Ejercicios de creación

EJEMPLO 1

ENUNCIADO

A partir de los ejercicios vistos anteriormente realiza un programa que simule ser una pequeña calculadora con las cuatro operaciones básicas (sumar, restar, multiplicar y dividir):

- Que muestre un menú inicial donde se elija la operación que se va a realizar.
- Si alguien escoge una opción no contemplada debe volver a mostrar las opciones iniciales.
- Que solicite los números implicados en cada operación y muestre el resultado.
- Que de opción a reiniciar el programa una vez realizada una operación.

SOLUCIÓN

Seguimos los mismos pasos vistos en el capítulo anterior en la creación de un programa.

Pasos 1 y 2: Análisis y diagrama de flujo del programa *Calculadora.*

Los elementos implicados son

- **Salidas**: Mostrar menú inicial, preguntar por números implicados, mostrar resultado.
- **Entradas**: Operación a realizar (operac), números implicados (sumando1, sumando2....), reinicio.
- **Almacenamiento de datos**: operacion a realizar (operac), números implicados, resultado, reinicio (S/N)
- **Operaciones**: bucle y condicional (SEGUN, SI-ENTONCES, REPETIR hasta que), operaciones lógicas (comparaciones, Y, O), operaciones aritméticas.

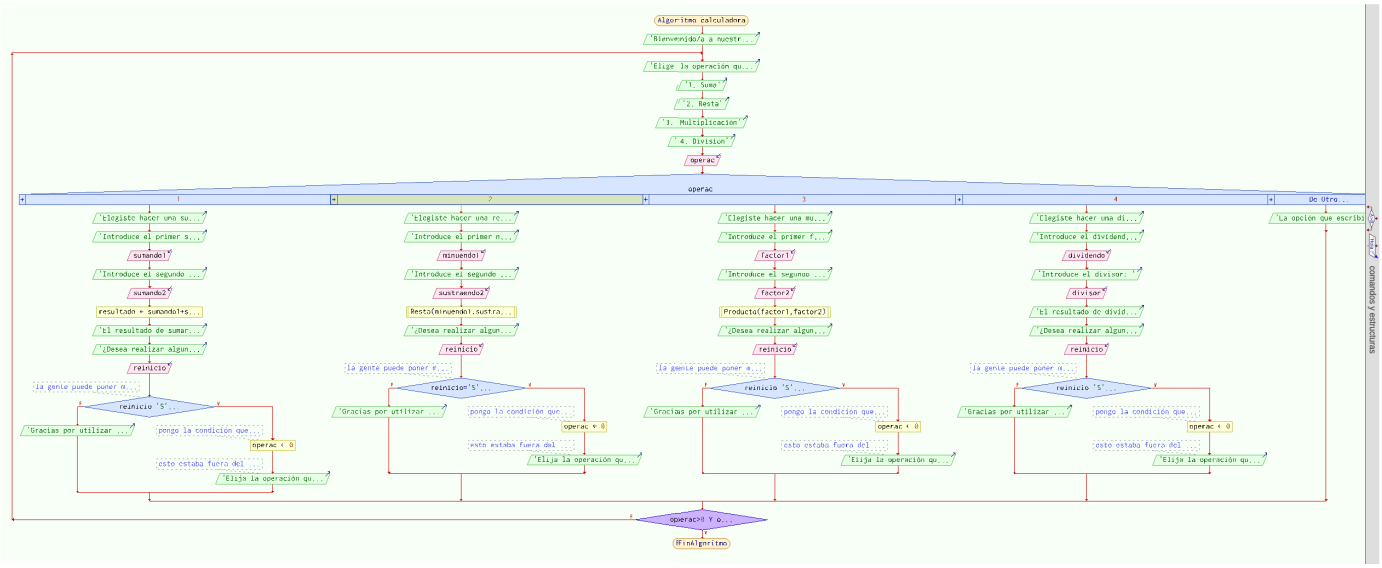
En cuanto a la estructura del programa se propone realizar funciones parciales para cada una de las operaciones.

