

Situación de aprendizaje 6. Programando las tablas de multiplicar

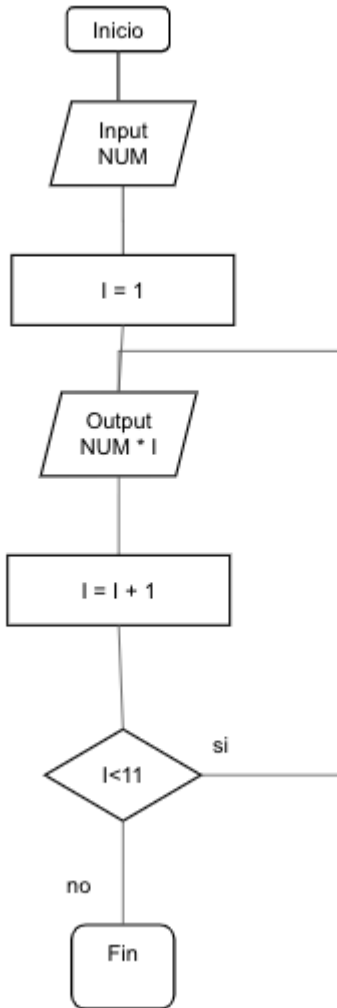
Esta práctica nos viene genial para practicar el uso de bucles junto con variables que se van a tener que ir incrementando para lograr nuestros objetivos.

Podemos plantear dos ejercicios referidos a las tablas de multiplicar:

1. Pedir el número de la tabla de multiplicar que queramos sacar por pantalla. Lo podemos complicar para que el usuario solo pueda meter números del 1 al 10 ambos inclusive y también para que introduzca S o N como únicas respuestas posibles ante la pregunta de si quiere seguir jugando.
2. Crear un programa que automáticamente nos visualice en pantalla las 10 tablas de multiplicar. Este ejercicio nos viene muy bien para practicar el concepto de meter un bucle dentro de otro, algo que al alumnado novel en programación le cuesta entender un poco al principio

Opción 1. Mostrando una tabla de multiplicar elegida por el usuario

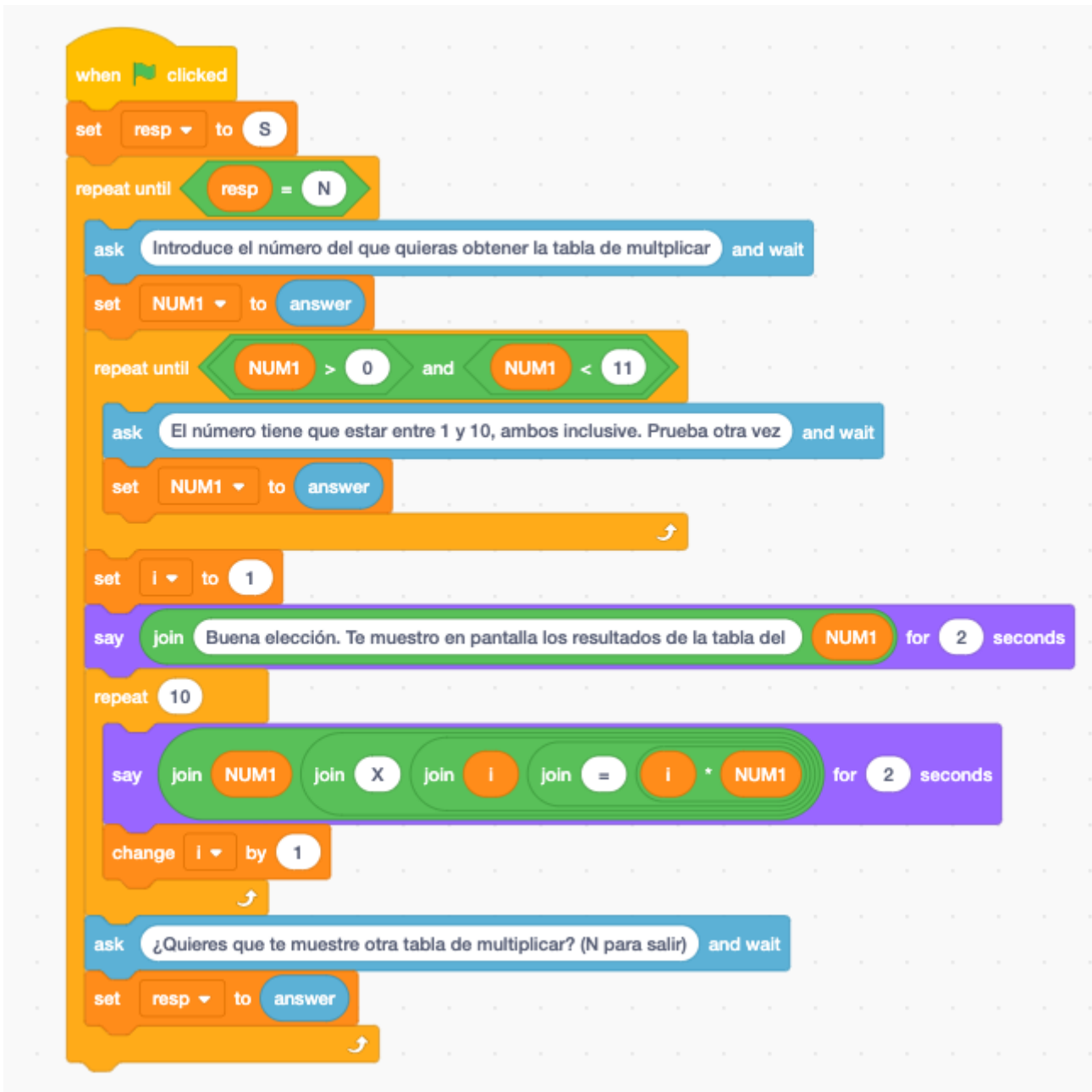
El diagrama de flujo que resuelve este problema sería algo parecido a lo siguiente:



