



# Guía Didáctica Segundo Nivel



**CONSTRUYENDO ALGORITMOS**



# ¿Cómo se programa un videojuego?

PARA CREAR CUALQUIER PROGRAMA INFORMÁTICO (COMO UN VIDEOJUEGO) SIEMPRE DEBERÁS CREAR UNA SOLUCIÓN SIGUIENDO ESTOS TRES PASOS

## Algoritmo

Un algoritmo es una secuencia de pasos lógicos necesarios para llevar a cabo una tarea específica, como la solución de un problema.

## Flujograma

El diagrama de flujo o flujograma es la representación gráfica del algoritmo o proceso. Puedes utilizar símbolos como:



## Programa (código)

Tu código de videojuego puede ser programado en lenguaje visual o de texto. En este caso puedes observar un programa (código) que permite al jugador darle instrucciones a un objeto (un personaje, un auto, un balón u otro) para que llegue a su destino.



# Ejemplo

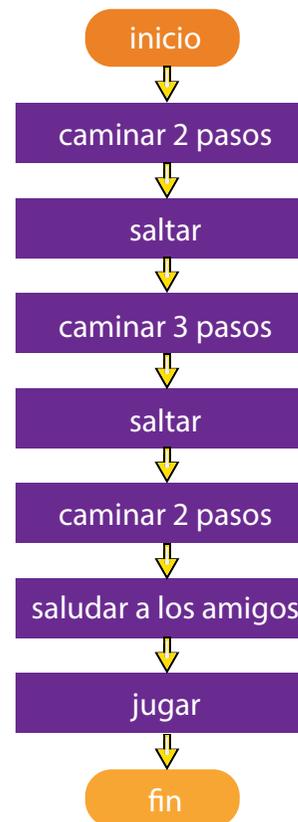
Observa como coddic sigue los pasos para evadir los obstaculos en el camino y llegar al parque.



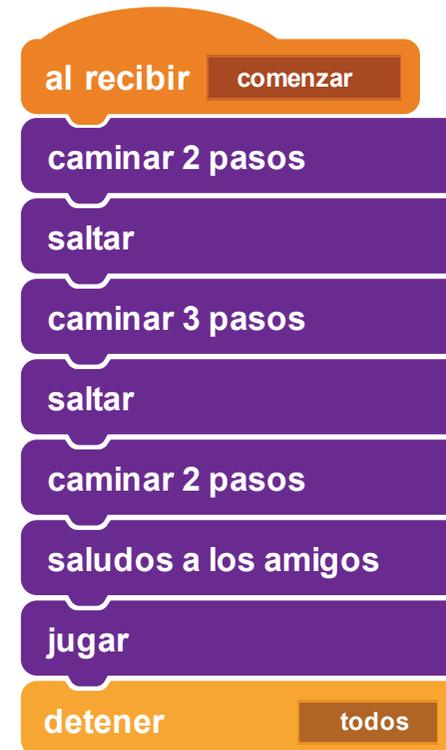
## Algoritmo

1. inicio
2. caminar 2 pasos
3. saltar
4. caminar 3 pasos
5. saltar
6. caminar 2 pasos
7. saludar a los amigos
8. jugar
9. fin

## Flujograma



## Programa (bloques de código en scratch)



[ver código](#)



*¿Cómo se programa  
un videojuego?*



# ¿Cómo creo mis videojuegos?



## Lenguajes de programación Visuales

Desarrollan tu creatividad, capacidad de resolución de problemas o el trabajo en equipo, que serán muy importantes para tí en tu futuro campo profesional. *Scratch* es un lenguaje de programación visual donde puedes crear videojuegos como:

Super Mario  [ver código](#)

Tetris  [ver código](#)

Star Wars  [ver código](#)