Copyright 2026

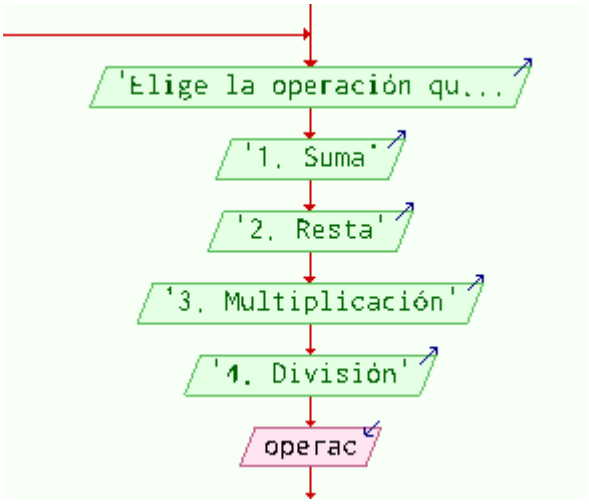
Para mostrar el menú se propone la estructura SEGUN ya que las opciones son discretas (4 operaciones) dentro de una estructura REPETIR hasta que la opción seleccionada sea la correcta.

El reinicio se realiza al finalizar cada operación, preguntando y si la respuesta es afirmativa, modificar la condición que volvía a mostrar el menú inicial (operac).

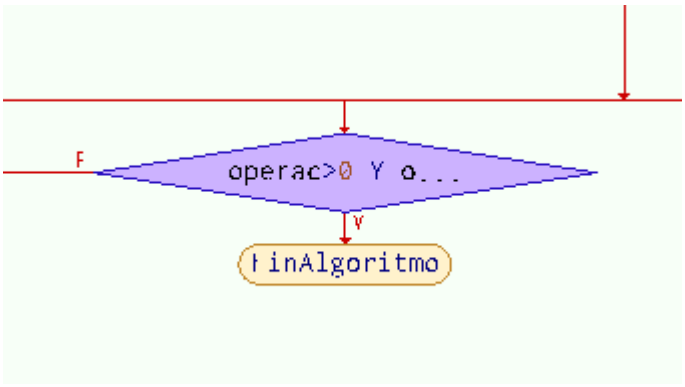
Diagrama de flujo:



Las dimensiones del diagrama no permiten distinguirlo bien, pero sí notar que hay una estructura REPETIR HASTA QUE la variable operac tenga un valor comprendido entre 1 y 4, que muestra el mensaje inicial con las operaciones a elegir.