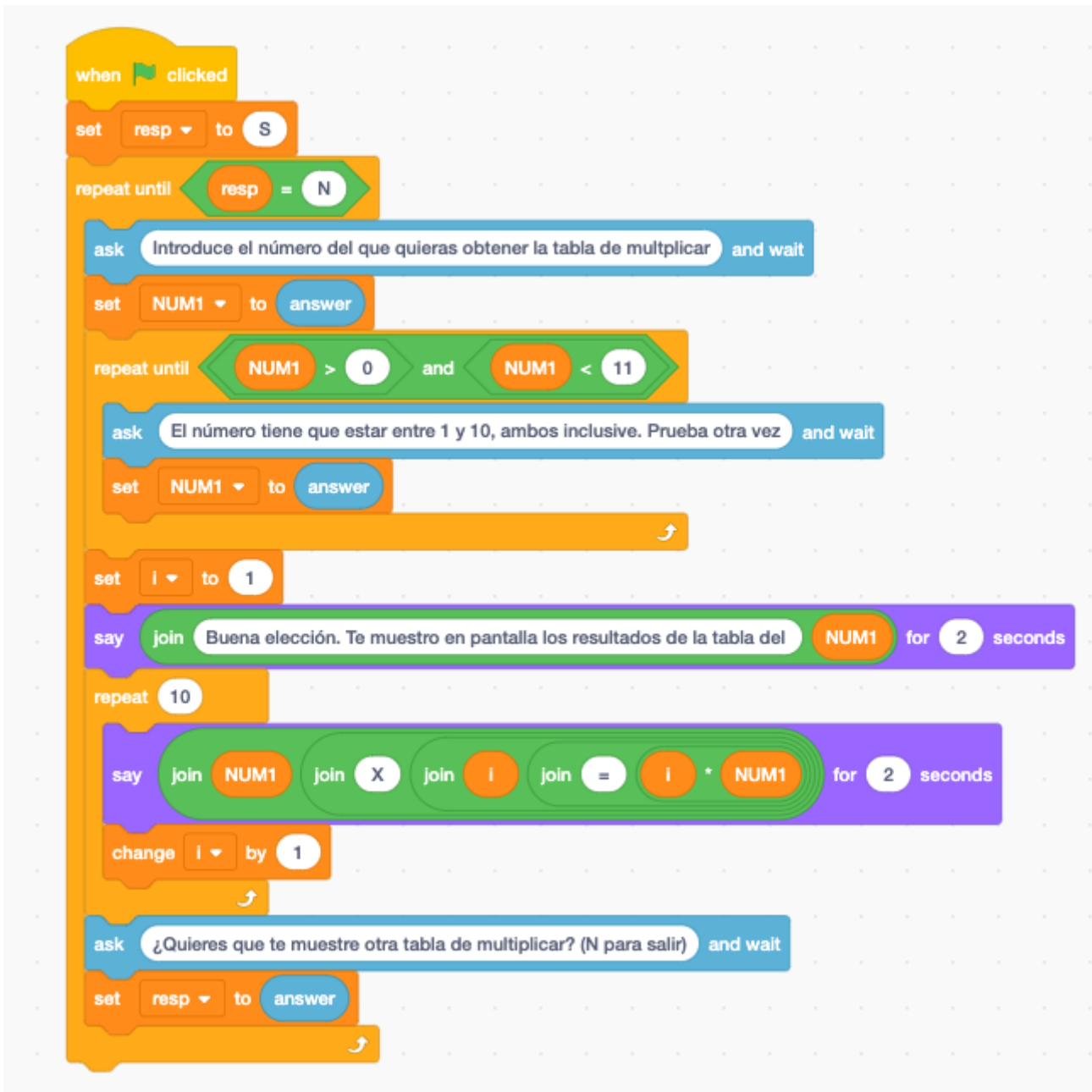
Se recomienda siempre que el alumnado no intente programar el ejercicio en su totalidad, sino que vayan identificando las cosas que tienen que hacer, comiencen por el problema a resolver y posteriormente las particularidades. En este caso, se recomienda que los pasos a la hora de afrontar el algoritmo sean los siguientes:

1. Programar la lógica referente a pedir un número, visualizar su tabla de multiplicar y finalmente preguntar si quiere seguir jugando, todo ello dentro de un bucle principal.
2. Programar el requisito de que el número esté comprendido entre 1 y 10, ambos inclusive
3. Programar que el usuario solo pueda introducir una S o una N ante la pregunta de si quiere seguir jugando.

El programa en su fase 1 tendría un código parecido al siguiente:



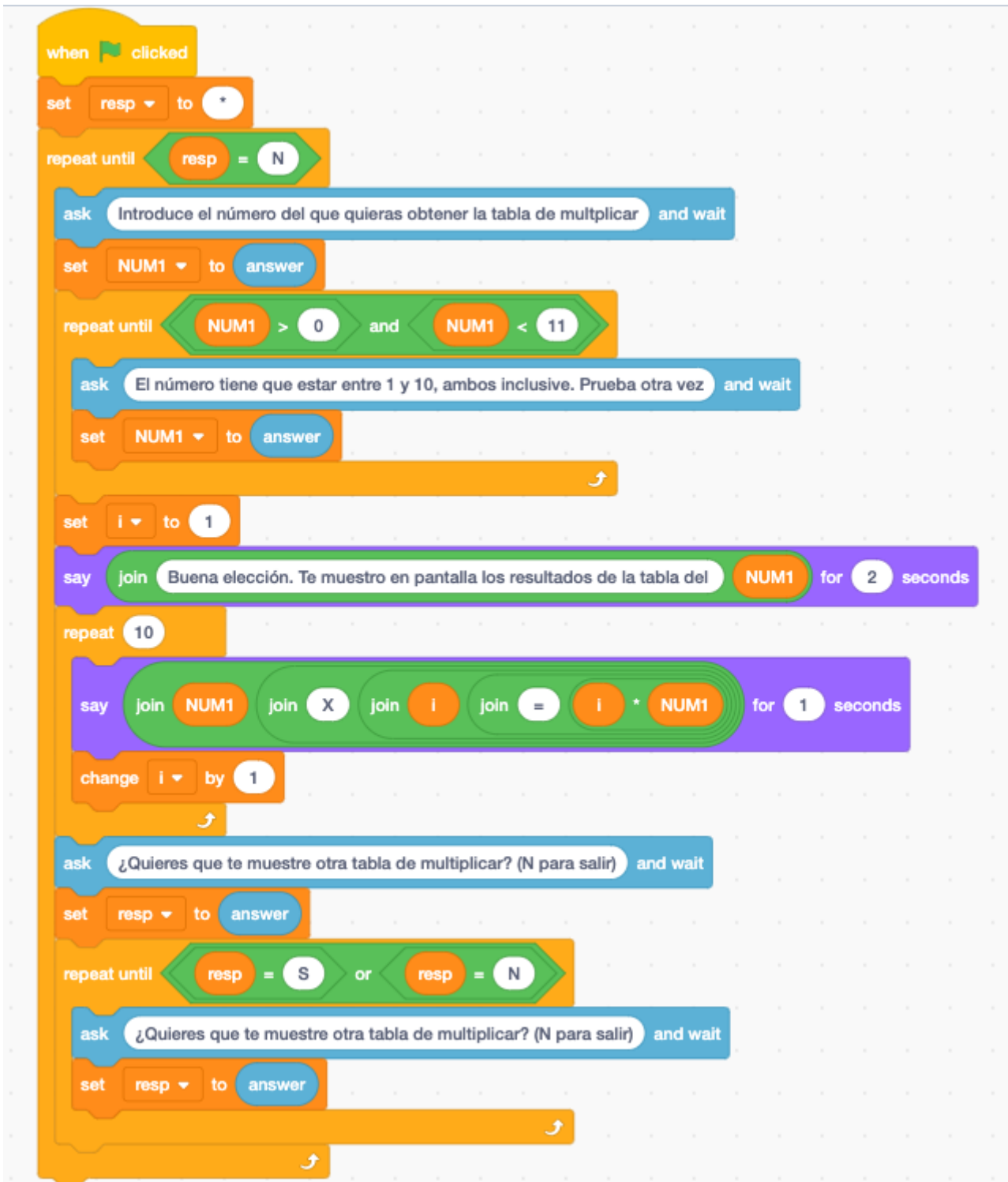
Cuando ya sabemos que nuestro programa funciona en líneas generales, podríamos incluir la programación de la limitación de que hasta que el usuario no introduzca un número entre 1 y 10 ambos inclusive, el programa no continua:



```

when clicked
  set resp to S
  repeat until (resp = N)
    ask "Introduce el número del que quieras obtener la tabla de mltiplicar" and wait
    set NUM1 to answer
    repeat until (NUM1 > 0 and NUM1 < 11)
      ask "El número tiene que estar entre 1 y 10, ambos inclusive. Prueba otra vez" and wait
      set NUM1 to answer
    set i to 1
    say join "Buena elección. Te muestro en pantalla los resultados de la tabla del" NUM1 for 2 seconds
    repeat 10
      say join NUM1 join " X " join i join " = " join i join " * " join NUM1 for 2 seconds
      change i by 1
    ask "¿Quieres que te muestre otra tabla de multiplicar? (N para salir)" and wait
    set resp to answer
  