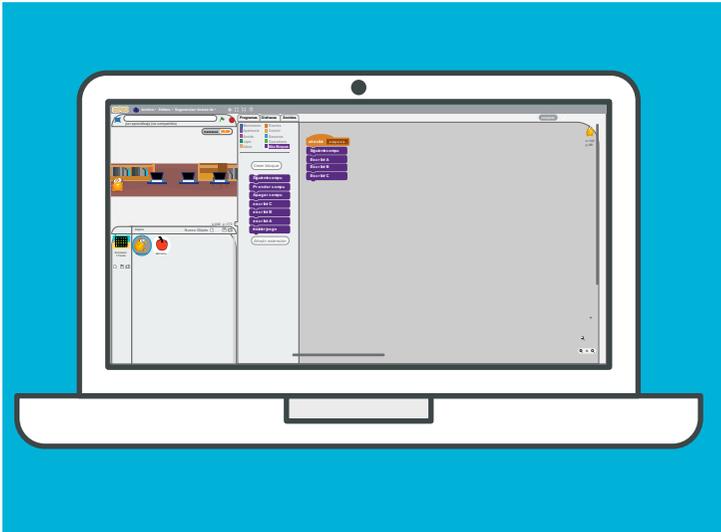
## Lenguajes de programación de texto

Exigen mayor precisión al momento de escribir cada línea de código. Generalmente te piden aprenderte de memoria algunas instrucciones. *Pilas Engine* es un lenguaje de texto que más adelante aprenderemos, aunque actualmente ya posee su propio lenguaje visual, llamado *Pilas Bloques*

Pilas Bloques  [ver código](#)



## ¿Cómo creo mis videojuegos?



### Scratch Online

Te permite crear tu videojuegos, ver los de otras personas y hasta editarlos y a tu gusto. Puede crear proyectos y compartirlo con tus amigos.

Para usarlo debes tener como requisito un email personal y crear un cuenta de usuario; además de disponer de internet en tu computador.

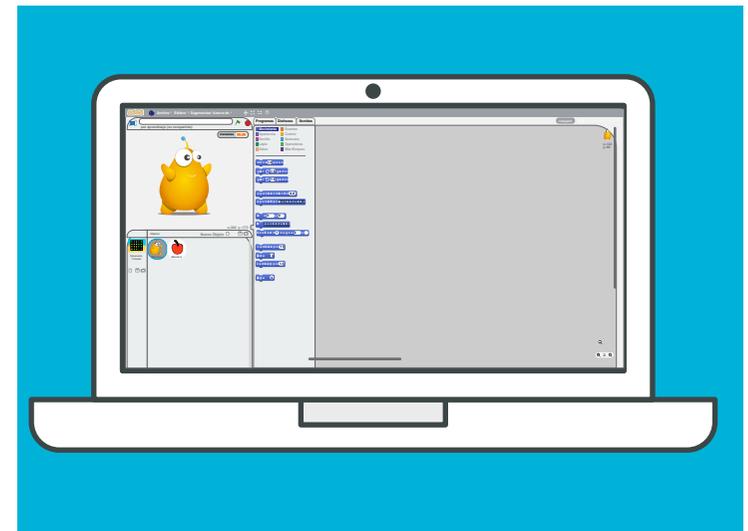
 <https://scratch.mit.edu>

### Scratch Offline

Puedes instalarlo en tu computador y usarlos para crear videojuegos aunque no tengas internet en casa.

No olvides que luego de crear tus videojuegos con esta aplicación, podrás compartirlos en tu cuenta de Scratch Online.

 <https://scratch.mit.edu/download>



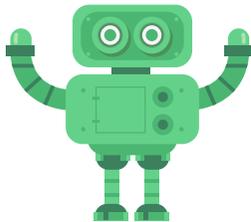


*Mis amigos  
“los autómatas”*



## ¿Qué son los autómatas u objetos?

Un autómata puede ser cualquier objeto que puedes utilizar para crear tu videojuego. Fíjate en algunos autómatas que podrías usar en tus videojuegos, los cuáles seguirán tus instrucciones u órdenes, como por ejemplo caminar. **RECUERDA QUE UN AUTÓMATA NO SIEMPRE ES UN ROBOT**



observa los siguientes objetos y autómatas en la siguiente interfaz (pantalla) de videojuego.