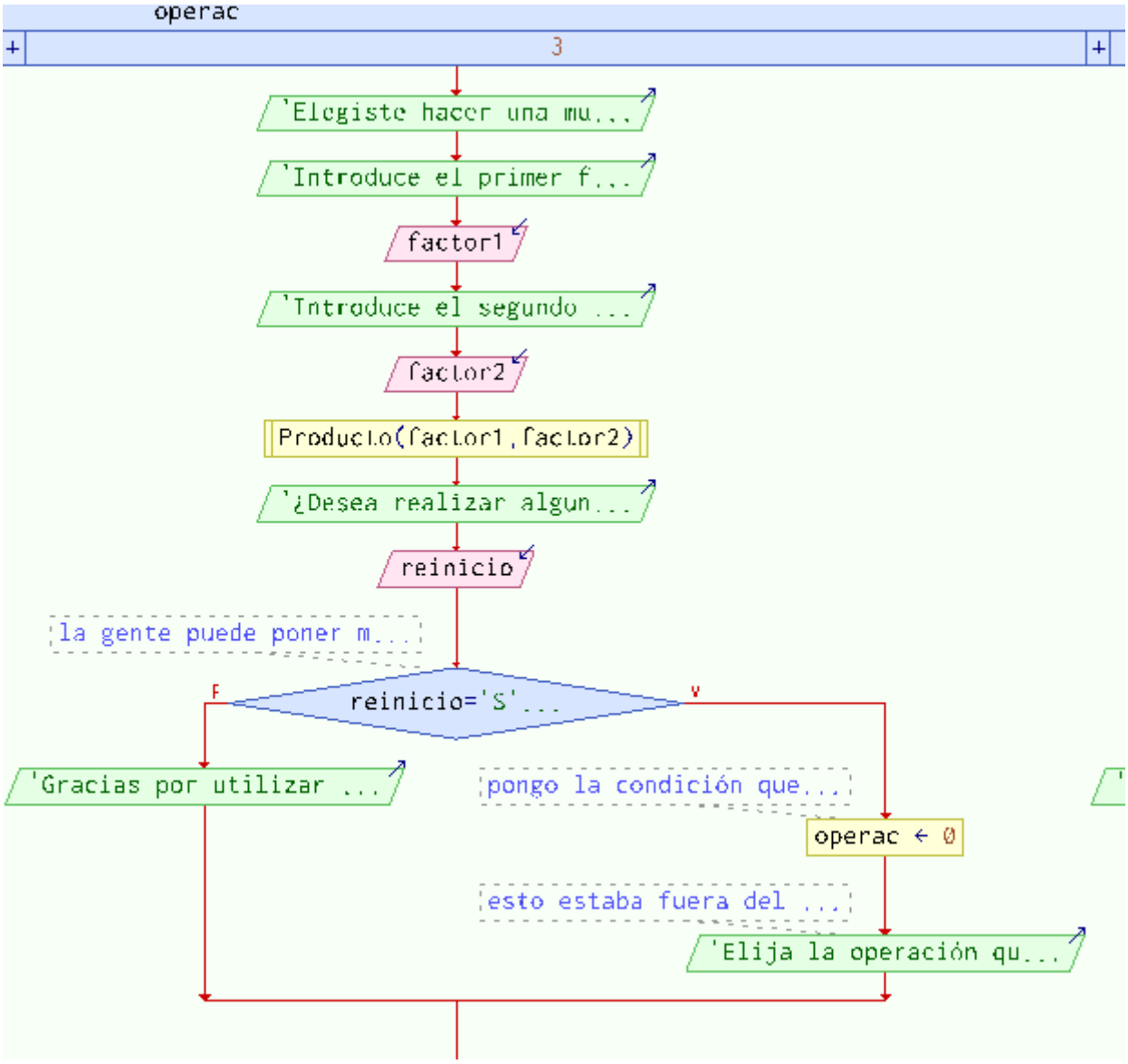
Copyright 2026



Dentro de ella una estructura SEGÚN que abre 5 opciones posibles, la última la de que se haya elegido un número distinto del 1 al 4 que vuelve a iterar el proceso, y las otras 4 una por cada operación.

También que cada una de las 4 operaciones termina con un bloque condicional para preguntar por el reinicio, y en caso de querer reiniciar, le da a la variable operac un valor diferente de 1,2, 3 y 4 lo que reactiva el bucle repetir hasta que.

Copyright 2026



Pasos 3, 4 y 5: Codificación, compilación y verificación del programa *Calculadora* con *PSeInt.*

Empezamos por definir las funciones. La suma la haremos directamente dentro del algoritmo principal para ver la diferencia.

Copyright 2026

```
Funcion resultado <- Division ( a,b )
    resultado<- a/b;
Fin Funcion

Funcion Resta ( a,b )
    resultado<-a-b;
    Escribir "El resultado de restar ",a," y ", b, " es ",resultado;
Fin Funcion

Funcion Producto ( a,b )
    resultado<-a*b;
    Escribir "El resultado de multiplicar ",a," por ", b, " es ",resultado;
Fin Funcion
```

