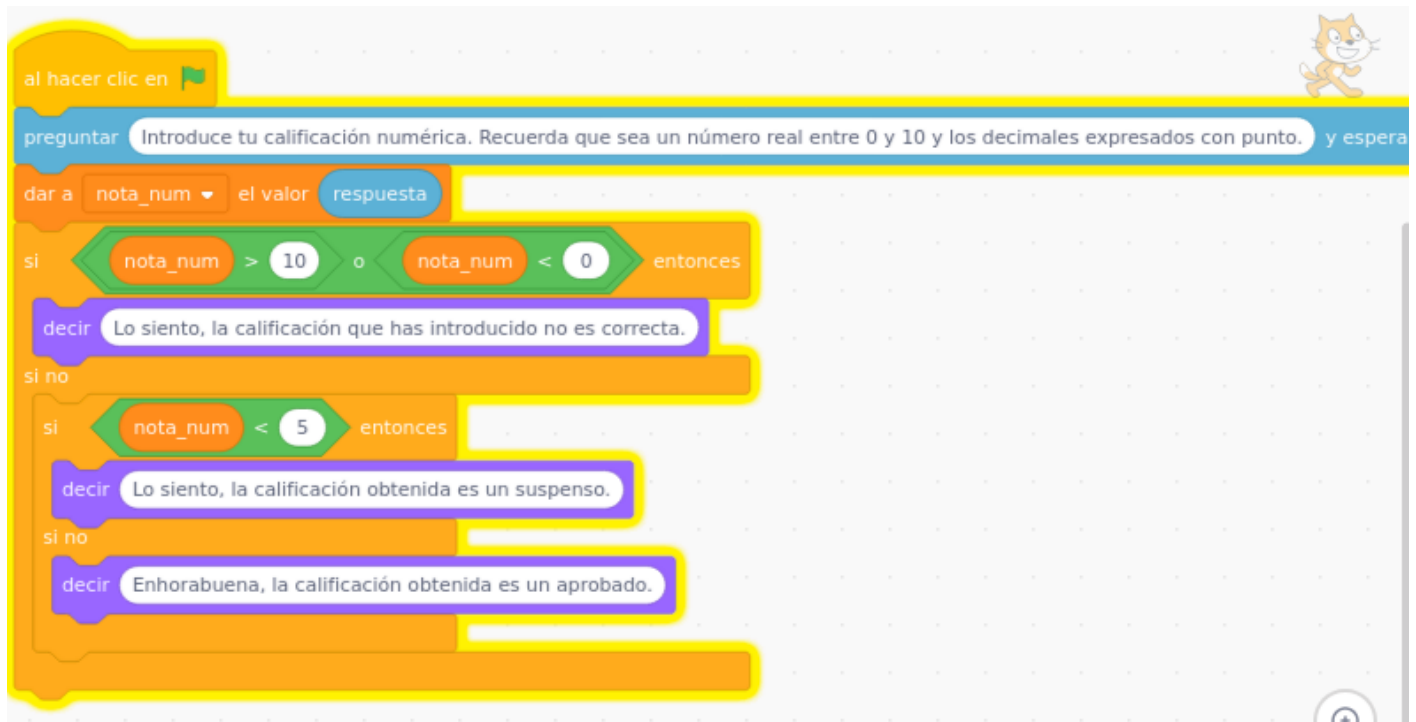
ENUNCIADO

Compara los siguientes programas.

Programa 1



Programa 2



a) ¿Hacen lo mismo?

b) Justifica sus diferencias y explica cuál utilizarías tú y por qué.

SOLUCIÓN

a) Efectivamente hacen lo mismo. Es un programa que al introducir una calificación numérica evalúa que esté en el rango correcto y si no, devuelve un mensaje de error, y si está en el rango correcto evalúa si corresponde a un APROBADO o un SUSPENSO. Es el programa descrito en el último ejemplo del [capítulo anterior](#).

b) La diferencia es:

- El programa 1 **primero evalúa si la nota introducida está en el intervalo correcto** mediante un operador Y, (ha de ser mayor O igual que cero Y menor O igual que 10) y en ese caso devuelve el mensaje de SUSPENSO si es menor que 5 y aprobado en el otro caso. Si no está en el intervalo devuelve un mensaje de error. Como en Scratch no existe el operador \leq hay que ir concatenando operadores Y y O y acaba resultando un código complejo.
- El programa 2 **primero evalúa si la nota introducida está fuera del intervalo correcto** (cosa que sucede para valores estrictamente menores que 0 o mayores que 10) y en ese caso devuelve el mensaje de error, y en el contrario anida otro condicional donde devuelve el mensaje de SUSPENSO si es menor que 5 y aprobado en el otro caso.



En el **programa 2** el código es más sencillo y por lo tanto es la opción más recomendable.

Financiado por el Ministerio de Educación y Formación Profesional y por la Unión Europea - NextGenerationEU



Revision #6

Created 5 June 2022 16:07:29 by Ana López Floría

Updated 17 January 2023 15:49:06 by Equipo CATEDU