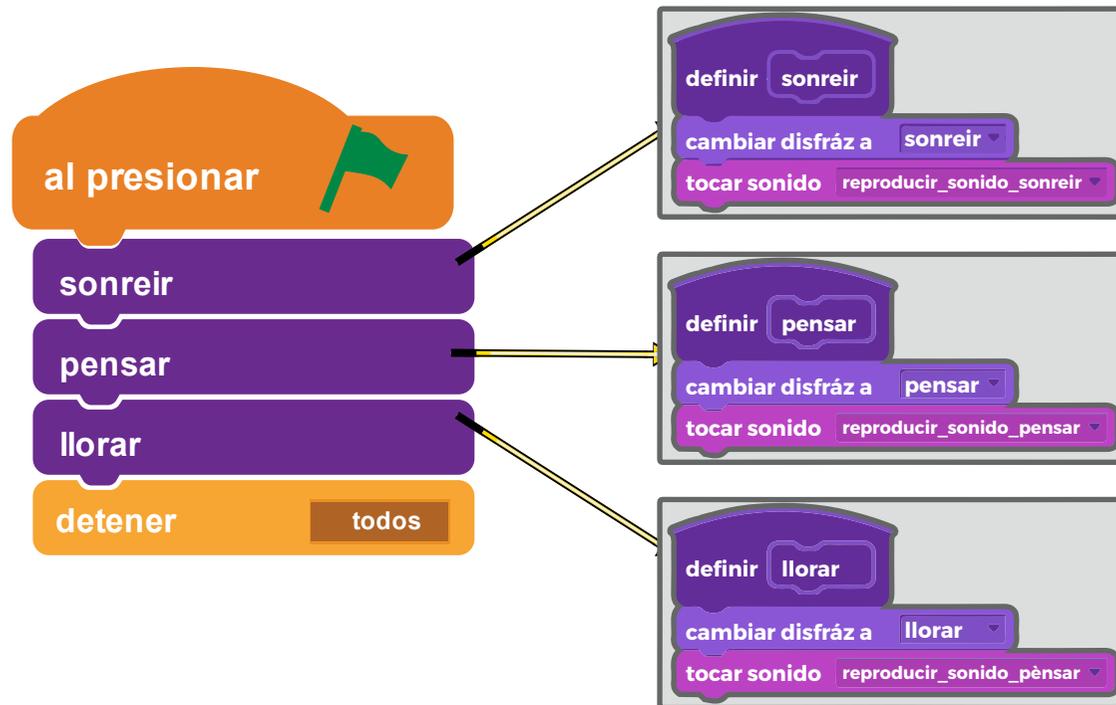




## Ejemplo

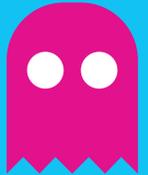
Este automáta llamado coddic puede tener varias acciones u órdenes como se visualizan a continuación en un conjunto de bloques de Scratch.

### Programa (bloques de código en Scratch)



 [ver código](#)

Debes tener en cuenta que un algoritmo o programa informático debe tener las instrucciones bien definidas y detalladas para que el autómata sepa que hacer. En el ejemplo anterior deberías indicarle cómo sonreir, pensar o llorar a coddic, como se muestra en la imagen anterior.



# *Eventos y movimientos*



# Bloques de eventos y movimientos

## Bloques de Eventos (inicio del juego)

- al presionar
- al presionar tecla espacio
- al hacer clic en este objeto
- cuando el fondo cambie a backdrop
- cuando volumen de sonido sea > 10
- al recibir mensaje 1
- enviar mensaje 1
- enviar mensaje 1 y esperar

Los eventos permiten detectar acciones del usuario o del propio videojuego. Si comparas los flujogramas con los bloques de eventos te darías cuenta que: INICIO es igual a

## Bloques de Movimientos

Te permitirán hacer que un objeto:

- se mueva
- se deslice
- se gire
- rebote

- mover 10 pasos
- girar 10 grados
- girar 10 grados
- apuntar en dirección 90
- apuntar hacia puntero del raton
- ir a x 0 y y 0
- ir a puntero del raton
- deslizar en 1 segs a x 0 y y 0
- cambiar x por 10
- fijar x a 0
- cambiar y por 10
- fijar y a 0
- rebotar si toca un borde
- fijar estilo de rotación izquierda-derecha
- posición en x
- posición en y
- dirección



# *Apariencia y sensores*



# Bloques de apariencia

Las instrucciones que se encuentran en este Bloque se encargan como su nombre lo indica de cambiar el aspecto o la forma como se observan los objetos en el escenario, así como también cambiar la posición del objeto en el escenario, poniendo al objeto en capas posteriores. OBSERVA ALGUNOS EJEMPLPLOS DEL USO DE BLQOUES DE EVENTOS Y APARIENCIA