El algoritmo principal quedaría de la siguiente forma:

```
Algoritmo calculadora
    Escribir "Bienvenido/a a nuestra calculadora"
    Repetir
        Escribir "Elige la operación que quieres realizar";
        Escribir "1. Suma";
        Escribir "2. Resta"
        Escribir "3. Multiplicación";
        Escribir "4. División";
        Leer operac;
        Segun operac Hacer
            1:
                Escribir "Elegiste hacer una suma.";
                Escribir "Introduce el primer sumando: "
                Leer sumando1;
                Escribir "Introduce el segundo sumando: "
                Leer sumando2;
                resultado← sumando1+sumando2;
                Escribir "El resultado de sumar ",sumando1," y ", sumando2, " es ",resultado;
                Escribir "¿Desea realizar alguna otra operación?(S/N)"
                Leer reinicio;
                Si reinicio='S'y reinicio='s' Entonces //la gente puede poner mayúsculas o minúsculas
                    operac←0; //pongo la condición que me vuelve a meter dentro del bucle repetir
                    Escribir "Elija la operación que desee realizar" //esto estaba fuera del bucle repetir, hay que ponerlo
                SiNo
                    Escribir "Gracias por utilizar el programa";
                Fin Si
```

Copyright 2026

```
        2:
            Escribir "Elegiste hacer una resta.";
            Escribir "Introduce el primer numero: "
            Leer minuendo1;
            Escribir "Introduce el segundo numero: "
            Leer sustraendo2;
            Resta(minuendo1, sustraendo2);
            Escribir "¿Desea realizar alguna otra operación?(S/N)"
            Leer reinicio;
            Si reinicio='S'∨ reinicio='s' Entonces //la gente puede poner mayúsculas o minúsculas
                operac←0; //pongo la condición que me vuelve a meter dentro del bucle repetir
                Escribir "Elija la operación que desee realizar" //esto estaba fuera del bucle repetir, hay que ponerlo
            SiNo
                Escribir "Gracias por utilizar el programa";
            Fin Si
        3:
            Escribir "Elegiste hacer una multiplicación.";
            Escribir "Introduce el primer factor: "
            Leer factor1;
            Escribir "Introduce el segundo factor: "
            Leer factor2;
            Producto(factor1,factor2);
            Escribir "¿Desea realizar alguna otra operación?(S/N)"
            Leer reinicio;
            Si reinicio='S'∨ reinicio='s' Entonces //la gente puede poner mayúsculas o minúsculas
                operac←0; //pongo la condición que me vuelve a meter dentro del bucle repetir
                Escribir "Elija la operación que desee realizar" //esto estaba fuera del bucle repetir, hay que ponerlo
            SiNo
                Escribir "Gracias por utilizar el programa";
            Fin Si

        4:
            Escribir "Elegiste hacer una división.";
            Escribir "Introduce el dividendo: "
            Leer dividendo;
            Escribir "Introduce el divisor: "
            Leer divisor;
            Escribir "El resultado de dividir ",dividendo," entre ", divisor, " es ",Division(dividendo,divisor);
            Escribir "¿Desea realizar alguna otra operación?(S/N)"
            Leer reinicio;
            Si reinicio='S'∨ reinicio='s' Entonces //la gente puede poner mayúsculas o minúsculas
                operac←0; //pongo la condición que me vuelve a meter dentro del bucle repetir
                Escribir "Elija la operación que desee realizar" //esto estaba fuera del bucle repetir, hay que ponerlo
            SiNo
                Escribir "Gracias por utilizar el programa";
            Fin Si
        De Otro Modo:
            Escribir "La opción que escribiste no es correcta. Elige otra."
    Fin Segun

Hasta Que operac>0 ∧ operac<5;


FinAlgoritmo
```

EJEMPLO2

ENUNCIADO

Crea un programa en Scratch para aprender las tablas de multiplicar. Las condiciones del programa son:

- Al inicio un personaje pregunta si quieres participar o no. Tiene que seguir preguntando hasta que se le diga que sí.
- Después preguntará del 1 al 10 qué tabla de multiplicar quiere practicar.
- A continuación mostrará en pantalla las operaciones de la tabla de multiplicar del número introducido.
- Al terminar se despedirá.

SOLUCIÓN

Seguimos los mismos pasos vistos en el capítulo anterior en la creación de un programa.