```

Finalmente, introducimos la programación de que la respuesta final del usuario solo puede ser S o N. Es el mismo tipo de bucle que en la anterior limitación:



```

when clicked
  set resp to *
  repeat until resp = N
    ask Introduce el número del que quieras obtener la tabla de multiplicar and wait
    set NUM1 to answer
    repeat until NUM1 > 0 and NUM1 < 11
      ask El número tiene que estar entre 1 y 10, ambos inclusive. Prueba otra vez and wait
      set NUM1 to answer
    set i to 1
    say join Buena elección. Te muestro en pantalla los resultados de la tabla del NUM1 for 2 seconds
    repeat 10
      say join NUM1 join X join i join = join i * NUM1 for 1 seconds
      change i by 1
    ask ¿Quieres que te muestre otra tabla de multiplicar? (N para salir) and wait
    set resp to answer
    repeat until resp = S or resp = N
      ask ¿Quieres que te muestre otra tabla de multiplicar? (N para salir) and wait
      set resp to answer
  
```

Un aspecto importante a destacar es que si bien es cierto que en la programación de ambas limitaciones hemos utilizado el mismo tipo de bucle con dos condiciones a tener en cuenta, en el primero hemos utilizado una unión de esas dos condiciones con el bucle **AND**, puesto que **necesitábamos que ambas se cumplieran**, y en el segundo hemos utilizado un operador **OR** porque **nos bastaba con que se cumpliera una de las dos**. Estaría bien enseñar a los alumnos (si es que aún no lo saben) como son las tablas lógicas del AND y del OR.

a	b	a + b	a	b	ab
0	0	0	0	0	0
0	1	1	0	1	0
1	0	1	1	0	0
1	1	1	1	1	1

a	a'
0	1
1	0

Fuente:

<https://www.monografias.com/trabajos104/simplificacion-circuitos-logicos-algebra-conmutacion/img4.png>

Este ejercicio, aunque parezca sencillo en su formulación, contiene en su programación unos cuantos aspectos básicos en el aprendizaje de programación que deben quedar bien asentadas para futuros aprendizajes más profundos.

Finalmente, la experiencia de juego sería algo parecida a lo siguiente:

<https://scratch.mit.edu/projects/746099667/embed>

Propuesta Extra Opción 2. Visualizando todas las tablas de multiplicar del 1 al 10

¿Te atreves a realizar este programa por tu cuenta? Y la experiencia de juego sería como se puede ver a continuación:

<https://scratch.mit.edu/projects/746109344/embed>

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



Revision #8

Created 2022-10-15 16:35:20 CEST by Cristian Ruiz

Updated 2023-01-17 15:51:21 CET by Equipo CATEDU