decir **hello**

pensar **Hmm** por **2** segundos

cambiar tamaño por **25**

enviar al frente

siguientedisfraz

cambiar fondo a **backdrop 1**

al presionar

pensar **¿Quién soy?**



al hacer clic en este objeto

cambiar disfraz a **costume 2**

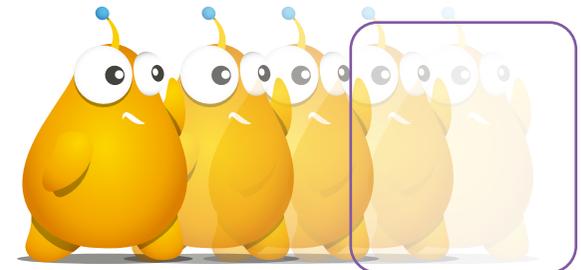


al presionar tecla **espacio**

mostrar

al presionar tecla **E**

esconder



al presionar tecla **H**

decir **MUY BIEN** por **2** segundos

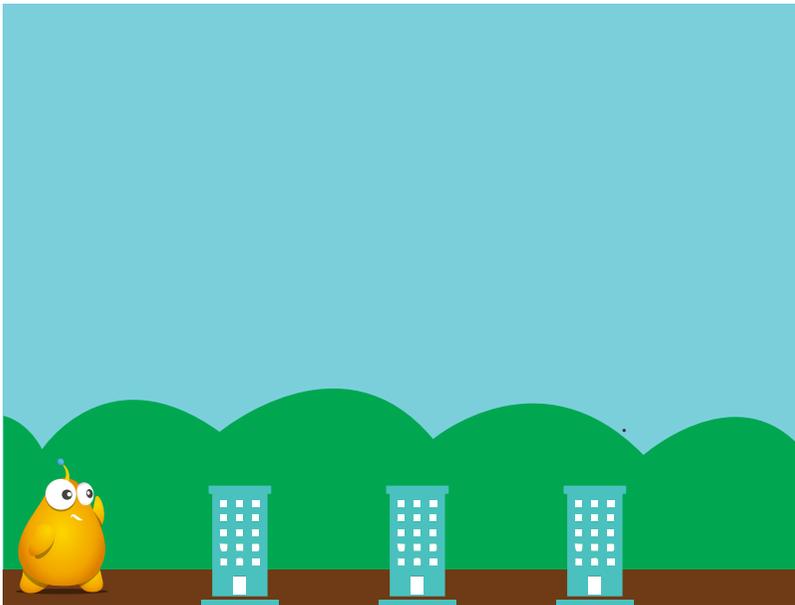




# Ejemplo

coddic debe destruir los números que no son binarios para salvar la galaxia. Al darle clic puedes cambiar por una nave más poderosa. OBSERVA COMO SE HA PROGRAMADO SUS CAMBIOS DE DISFRAZ (nave) Y MENSAJES.

## Interfaz



## Programa

- al presionar tecla **flecha arriba**
  - apuntar en dirección **0**
- al presionar tecla **flecha derecha**
  - apuntar en dirección **90**
- al presionar tecla **flecha izquierda**
  - apuntar en dirección **-90**
- al presionar tecla **O**
  - ir a x **0** y **0**

## Acciones

- al presionar tecla **espacio**
  - disparar

## Cambio de disfraz (nave)

- al hacer clic en este objeto
  - cambiar disfraz a **costume 2**

## Pensar y decir (frases en pantalla)

- al presionar **bandera**
  - Muestra una frase de saludo al inicio del juego.
  - decir **Hola amigos defensores de la galaxia**
- al presionar tecla **E**
  - esconder
  - al presionar la tecla O la nave desaparecerá.
- al presionar tecla **M**
  - mostrar
  - al presionar la tecla M la nave volverá a aparecer en pantalla.

Este juego puedes reinventarlo a tu gusto, ubicando otras instrucciones, acciones o mensajes en cada objeto [ACCEDE AL CÓDIGO DESDE EL SIGUIENTE ENLACE.](#)  [ver código](#)



# Bloques de control y sensores

Estos dos tipos de bloques de programación pueden ser combinados para controlar y ejecutar acciones en distintos objetos.

## Controlan la secuencia de sentencias

- esperar 1 segundos
- repetir 10
- por siempre
- si entonces
- si entonces si no
- esperar hasta que
- repetir hasta que
- detener todos

## Detectan acciones del jugador o de objetos

- Tocar otro Objeto.
  - ¿tocando puntero del ratón ?
- Tocar un color determinado.
  - ¿tocando el color
- Cuando un color determinado toca otro color.
  - ¿color tocando ?
- Entrada de variables por medio del teclado.
  - distancia a puntero del ratón
- Mostrar una mensaje y que el jugador escriba una respuesta
  - preguntar what's your name? y esperar
- Usar una respuesta escrita por el usuario
  - respuesta
- Si se presionó alguna tecla en específico.
  - ¿tecla espacio presionada?
- Si se presionó alguna tecla del ratón.
  - ¿ratón presionado?

```

al presionar tecla espacio
preguntar ¿Cómo te llamas? y esperar
  