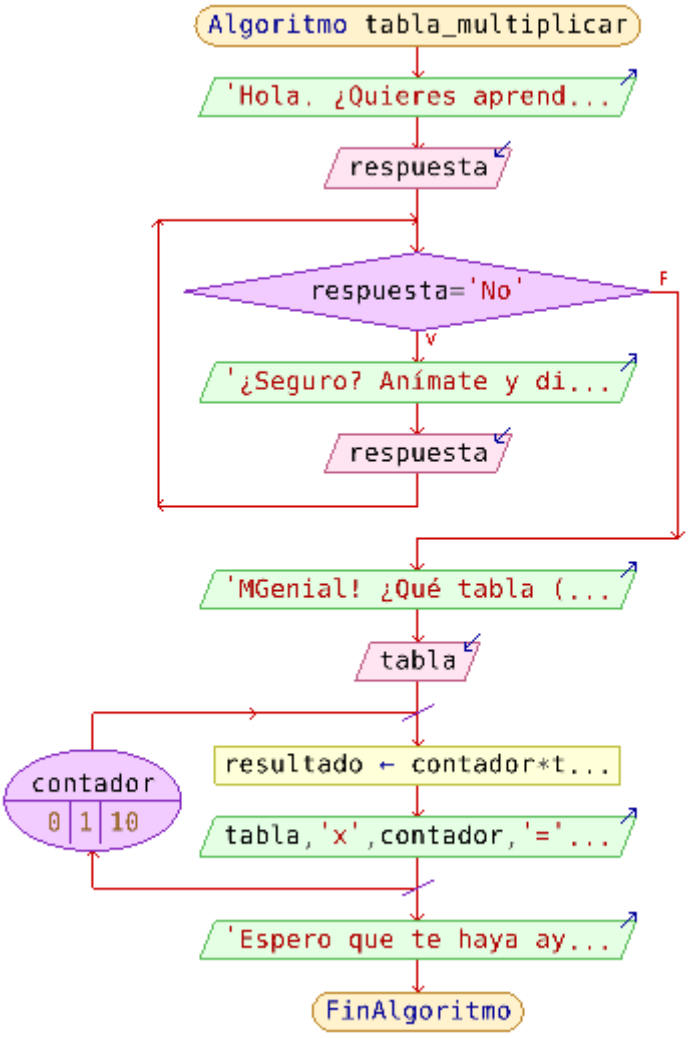
Pasos 1 y 2: Análisis y diagrama de flujo del programa *Tabla de multiplicar*

Los elementos implicados son

- **Salidas**: Saludar, preguntar si se participa, preguntar la tabla a recitar, recitar la tabla, saludar al final.
- **Entradas**: Respuesta Sí/No a participar, número de la tabla.
- **Almacenamiento de dato**s: almacenamos las respuestas obtenidas, y una variable contador para recorrer la tabla del 0 al 10
- **Operaciones**: bucle condicional (repetir mientras la respuesta sea no), bucle con número prefijado (multiplicar del 0 al 10), multiplicaciones, comparaciones, concatenaciones.

Diagrama de flujo:

Copyright 2026



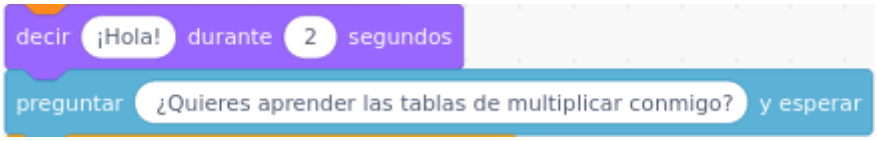
Pasos 3, 4 y 5: Codificación, compilación y verificación del programa *Tabla de multiplicar* con Scratch

Como siempre empezamos por definir las variables. Definimos una variable contador y le asignamos el valor 0 al inicio del programa. La otra variable no será necesario crearla puesto que al ser respuesta de una pregunta, la crea el programa por defecto.