```



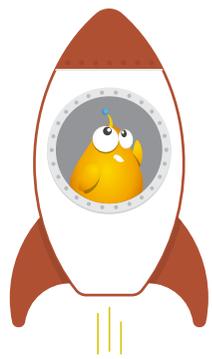
```

al hacer clic en este objeto
si ¿tocando trofeo ? entonces
  festejar
  
```



```

al presionar tecla 9
repetir 9
  disparar
  
```





# *Repeticiones*



# Bloques de repetición

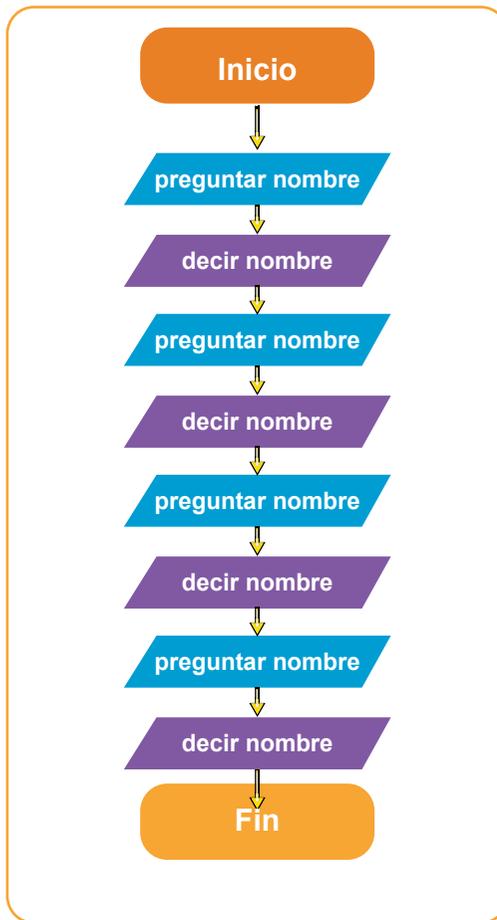
Los bloques de repetición simple te permiten que un conjunto de sentencias sean repetidas, determinando uno el número de veces que estos ciclos se repiten de acuerdo al objetivo que desees.

## Problema

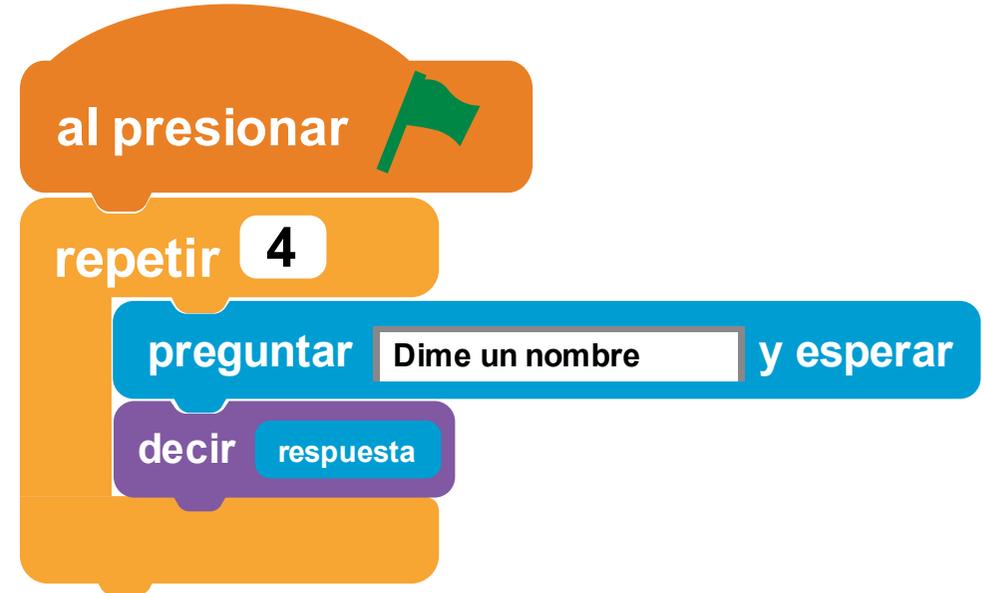


Pregunta y repite un nombre que el jugador escribe por 10 veces.

## Flugograma



## Programa





# *Repeticiones, operadores y datos*



# Bloques de operadores y datos

Puntajes en mis juegos: Los puntajes en los videojuegos pueden ser controlados con estos bloques que siempre funcionan juntos



El Bloque de Operadores contiene una serie de instrucciones que te va a permitir hacer una variedad de operaciones las cuales se encuentran divididas en tres grupos:



- Operaciones matemáticas.
- Operaciones booleanas
- Operaciones con cadenas de texto.



número al azar entre 1 y 10



unir hello world



letra 1 de world



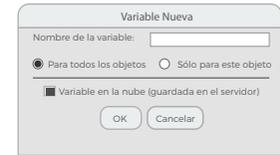
longitud de world



El Bloque de Datos permite:



En programación, las variables son espacios reservados en la memoria que, como su nombre indica, pueden cambiar de contenido a lo largo de la ejecución de un programa



Guardar datos (números, letras, símbolos u otros)



Cambiar datos guardados



Mostrar un dato en la pantalla



Esconder un dato de la pantalla

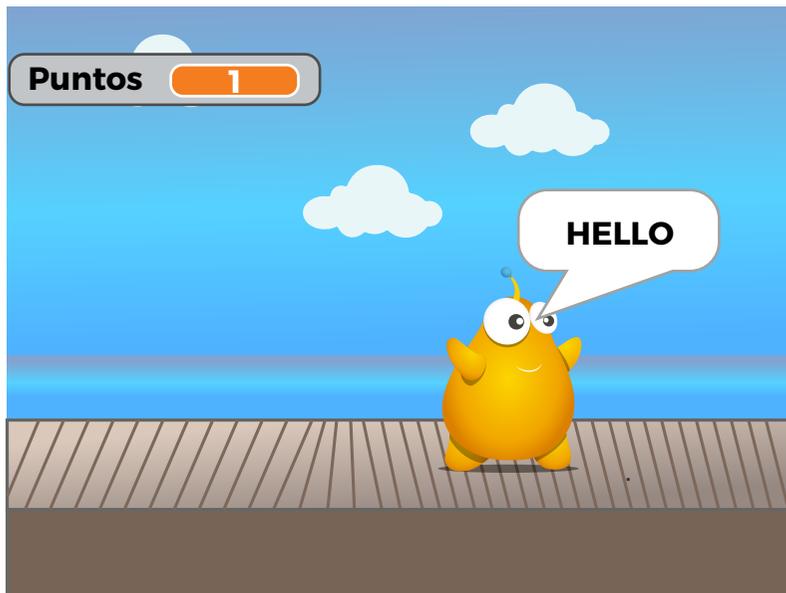




## Ejemplo

El siguiente juego puedes observar que coddic debe recoger monedas, y estrellas para llegar al trofeo. coddic debe ir respondiendo preguntas que tu puedes ayudarle a responder.

### Interfaz



### Pregunta inicial



En este juego se puede ver como se combinan bloques de datos y operadores que permiten:

1. usar variables (puntaje). Cada vez que reinicia el juego, el puntaje vuelve a ser 0.
2. controlar (siempre) mediante una pregunta ( si... entonces) . Comparar valores (si el puntaje es mayor a 10 muestra una animación de coddic)
3. Hacer una suma de dos números dados por el jugador

Este juego puedes reinventarlo ACCEDIENDO AL CÓDIGO DESDE EL SIGUIENTE ENLACE.

 [ver código](#)



# *Control (repetición) Condicional*



# Bloques de condición

Las condiciones te permitirán controlar el flujo de acciones de tu videojuego de acuerdo a una pregunta que Scratch analizará y detectará si es verdadera o falsa la respuesta.

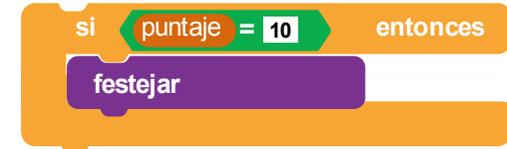
Controla el flujo de acciones **SOLAMENTE CUANDO LA CONDICIÓN ES VERDADERA**



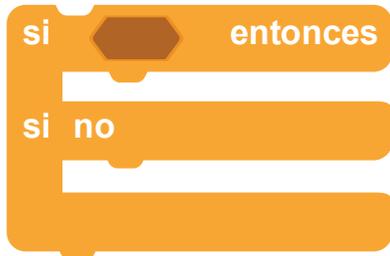
### algoritmo

Si puntaje es igual a 10 entonces  
coddic gana

### programa



Controla el flujo de acciones **CUANDO LA CONDICIÓN DEVUELVE UNA RESPUESTA VERDADERA O FALSA**