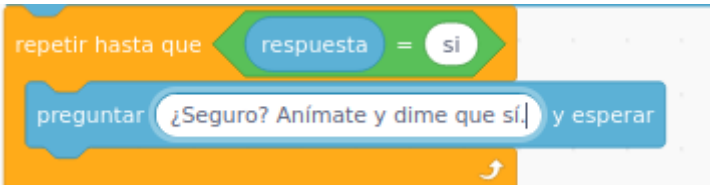




Saludamos y realizamos la pregunta con los bloques correspondientes de **Apariencia** y **Sensores** ya que son **Salidas**.

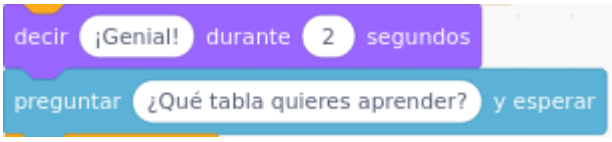


Ahí comienza el primer bucle, en este caso condicional puesto que depende de la respuesta obtenida. En scratch el bloque que realiza esta operación es **Repetir hasta que**



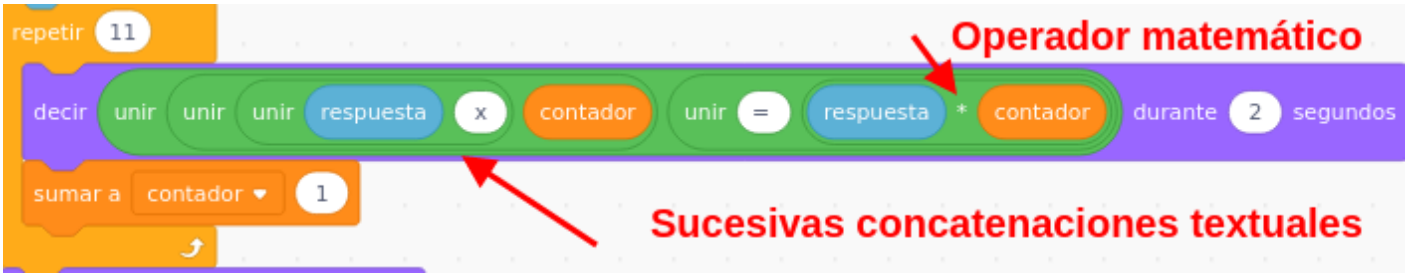
> Cuando realizamos comparaciones con textos, hay que tener en cuenta las variaciones de mayúsculas, minúsculas, tildes, etc...Lo mejor en ese caso sería usar el operador "O" que nos admite como válidas cualquiera de las posibilidades (Sí, Si, SI, si)

Una vez obtenida la respuesta afirmativa (ENTRADA), pasamos a preguntar de nuevo por el número del 1 al 10 del que se quiere usar la tabla (SALIDA).