### algoritmo

Si puntaje es igual a 10 entonces  
festejar  
esconder puntaje  
Si no  
mostrar puntaje

### programa



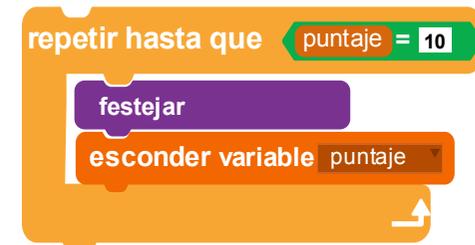
Repite un número determinado de veces un conjunto de acciones de acuerdo a una condición dada. La ventana es de este bloque es que **evitas usar dos bloques : 1 de control de repetición y 1 de condición.**



### algoritmo

Repetir hasta que puntaje sea igual a 10  
esconder puntaje  
mostrar coddic

### programa



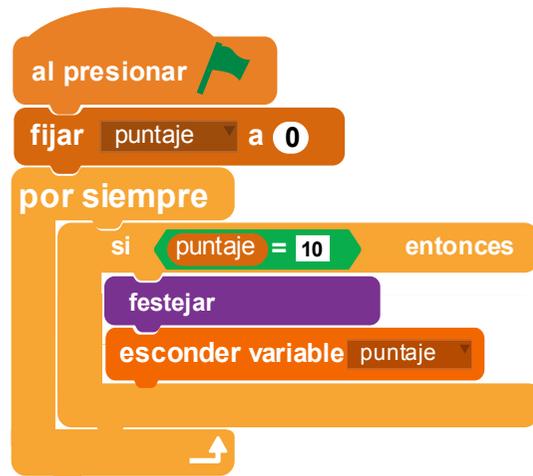


## Ejemplo

Este juego hace que coddic recoja premios (estrellas, diamantes, vidas) mientras caminar por el las montañas del Ecuador. Al mismo tiempo coddic debe evitar chocar con los oponentes del camino.

Aquí se usa dos bloques:

1. **Bloque de repetición** --> por siempre
2. **Bloque de condición** --> si....entonces



Aquí se controla que el puntaje llegue a 10 y en ese momento se reproduzca una animación de ganar. También se controla el número de vidas que si llega a 0 se detiene el juego.

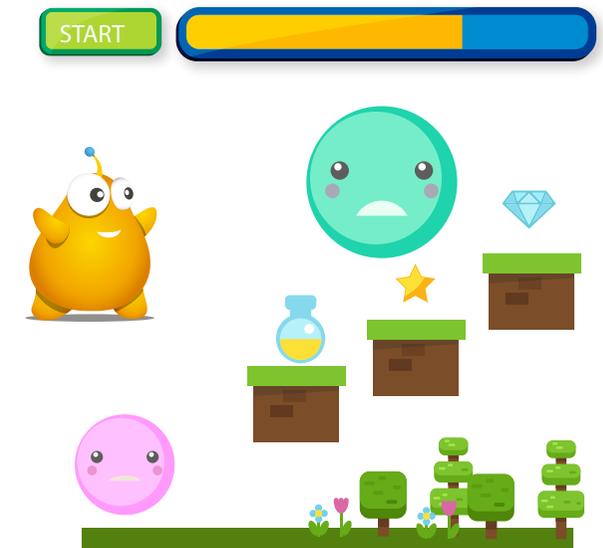
FÍJATE QUE AMBOS GRUPOS DE BLOQUES CUMPLEN LA MISMA FUNCIÓN PERO DE FORMA DIFERENTE.

Aquí se usa un bloque:

1. **Bloque de repetición condicional** --> repetir hasta que



### Interfaz



Este juego puedes reinventarlo ACCEDIENDO AL CÓDIGO DESDE EL SIGUIENTE ENLACE.

 [ver código](#)



# *Control disfraces y sensores II*



## Bloques de control, sensores y apariencia

A continuación verás algunas combinaciones de bloques que puedes usar para la creación de tus videojuegos.



Aquí puedes ver una combinación de bloques de :

### 1. Bloque de eventos

- Al presionar bandera verde

### 2. Bloques de control

- Por siempre
- Si \_\_\_\_\_ entonces

### 3. Bloque de sensores

- ¿Tocando ?