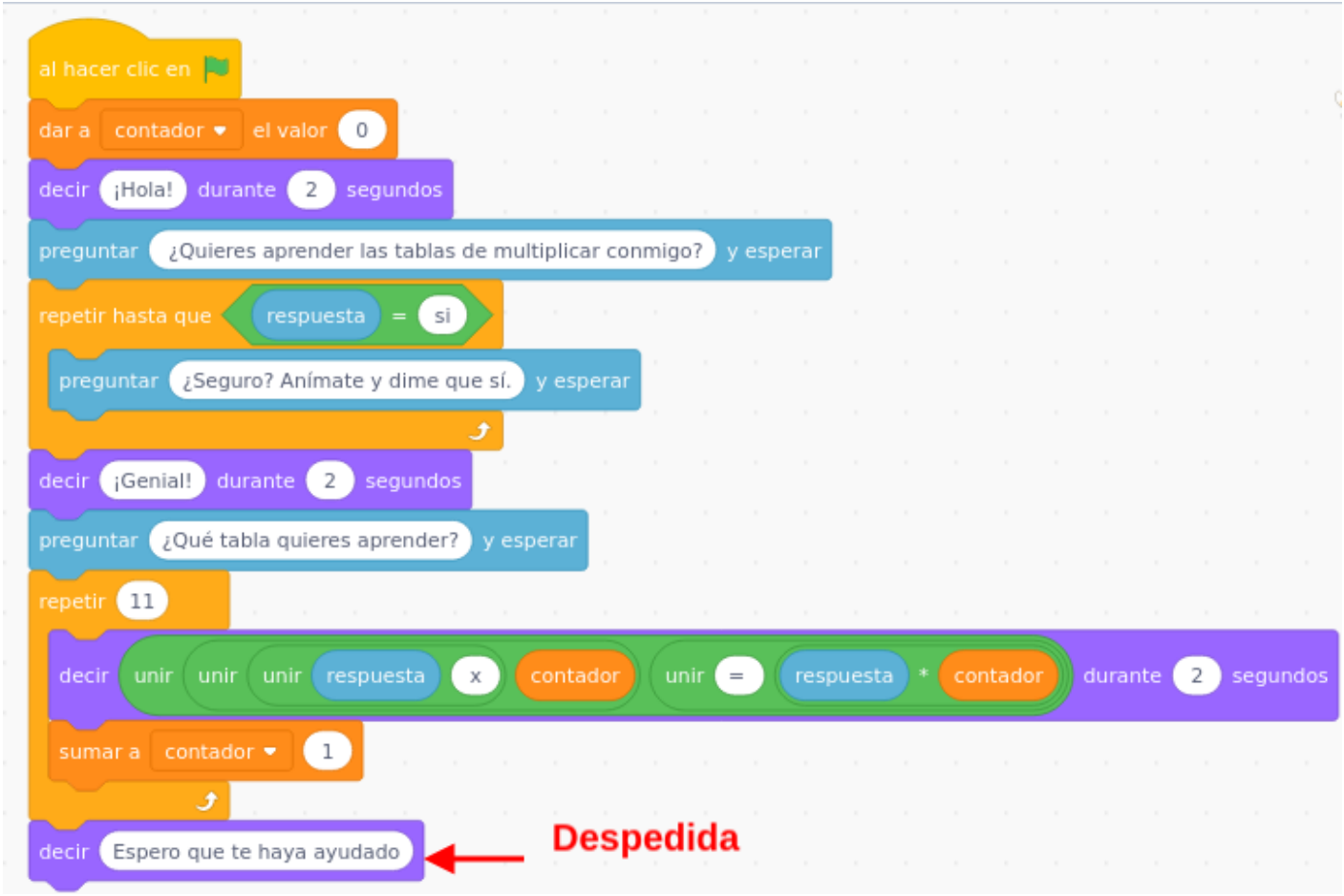


Con la respuesta obtenida (ENTRADA) realizamos otro bucle en este caso del 0 al 10 por lo que son 11 veces, donde se va a mostrar en pantalla al personaje diciendo el numero de la tabla x contador = resultado y el valor del contador se va incrementando de uno en uno en cada iteración.