### 4. Bloque de apariencia

- Cambiar disfraz
- Cambiar tamaño

Estas combinaciones de bloques te permitirán controlar siempre los choques entre objetos, como en este caso la estrella con coddic. Al darse el choque se ejecutarán acciones que imagines y ubiques, como en este caso coddic cambia de disfraz y crece un 10%.





# *Sonidos*



# Creación y uso de sonidos

El lenguaje de SCRATCH, cuenta con un grupo de instrucciones que permiten la reproducción de archivos de sonido dentro de la animación que estés haciendo. Estos archivos de sonido los puedes importar directamente al área del Objeto y una vez ahí puedes usar las instrucciones para su reproducción durante la animación.

## Sonido

tocar sonido **meow**

tocar sonido **meow** y esperar

detener todos los sonidos

tocar tambor **1** durante **0,25** pulsos

silencio por **0,25** pulsos

tocar nota **60** durante **0,5** pulsos

fijar instrumento a **1**

cambiar volumen por **-10**

fijar volumen a **100** %

volumen

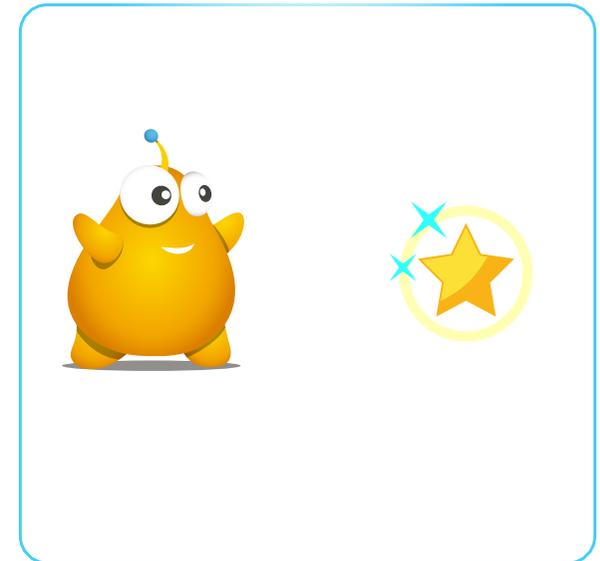
cambiar tempo por **20**

fijar tempo a **600** ppm

tempo

```

al presionar
  fijar puntaje a 0
  por siempre
    si tocando estrella entonces
      cambiar puntaje por 1
      decir Wow tengo una estrella más por 2 segundos
      tocar sonido recoge moneda
  
```

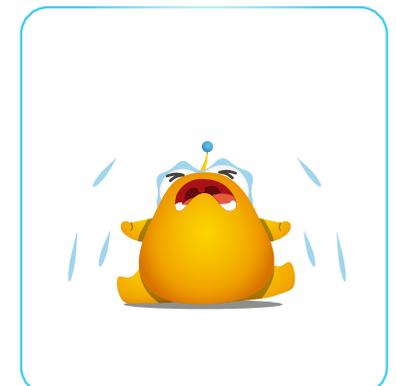


```

al presionar
  por siempre
    si puntaje = 0 entonces
      perder
  
```

```

definir perder
  decir he perdido por 2 segundos
  tocar sonido sonido_perder
  cambiar disfraz a llorar1
  cambiar disfraz a llorar2
  cambiar disfraz a llorar3
  cambiar disfraz a llorar4
  
```





# Ejemplo

FÍJATE QUE CADA DIBUJO TIENE POSICIONES DIFERENTES : parado, alzando el pie derecho, alzando el pie izquierdo entre otros.

The screenshot shows the Scratch interface with the 'Sonidos' (Sounds) editor open. A yellow character is on the stage. The 'Objetos' (Objects) palette shows the character and a star. The 'Sonidos' editor shows a sound waveform and a volume slider. Annotations 1, 2, and 3 highlight specific elements: 1 points to the character in the objects palette, 2 points to the 'Fondos' (Backgrounds) tab, and 3 points to the sound editor area.

**1**

**2**

**3**

FÍJATE que puedes aplicar efectos a los sonidos

RECUERDA que los sonidos puedes controlarlos con bloques de código.

Tú puedes usar cuando sonidos necesites para tu videojuego

Este juego puedes reinventarlo ACCEDIENDO AL CÓDIGO DESDE EL SIGUIENTE ENLACE.

[ver código](#)

# CONSTRUYENDO ALGORITMOS

## Guía Didáctica Segundo Nivel



 coddic  
 coddic  
 @coddic\_ec