Copyright 2026

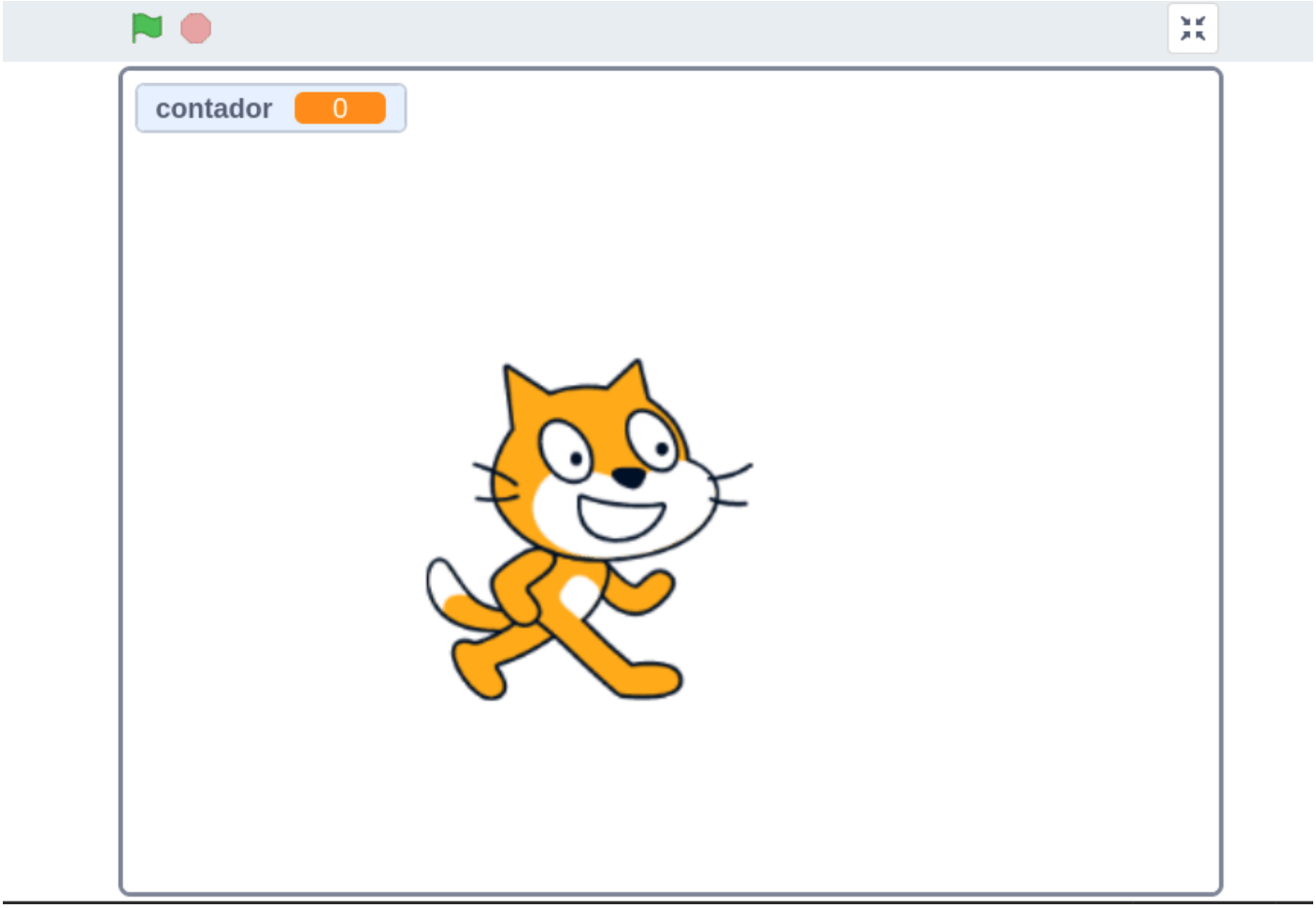


Tras una última salida con la despedida, el programa quedaría así.



Por último vendría el momento de verificación del funcionamiento.

Copyright 2026



Consideraciones finales:

- Este programita se podía haber hecho de muchas otras formas, como casi todos los programas. Se pueden utilizar estructuras condicionales anidadas, por ejemplo. Te invitamos a que explores algunas de ellas.
- Se ha escogido esta forma porque es una de las que menos bloques de código requiere. Es una buena práctica para nuestro alumnado exponerle a crear un programa de distintas formas y hacer un análisis crítico de cada una de ellas.
- Es bueno tener presente siempre en programación las siglas DRY (Don't Repeat Yourself) Si se repiten muchas líneas de código o muchos bloques iguales, probablemente hay una forma más eficiente de hacerlo.

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

Financiado por la Unión Europea NextGenerationEU
Plan de Recuperación, Transformación y Resiliencia
GOBIERNO DE ARAGON

Copyright 2026

Revision #11
Created 2022-06-05 16:09:50 CEST by Ana López Floría
Updated 2023-01-17 15:49:12 CET by Equipo CATEDU

Copyright 2026